université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

NNT : 2017SACLN054

école normale supérieure paris—saclay

THÈSE DE DOCTORAT

de

L'Université Paris-Saclay

École doctorale de mathématiques Hadamard (EDMH, ED 574)

Établissement d'inscription : Ecole normale supérieure de Paris-Saclay

Laboratoire d'accueil : Centre de mathématiques et de leurs applications, UMR 8536 CNRS

Spécialité de doctorat : Mathématiques appliquées

Thomas MOREAU

Représentations Convolutives Parcimonieuses – application aux signaux physiologiques et interprétabilité de l'apprentissage profond

Date de soutenance : 19 Décembre 2017

Après avis des rapporteurs :
Julien MAIRAL (INRIA Grenoble)
Stéphane MALLAT (École Normal Supérieure)
René VIDAL (Université Johns Hopkins)

Jury de soutenance :

Stéphanie ALLASSONNIÈRE	(Université Paris-Descartes) Présidente
Alexandre GRAMFORT	(INRIA Saclay) Examinateur
Julien MAIRAL	(INRIA Grenoble) Rapporteur
Stéphane MALLAT	(École Normale Supérieure) Rapporteur
Laurent OUDRE	(Université Paris 13) Codirecteur de thèse
Nicolas VAYATIS	(ENS Paris-Saclay) Codirecteur de thèse
Pierre-Paul VIDAL	(Université Paris-Descartes) Examinateur
René VIDAL	(Université Johns Hopkins) Rapporteur

Fondation mathématique FMJH Jacques Hadamard

CMLA

Contents

Acknowledgement

Foremost, I would like to thank my advisors, Nicolas Vayatis and Laurent Oudre, for their support during these three years. They gave me the perfect environment to develop my ideas and the wonderful opportunity to work on exciting real life projects. Thanks to both of you for making these three years such a interesting time in my life, for your encouragements and for the many opportunities and collaborations you made possible.

I would like to thank Julien Mairal, Stéphane Mallat and René Vidal for doing me the honor of reviewing this manuscript and for their insightful comments that helped me improve its quality. I am also grateful to Stéphanie Allassonnière, Alexandre Gramfort and Pierre-Paul Vidal for accepting to be part of the jury.

My gratitude also goes to the people who participated, directly or not, to the elaboration of this thesis, and in particular my co-authors Joan Bruna and Julien Audiffren, with whom I enjoyed working and discussing the ideas we developed. It has also been a pleasure to work with Olivier Grisel, who taught so much and continue to do so during our Fridays coding sessions, and with whom I enjoyed sharing all these tiramisus.

I would also like to express my gratitude to all the people from the Cognac-G team with whom I had the chance to collaborate during this thesis. The study of the human walking with the Cognac-G team have been a thrilling experience and I have enjoyed my interactions with medical doctors: Damien Ricard, Pierre-Paul Vidal, Alain Yelnik, Catherine De Waele, Stéphane Buffat, Alfredo Pulini, Clément Provost and a special note for Remi Barrois-Müller, with whom I shared the hardship of robust "data annotation". I would also like to thanks Pierre-Paul Vidal, Matthieu Robert, Emile Contal, Pierre Humbert and Ludovic Minvielle for the Oculo project, which has been very instructive and exciting.

A big thanks to everyone that have been supporting me for these past three years. Many thanks to my colleagues at CMLA, Julien, Thomas, Mathilde, Pierre, Argyris, Batiste, Rémi D, Rémi L, Myrto, Steven, Ludo, Alice, Kévin, Mounir, Ioannis, Juan and Asma for the good mood in the lab and thanks to Cedric and Charles, for sharing this three year adventure, with its ups and downs. Also, many thanks to my friends for all the energy they gave me, les volleyeurs, la Kès, Youmpi and the Brocoloc. A special thought goes to Alexandre, Coralie and Hugo, who were always there when I needed the most. To my family, who always pushed me to go higher and to achieve my goals, I am very grateful that you supported me for all these years. Finally, thank you Marine for sharing my life during this adventure and being here for me.

Abstract

Convolutional representations extract recurrent patterns which lead to the discovery of local structures in a set of signals. They are well suited to analyze physiological signals which requires interpretable representations in order to understand the relevant information. Moreover, these representations can be linked to deep learning models, as a way to bring interpretability in their internal representations. In this dissertation, we describe recent advances on both computational and theoretical aspects of these models.

Our main contribution in the first part is an asynchronous algorithm, called DICOD, based on greedy coordinate descent, to solve convolutional sparse coding for long signals. Our algorithm has super-linear acceleration. We also explored the relationship of Singular Spectrum Analysis with convolutional representations, as an initialization step for convolutional dictionary learning.

In a second part, we focus on the link between representations and neural networks. Our main result is a study of the mechanisms which accelerate sparse coding algorithms with neural networks. We show that it is linked to a factorization of the Gram matrix of the dictionary. Other aspects of representations in neural networks are also investigated with an extra training step for deep learning, called post-training, to boost the performances of trained networks by improving their last layer's weights.

Finally, we illustrate the relevance of convolutional representations for physiological signals. Convolutional dictionary learning is used to summarize signals from human walking and Singular Spectrum Analysis is used to remove the gaze movement in young infant's oculometric recordings.

Notation

General

$\mathcal{O}_n$	Orthogonal matrices in $\mathbb{R}^{n\times n}$
$L_2(\mathbb{R})$	Space of functions f from $\mathbb{R}$ to $\mathbb{R}$ with $\int_{\mathbb{R}} f^2 < +\infty$
$[\![n_1, n_2]\!]$	Ensemble of integers between n_1 and n_2
$\boldsymbol{I}_n$	Identity matrix in $\mathbb{R}^{n\times n}$

Time series

$\mathcal{X}_T^P$	Ensemble of multivariate signals in $\mathbb{R}^P$ of length $T \in \mathbb{N}$
X	Element (signal) of $\mathcal{X}_T^P$
$X[t]$	Value of the signal X at time sample t.
X_p	p-th channel of X, for $p \in [\![1, P]\!]$. Note that for all $t \in [\![0, T-1]\!]$ $$X[t] = \begin{pmatrix} X_1[t] \\ .. \\ X_P[t] \end{pmatrix}$$
$z * D$	The convolution between $z \in \mathcal{X}_L^1$ and $D \in \mathcal{X}_W^P$. The resulting signal X is in $\mathcal{X}_T^P$, with length $T{=}L{+}W{-}1$, and for $[\![0, T-1]\!]$, $$X[t] = (z * D)[t] = \sum_{\tau=0}^{W-1} z[t-\tau]D[\tau] \ .$$

1 Résumé

Contents

1.1 Motivations de la thèse : de la caractérisation à la compréhension

1.1.1 L'acquisition en continu : le cas des signaux temporels

Au cours des dernières décennies, les capteurs suivant l'évolution de notre environnement, de notre comportement ou de nos activités se sont multipliés. Sur internet, les entreprises de publicité enregistrent les pages que nous visitons, le temps passé sur celles-ci et même les mouvements de la souris dans la page. Dans les villes, des capteurs sont installés pour enregistrer quantité d'information sur l'activité de la population ou sur la qualité de l'air. Les badges de transport en commun permettent de suivre chaque jour les trajets de millions de personnes. Les caméras de vidéo-surveillance enregistrent en continu les flux de personnes et de vehicules dans les rues. Après acquisition, ces informations sont enregistrées dans d'énormes bases de données à travers le monde. Cependant, peu d'informations sont extraites de ces données, relativement à leur volume. Ces signaux qui suivent l'évolution de notre vie quotidienne ne sont pas très bien compris. Pour la vidéo-surveillance, des algorithmes extraient automatiquement les objets présents à l'image, mais l'intervention humaine est nécessaire pour comprendre la scène et détecter les dangers éventuels. Le traitement des signaux longs et multivariés est en effet une tache complexe qui demande d'extraire les événements dans le temps, de caractériser les comportements normaux et d'être capable de détecter les anomalies dans le signal. L'étude des propriétés statistiques globales des données enregistrées n'est souvent pas suffisante pour accéder à ce genre d'informations. La moyenne et la variance des caractéristiques des séries temporelles ne permettent pas de distinguer des différences fines dans le temps. Il est donc nécessaire de concevoir des outils statistiques

avancés pour l'étude de la structure temporelle des signaux. La conception de ces outils doit venir d'un effort interdisciplinaire, afin de rassembler les compétences en apprentissage statistique, en traitement du signal et en reconnaissance de motifs autour d'experts de ces signaux, des professionnels du marketing aux climatologues, en passant par les médecins.

Le domaine médical constitue un parfait exemple de milieu où la compréhension automatisée de signaux pourrait changer la donne et mener à de nombreuses applications. La motricité humaine est un processus très complexe, faisant intervenir de nombreux muscles qui doivent se coordonner entre eux. Différentes pathologies peuvent avoir un impact sur la capacité d'un patient à marcher. Les neurologues ou les spécialistes en ORL sont capables de détecter et de distinguer à l'oeil nu des différences subtiles dans la démarche du patient. Les neurologues peuvent en effet de diagnostiquer des neuropathies, tel le syndrome de Parkinson, en regardant un patient marcher. Ils observent la démarche à différents niveaux, de l'aisance globale du patient aux potentielles asymétries entre les côtés. Ensuite, ils regardent l'évolution des pas au cours de l'exercice, pour détecter si le patient se fatigue. Cette analyse permet d'extraire beaucoup d'informations qualitatives sur la condition du patient. Avec l'expérience, les spécialistes sont capables de diagnostiquer très rapidement les patients et de leur apporter les soins adéquats. Les capteurs inertiels permettent aujourd'hui d'enregistrer la marche des patients en consultation. La quantification des informations extraites par le docteur à partir de ces capteurs est un véritable challenge qui pourrait, à terme, changer la manière dont sont suivis les malades. La première étape pour s'attaquer à ce défi est de comprendre ces signaux. En effet, les intuitions du médecin sur le patient se transposent rarement en une propriété du signal. Il est donc nécessaire de pouvoir représenter le signal de manière interprétable, afin que les experts puissent transposer leurs connaissances du phénomène sur des caractéristiques du signal.

Au cours de ma thèse, j'ai collaboré avec Cognac-G, une équipe de recherche regroupant des chercheurs en apprentissage statistique et des chercheurs cliniciens, dans le but de quantifier le comportement humain ou animal. Dans ce but, plusieurs protocoles ont été définis, sur un large champs d'applications, de la respiration des souris ou la locomotion humaine au mouvements des yeux chez le nourrisson. Ces protocoles doivent permettre de quantifier objectivement les phénomènes d'intérêt grâce à des capteurs, qui enregistrent des séries temporelles univariées ou multi-variées, aussi appelées signaux physiologiques. Deux exemples connus de ce genre de signaux sont les électrocardiogrammes (ECG) pour l'activité du coeur et les électroencéphalogrammes (EEG) pour celle du cerveau. La premier défi de ces études réside dans l'extraction des informations d'intérêt à partir de tels signaux, afin de les interpréter et de comprendre les mécanismes biologiques, physiologiques ou biomécaniques qui les produisent. Le second défi est d'automatiser ce processus de quantification afin de développer des outils qui pourront être utilisés par les médecins pour le suivi longitudinal et la comparaison entre leurs patients.

L'objectif de cette thèse est de concevoir et d'étudier des outils statistiques de comparaison des séries temporelles capables de répondre à ces deux défis.

1.1.2 Comparer des signaux temporels : quantification des différences et interprétabilité

Absence de distance canonique entre signaux

Pour les données vectorielles, la plupart des distances utilisent des comparaisons terme à terme entre les points, comme par exemple la distance euclidienne. Ces mesures de similarité ne sont pas adaptées pour les signaux temporels, du fait de problèmes d'alignement entre les échantillons temporels des différentes séries. Lorsque celles-ci ne sont pas collectées dans un environnement extrêmement contrôlé, leur longueur et le décalage qui peut exister entre elles peuvent beaucoup varier, ce qui rend difficile le problème de recalage entre les séries.

La technique de la déformation temporelle dynamique, en anglais *Dynamique Time Wrapping* (DTW), introduite par Sakoe & Chiba (1971) permet de calculer un alignement entre deux signaux. Elle est basée sur la comparaison des différents échantillons temporels par une distance vectorielle. L'avantage de cette distance est que les séries sont alignées automatiquement et qu'il est possible de comparer des signaux de longueurs différentes. La DTW utilise la programmation dynamique et les équations de Bellman (1952) pour calculer l'alignement qui minimise la distance entre les deux séries. Cet alignement est lié à une distance entre les signaux, qui peut être utilisée pour étendre des méthodes vectorielles aux séries temporelles comme les k plus proches voisins ou le classifieur SVM. Des relaxations de la DTW basées sur des modifications continues des équations de Bellman, appelées soft-DTW, ont été développées pour définir des distances lissées (Bahl & Jelinek, 1975) ou des noyaux (Saigo et al., 2004).

Cette classe de distance, basée sur le calcul d'un alignement entre les séries, est prometteuse car elle permet de résoudre le problème de l'alignement et offre une grande flexibilité. Cependant, le coût de calcul de la distance entre deux séries de longueur T_1 et T_2 est proportionnelle à leur produit $\mathcal{O}\left(T_1 T_2\right)$. Ces méthodes sont donc coûteuses à utiliser pour des séries temporelles longues. Cette distance n'est pas non plus différentiable, ce qui complique l'adaptation de la plupart des modèles vectoriels à celle-ci. Sur ce dernier point, les résultats récents de Cuturi & Blondel (2017) ouvrent de nouvelles pistes de recherche pour le traitement des signaux, en dérivant un algorithme de complexité $\mathcal{O}\left(T_1 T_2\right)$ pour calculer la dérivée de la soft-DTW.

De plus, le fait de considérer des signaux longs comme des vecteurs les placent dans un espace de très grande dimension. Or, lorsque la dimension croît, les distances terme à terme deviennent moins discriminantes. Les distances entre les points se concentrent, de par l'effet moyennant de la dimension. Il est donc nécessaire d'utiliser d'autres outils statistiques pour comparer des signaux. Les méthodes les plus communes peuvent être classées entre les méthodes basées sur l'extraction de propriétés, à partir de modèles, et les méthodes dites bout-à-bout. Nous décrivons ci-dessous ces deux approches.

Modèles basées sur l'extraction de propriétés

Une approche pour la comparaison entre signaux est de comparer des propriétés spécifiques extraites des signaux. Par exemple, la linéarité et la périodicité d'un signal peuvent être quantifiées et utilisées pour déterminer à quel point il est similaire à d'autres signaux. En utilisant ces propriétés globales, une sinusoïde est plus proche d'une autre sinusoïde de fréquence proche que d'un signal linéaire. La quantification de propriétés caractéristiques d'un signal peut résulter de plusieurs modèles du signal, des

outils du traitement du signal, comme les coefficients de Fourier, jusqu'aux modèles de signaux statistiques tel que les modèles ARMA. Ainsi, différentes propriétés du signal, ici appelées caractéristiques (*features* en anglais), peuvent être extraites et la distance entre deux signaux est calculée en utilisant une distance terme-à-terme entre celles-ci.

Un des challenges de ce type de comparaison est la quantification automatisée des caractéristiques pour un jeu de signaux. En effet, pour des signaux bruités ou des signaux non stationnaires ou hétérogènes, l'estimation des coefficients de Fourier ne donne pas des résultats stables et fiables. Il est souvent nécessaire d'utiliser des méthodes plus complexes. Les outils de traitement du signal ne sont pas conçus pour être utilisés sur des populations de signaux hétérogènes et beaucoup de ces outils nécessitent de régler manuellement les paramètres pour fonctionner sur tous les signaux du jeu de données. De même, pour les modèles de signaux statistiques, l'estimation des paramètres peut aussi être instable lorsque certains signaux ne vérifient pas les hypothèses du modèle. Cet effort d'automatisation de l'usage des méthodes de traitement et de modélisation des signaux sort de leur cadre d'application du fait des objectifs de généralisation et de robustesse que cela impose. L'adaptation d'une seule de ces méthodes à un usage statistique demande une bonne connaissance de celle-ci et des données sur lesquelles elle sera appliquée.

La nécessité de choisir *a priori* les propriétés à inclure dans les comparaisons est un autre inconvénient de ce type de méthode. Cette sélection doit être faite manuellement et a un fort impact sur les performances des modèles qui les utilisent. Dans la plupart des cas, les éléments de comparaison ne sont pas connus et doivent être conçus avec une méthode essai/erreur pour capturer les propriétés adaptées au problème considéré. Ce processus, appelé *feature engineering*, est long et nécessite souvent les connaissances d'un expert du type de données considéré pour avoir une intuition des caractéristiques d'intérêts. Un exemple de ce processus est donné dans le domaine du traitement de l'image, avec les descripteurs *Scale-Invariant Feature Transform* (SIFT). Ces descripteurs, développés par Lowe (1999) pour capturer les variations locales d'intensité dans une image de manière à être invariants aux rotations, aux translations et aux changements d'échelle, permettent de comparer aisément des sous-parties du signal et ont été utilisés avec succès pour des tâches de reconnaissance d'images. Ils ont ensuite été raffinés pendant près d'une décennie pour diverses applications, afin d'améliorer les performances sur chacune des tâches. Cependant, ces *features* ne peuvent pas être utilisés pour d'autres applications avec des données différentes comme les signaux audio. Et même pour les images, le choix de caractéristiques peut dépendre de la tâche, et les descripteurs SIFT ne sont pas forcément les mieux adaptés. La multiplication des applications et le besoin de résoudre plusieurs problèmes à la fois rend le design de *features* peu pratique.

Lorsqu'aucune information sur les propriétés d'intérêt n'est disponible, il est possible d'utiliser une approche de *bag of features*. Cette technique est inspirée du modèle de sac de mots (Harris, 1954), utilisé en traitement du langage naturel (NLP) et a été utilisée avec succès avec des images par Qiu (2002). L'idée est d'inclure dans le modèle un large choix de caractéristiques communes et de procéder ensuite à une choix des *features* critiques pour le modèle avec des méthodes de sélection de variables comme par exemple le LASSO (Tibshirani, 1996) ou les méthodes de régression pénalisées avec des normes structurées telles que le Group LASSO (Yuan & Lin, 2006). Cette étape de sélection est cruciale pour garder une interprétabilité dans la comparaison. Elle permet en effet de ne conserver que les propriétés jugées importantes par le modèle pour résoudre la tâche.

Modèles bout-à-bout

Une autre approche efficace pour comparer des signaux est d'utiliser des méthodes dites bout-à-bout, ou *end-to-end* en anglais. Ces méthodes opèrent directement sur les signaux bruts et intègrent une partie qui calcule pour chaque donnée d'entrée une représentation interne, utilisée pour les comparaisons. Lors de la phase d'apprentissage, la représentation est entraînée en même temps que le modèle statistique pour résoudre la tâche. Les modèles bout-à-bout classiques sont les réseaux de neurones utilisés dans l'apprentissage profond (*cf.* Goodfellow et al. 2016 et références associées). Ces réseaux utilisent des représentations internes successives des données pour extraire l'information, et la dernière couche utilise la représentation finale pour résoudre la tâche considérée. Comme cette représentation est apprise en simultané avec le solveur, elle est adaptée pour résoudre le problème. Un autre exemple de technique bout-à-bout a été proposé par (Mairal et al., 2012) avec l'apprentissage d'un dictionnaire adapté à la tâche (en anglais *task-driven dictionary learning*). Dans leur article, les auteurs proposent d'apprendre une représentation des données, basée sur l'apprentissage d'un dictionnaire, en même temps qu'un modèle pour résoudre une tâche supervisée. Ce modèle est appris pour résoudre la tâche à partir de la représentation sur le dictionnaire, et les deux parties sont apprises conjointement. Ceci permet d'adapter la représentation des données au problème à résoudre, comme cela est fait dans le cas des réseaux de neurones.

Ces méthodes diffèrent des techniques basées sur l'extraction de propriétés car les propriétés comparées par les modèles bout-à-bout ne sont pas connues à priori mais apprises à partir des données, en association avec la tâche à résoudre. Cette co-adaptation de la représentation et du modèle statistique explique en grande partie le succès de ces méthodes. De plus, elles permettent d'éviter la phase coûteuse d'automatisation de l'extraction de propriétés dans les signaux, ce qui rend leur utilisation plus rapide et efficace. Cependant, le fait de ne pas connaître les propriétés comparées rend ces modèles moins interprétables. L'interprétation des représentations internes des réseaux de neurones est complexe. Cela rend la comparaison entre les signaux plus opaque. Ce problème est moins important pour les algorithmes d'apprentissage de dictionnaire. L'utilisation de la parcimonie dans leurs représentations permet de donner du sens à l'information extraite en termes d'analyse de motifs. Ainsi, les comparaisons qui utilisent ces représentations sont plus faciles à étudier.

Un autre inconvénient de ces techniques est qu'elles conduisent généralement à des problèmes d'optimisation non convexes. Les propriétés théoriques de ces modèles sont mal comprises et la convergence des algorithmes d'apprentissage n'est pas garantie. En pratique, ces modèles peuvent être entraînés à condition de disposer d'un jeu de données de très grande taille. En effet, contrairement aux modèles peu profonds, ces modèles nécessitent de grandes quantités de données pour l'apprentissage, et ils ne sont pas robustes aux erreurs d'étiquetage. Le manque de garanties théoriques pour ces modèles ne permet pas de quantifier ces besoins en exemples d'entraînement et peut donc être un frein à leur utilisation.

1.1.3 Donner du sens : représentations prédéfinies et dictionnaires empiriques

Une représentation est un moyen visuel de résumer un signal, dans le but de comprendre ses propriétés. Pour les modèles basés sur l'extraction de propriétés comme pour les modèles bout-à-bout, l'utilisation de représentations capables de mettre en

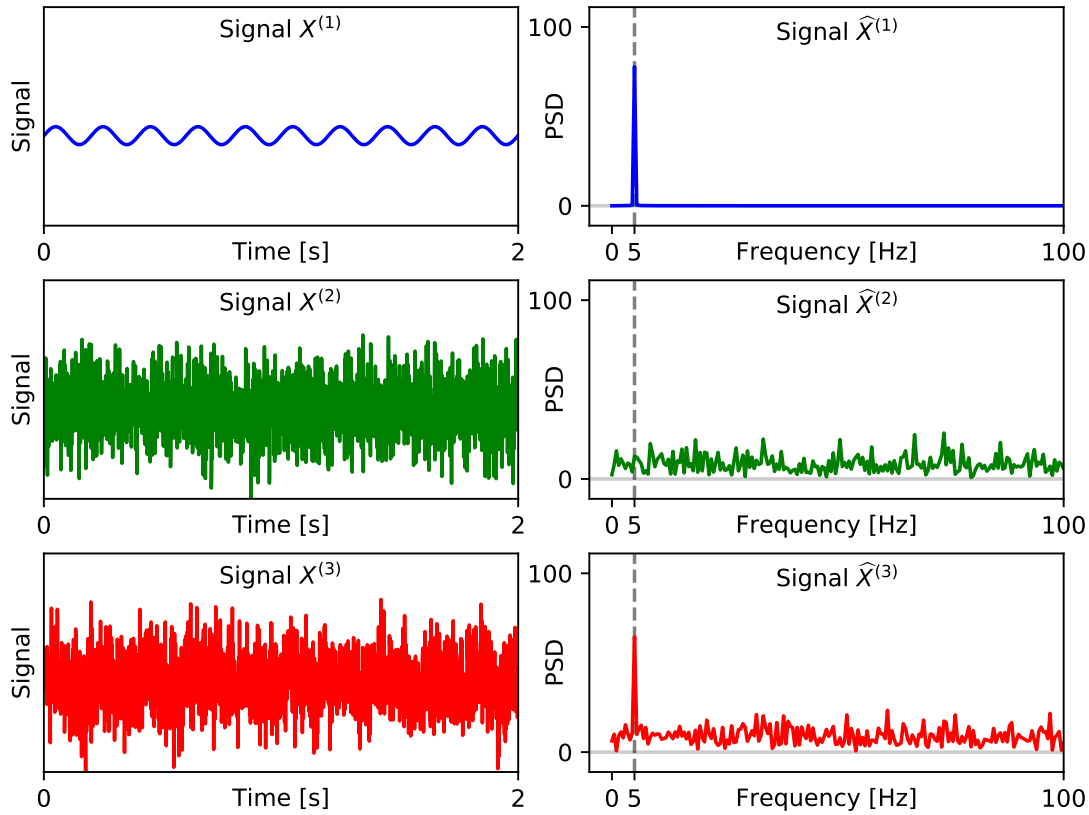


FIGURE 1.1: Comparaison de trois signaux avec une représentation temporelle (*gauche*) et une représentation de Fourier (*droite*). Avec la représentation temporelle, les signaux $X^{(2)}$ et $X^{(3)}$ semblent être plus proches mais dans le domaine fréquentiel, $X^{(3)}$ est aussi proche de $X^{(1)}$ et l'on peut voir que le troisième signal est la somme des deux premiers $X^{(3)} = X^{(1)} + X^{(2)}$.

avant les principales différences entre des classes de signaux est fondamentale. En effet, des représentations discriminantes permettent de sélectionner les caractéristiques utiles à extraire pour la comparaison des signaux. Pour les modèles bout-à-bout, ces représentations peuvent permettre d'interpréter le processus de décision. Nous décrivons ci-dessous différentes méthodes de représentation des signaux temporels.

Représentations globales

La représentation la plus commune est sans doute le tracé temporel des valeurs prises par le signal. Ce genre de représentation est utile, car très général, et l'on peut y détecter aisément certaines propriétés du signal, étant habitué à voir ces tracés. En effet, nombre de ses caractéristiques sont reconnaissables avec ces figures : la linéarité, la périodicité, la stationnarité, les formes récurrentes, les artefacts ou les ruptures par exemple. Les experts ayant l'habitude de ces visualisations peuvent en extraire des informations importantes, tels les cardiologues capables de diagnostiquer des maladies cardiaques à partir de l'analyse des relevés d'électrocardiogramme (ECG). Mais ces représentations canoniques sont moins faciles à analyser lorsqu'il n'y a pas de motifs clairs dans le signal, notamment en présence de bruit.

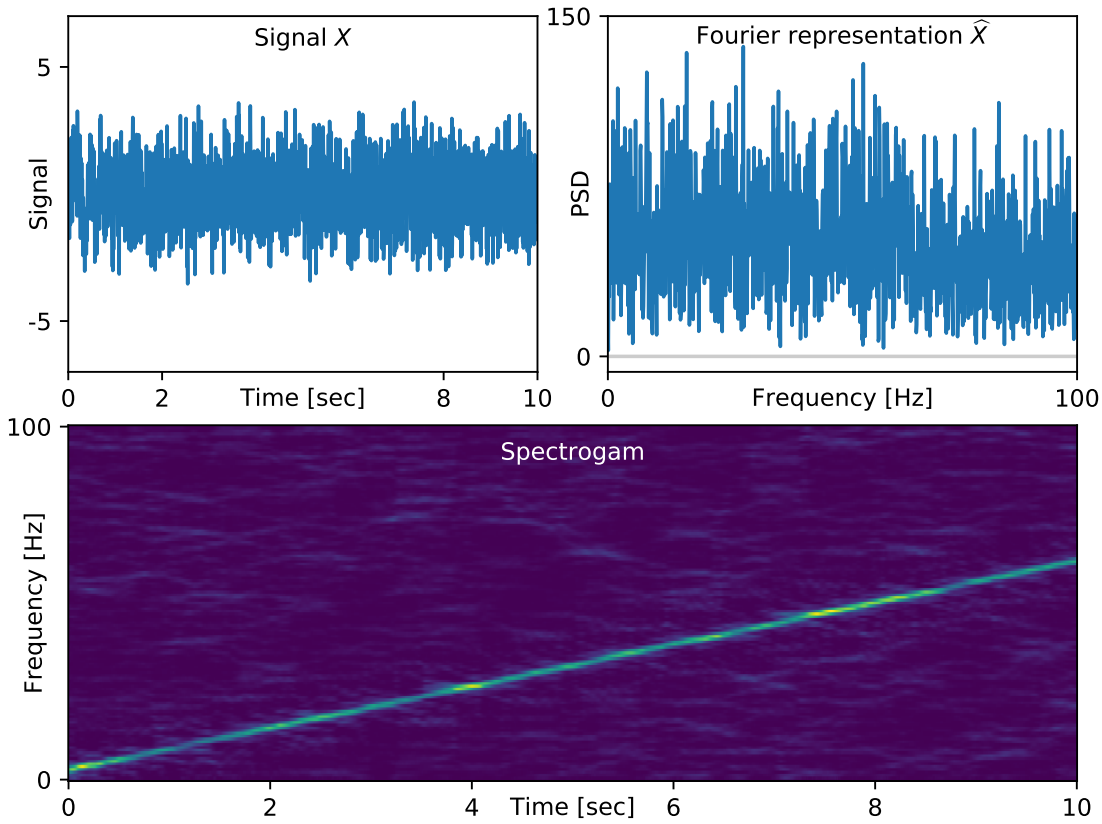


FIGURE 1.2: Différentes représentations pour le signal $X[t] = \sin((t+2)t/2) + \epsilon[t]$ avec ϵ un bruit blanc gaussien. (*En haut à gauche*) représentation temporelle, (*en haut à droite*) représentation de Fourier et (*en bas*) spectrogramme.

Une autre représentation très commune des signaux est leur spectre de Fourier. Cette représentation est liée aux propriétés harmoniques du signal et atténue l'effet du bruit s'il est indépendant. La Figure 1.1 présente un exemple de trois signaux représentés dans le domaine temporel et dans le domaine fréquentiel de Fourier. Les deux signaux bruités $X^{(2)}$ et $X^{(3)}$ semblent très proches en utilisant la représentation temporelle mais la représentation de Fourier montre que le signal $X^{(3)}$ a aussi une composante harmonique, de même fréquence que celle de $X^{(1)}$. La représentation de Fourier, on peut voir que le signal $X^{(3)}$ est la somme des signaux $X^{(1)}$ et $X^{(2)}$. Cet exemple montre l'importance de sélectionner correctement la représentation utilisée pour étudier des signaux, car celle-ci conditionne les propriétés des séries qu'il est possible de comparer. Ces deux représentations permettent de mettre en avant des caractéristiques globales des signaux et permettent donc de distinguer les signaux de manière globale.

Extraire la structure locale

Pour les signaux non stationnaires ou bruités, les propriétés globales ne permettent pas de distinctions précises entre les signaux et peuvent être difficiles à estimer. Par exemple, l'estimation du spectre de Fourier sur l'ensemble d'un signal non stationnaire n'est pas stable et il est compliqué d'extraire les harmoniques utiles. Pour ces séries, les informations sont contenues dans les structures locales du signal. L'extraction de ces structures ne peut se faire que par l'analyse locale du signal. Une extension naturelle de la représentation de Fourier pour l'analyse locale du signal a été proposée par Gabor

FIGURE 1.3: (*haut*) Représentation temporelle d'un signal d'accélérométrie le long de l'axe verticale pendant un exercice de marche. Les barres rouges verticales indiquent les emplacement des activations des pas et rendent compte de la régularité de la marche. (*bas*) Trois pas différents dans le signal : ces trois pas pourraient être résumés par un pattern unique, du fait de leur faible différence.

(1946), puis développée avec la transformée temps-fréquence (STFT). Cette analyse utilise la transformée de Fourier sur des sous-fenêtres du signal de départ. L'information n'est pas agrégée globalement mais présentée en fonction du temps et de la fréquence, ce qui permet de révéler les transitions dans le signal. La Figure 1.2 montre que cette représentation met en valeur des variations de la fréquence au cours du temps qui ne sont pas visibles avec des représentations globales. L'utilisation de méthodes globales sur des sous-portions du signal est une méthode populaire de représentation des signaux. Par exemple, les approximations linéaires par morceaux quantifient la linéarité de sous-segments du signal et réduise sa complexité (*cf.* Keogh et al. 2001 et références associées). La transformée en ondelettes est un autre exemple de représentation pour la structure locale du signal. L'analyse en ondelettes la plus courante représente le signal de manière parcimonieuse, en concentrant l'information autour des discontinuités (*cf.* Mallat 2008 et références associées). Le caractère multi-échelle de cette transformée permet de plus de mettre en avant des phénomènes de durée différente. Une version multi-couche de cette transformée a été proposé avec la transformée *scattering* (Mallat, 2012). Cependant, toutes ces représentations analysent des propriétés spécifiques du signal, connues à priori. L'analyse de Fourier révèle les propriétés harmoniques du signal, tandis que les approximations linéaires par morceaux quantifient sa linéarité. Lorsque l'on ne connaît rien de la structure du signal, la recherche d'une représentation discriminante doit être faite par tâtonnement.

Représentation par motifs

Pour les signaux dont on ne connaît pas la structure, l'adaptabilité de la méthode de représentation est cruciale. Une idée pour résumer un signal est d'extraire automatiquement les motifs récurrents dans ce signal. Les caractéristiques des structures locales sont alors apprises directement à partir des données, ce qui permet d'extraire des structures non-analytiques et non-connues *a priori*. Ces structures locales sont appelées motifs, en anglais *patterns*. Les représentations basées sur les motifs ont d'abord été développées pour les données vectorielles comme un moyen de réduire la variabilité des points et de réduire le bruit. Hotelling (1933) a proposé l'Analyse en Composantes Principales pour calculer les vecteurs qui expliquent le plus de variance possible dans les données observées. Ces composantes principales peuvent être vues comme des vecteurs qui représentent les motifs typiques dans les données. Une autre représentation vectorielle basée sur les motifs est l'algorithme de K-moyennes (Macqueen, 1967). Cette méthode assigne chaque point du jeu de données à un des K groupes de points et les points sont ensuite représentés par le barycentre de tous les éléments du groupe auquel ils appartiennent. Beaucoup d'autres techniques de réduction de dimension peuvent ainsi être analysées comme des techniques de représentation basées sur des motifs, notamment l'analyse en composantes indépendantes (ICA, Jutten & Herault 1991), ou la factorisation de matrice positive (NMF, Paatero & Tapper 1994). Olshausen & Field (1997) ont proposé l'apprentissage de dictionnaires parcimonieux, qui apprend des motifs, appelés atomes

d'un dictionnaire, à partir des données et les utilise pour encoder les signaux originaux. Cette méthode est un cadre très général pour l'apprentissage de motifs et peut être étendue aux signaux temporels.

Pour les signaux temporels, les motifs sont des sous-signaux typiques, qui peuvent être répétés dans le temps. La série peut alors être encodée par un signal d'activation de ces *patterns*, séparant ainsi les variations de la série et leur localisation dans le temps (Vautard & Ghil, 1989; Grosse et al., 2007). Le fait que seul un nombre limité de motifs soit utilisé permet de réduire la complexité de la représentation et l'utilisation d'activations parcimonieuses permet la localisation dans le temps de ces variations. Ce type de représentation est très naturel dans le contexte des signaux physiologiques comme les ECG, EEG, les mouvements des yeux ou l'accélération verticale du pied pendant la marche, comme présenté dans la Figure 1.3. Ces méthodes pour extraire des motifs des signaux ont été introduites en utilisant les intuitions provenant de l'apprentissage de dictionnaire pour des données vectorielles. Bien qu'elles n'aient pas fait l'objet de beaucoup d'études, ces méthodes ont montré de bons résultats sur des applications en traitement d'images et de la parole, du fait de leur adaptabilité et de leur interprétabilité. Le choix du design du dictionnaire permet de changer la taille et l'échelle des atomes, les variations de résolution permettent d'accéder à différents niveaux du signal. Une propriété très importante de ces représentations est la séparation entre les motifs et leur localisation dans le temps. La Figure 1.3 montre qu'il est plus facile d'étudier la régularité des pas à partir du signal d'activation en rouge pointillé qu'à partir du signal original en bleu, comme les variations sont résumées par un pattern unique, et les petites perturbations autour de ce pattern sont abandonnées.

1.2 Contributions de la thèse

1.2.1 Résumé des travaux

Au cours de mon doctorat, je me suis intéressé aux questions d'interprétabilité de la représentation des signaux temporels. Les représentations convolutives des signaux sont des méthodes qui permettent de représenter un signal de manière intuitive et interprétable. Cependant, ces méthodes ont un coût de calcul élevé et un grand nombre de paramètres qui les influencent. D'un autre coté, les réseaux de neurones sont très efficaces et résolvent en pratique beaucoup de tâches mais il est difficile d'interpréter les résultats obtenus. L'étude conjointe de ces deux classes de modèles et des liens qui peuvent exister entre elles permet d'amener de nouvelles perspectives pour combler les inconvénients de chacune de ces méthodes. Nous listons ici les contributions faites dans ce manuscrit et les publications associées.

Dans la Partie I, nous nous intéressons aux représentations convolutives et à l'amélioration de leur coût de calcul.

Chapitre 3 : Représentation convolutive. Les représentations convolutives sont utilisées pour modéliser des signaux en extrayant des motifs qui résument les variations locales du signal. Ces représentations sont particulièrement adaptées pour les signaux quasi-périodiques, comme les signaux physiologiques qui présentent souvent des motifs très marqués. Le Chapitre 3 présente ce modèle en détail, ainsi que sa version parcimonieuse.

Definition 1.1. *La représentation convolutive modélise un signal multivarié* $X \in X_T^P$ *comme la somme de* K *produits de convolution entre un motif multivarié* $\boldsymbol{D}_k \in \mathcal{X}_W^P$ *et un signal d'activation* $Z_k \in \mathcal{X}_L$*, avec* $L{=}T{-}W{+}1$*, tel que*

$$X[t] = \sum_{k=1}^{K} (Z_k * \boldsymbol{D}_k)[t] + \mathcal{E}[t], \qquad \forall t \in [\![0, T-1]\!] \ .$$

Le signal $\mathcal{E} \in \mathcal{X}_T^P$ *représente un terme de bruit additif, de même dimension que* X*.*

Nous décrivons ensuite les algorithmes de l'état de l'art pour calculer les motifs dans ces représentations ainsi que les coefficients associées.

Chapitre 4 : Interprétabilité de l'Analyse du Spectre Singulier. L'Analyse du Spectre Singulier, en anglais *Singular Spectrum Analysis* (SSA), est une technique utilisée pour l'analyse de signaux courts et bruités. Cette méthode extraie les sous-séries de longueur W du signal original et construit la matrice de W-trajectoires $\boldsymbol{X}^{(W)}$, dans laquelle chaque ligne est l'une des sous-séries extraites, *i.e.*,

$$\boldsymbol{X}^{(W)} = \begin{bmatrix} X[0] & X[1] & \dots & X[W-1] \\ X[1] & X[2] & \dots & X[W] \\ \dots & & & \\ X[T-W-1] & X[T-W] & \dots & X[T-1] \end{bmatrix} .$$

Cette matrice est ensuite réduite avec une décomposition en valeur singulière, en anglais *Singular Value Decomposition* (SVD), comme une somme de matrices de rang 1,

$$\boldsymbol{X}^{(W)} = \sum_{k=1}^{W} \lambda_k U_k V_k^{\mathsf{T}} \ ,$$

avec $U \in \mathbb{R}^{L \times K}$ unitaire et $V \in \mathcal{O}_K$. Pour chacune des matrices $\lambda_k U_k V_k^{\mathsf{T}}$, un signal $Y^{(k)} \in \mathcal{X}_T^1$ peut être reconstruit en prenant comme valeur au temps t la moyenne le long de la t-ième anti-diagonale de la matrice, *i.e.*

$$Y^{(k)}[t] = \frac{1}{W_t} \sum_{\tau=0}^{W_t-1} \left(\lambda_k U_k V_k^{\mathsf{T}}\right)_{\tau, t-\tau}$$

où $W_t = \min(t, T-t, W)$. Les signaux ainsi reconstruits sont liés à la tendance et la saisonnalité de la série étudiée. Afin d'améliorer l'interprétabilité de la décomposition, la SSA requiert une étape manuelle de groupement des composantes $Y^{(k)}$. Cette étape s'effectue en calculant une partition $\left\{ I_m \subset [\![1, W]\!] \ ; \ \cup_{m=1}^{M} I_m = [\![1, K]\!] \right\}$ et les composantes finales $\{C^{(m)}\}_{m=1}^{M}$ sont obtenues en sommant les composantes du groupe I_m, *i.e.*

$$C^{(m)} = \sum_{k \in I_m} Y^{(k)} \ .$$

Les contributions faites dans le Chapitre 4 sont les suivantes.

- Dans le Théorème 1.2, nous montrons que la décomposition ainsi obtenue peut être interprétée comme une représentation convolutive du signal et nous mettons en valeur les propriétés des motifs ainsi extraits. Nous notons pour $Y \in \mathcal{X}_T^P$ la

norme

$$\|Y\|_w^2 = \sum_{t=0}^{T-1} W_t \|Y[t]\|_2^2$$

où $W_t = \min(t, T-t, W)$.

Théorème 1.2. *Les K premiers triplets singuliers $\{\lambda_k, U_k, V_k\}_{k=1}^K$, calculés par la SSA pour le signal $X \in \mathcal{X}_T^1$ univarié, donnent une solution du problème d'optimisation suivant,*

$$(Z^*, \boldsymbol{D}^*) = \underset{Z \in \mathcal{X}_L^K, \{\boldsymbol{D}_k\}_{k=1}^K \subset \mathcal{X}_W^1}{\operatorname{argmin}} \left\| X - \sum_{k=1}^K Z_k * \boldsymbol{D}_k \right\|_w^2$$

où les atomes dans $\boldsymbol{D} = \{\boldsymbol{D}_k\}_{k=1}^K$ forment une famille orthonormale.
Les valeurs de $\boldsymbol{D}^*$ et Z^* sont données par $\boldsymbol{D}_k^*[t] = V_{k,t}$ et $Z_k^*[t] = \lambda_k U_{k,t}$.

- Nous présentons un cadre unifié pour l'automatisation de l'étape de groupement des composantes. Les stratégies de regroupement peuvent ainsi être décrites en trois phases :

  1. Sélection des composantes de la SSA non liées au bruit.
  2. Formation d'une matrice d'adjacence entre ces composantes.
  3. Création des groupes I_m de composantes adjacentes.

- Nous proposons deux nouvelles mesures de similarité, **wCG** and **HSG**, dans le but de calculer la matrice d'adjacence pour le groupement, ainsi qu'une une nouvelle méthode de formation des groupes, **HM**, basées sur l'importance de chaque composante. Ces méthodes nouvelles sont ensuite comparées aux méthodes proposées dans la littérature sur des signaux simulés.

Chapitre 5 : Codage Parcimonieux Convolutif Distribué (DICOD). Dans le Chapitre 5, nous nous intéressons à l'approche gloutonne de la descente par coordonnée pour résoudre le problème de représentation convolutive parcimonieuse suivant,

$$Z^* = \underset{Z \in \mathcal{X}_L^K}{\operatorname{argmin}} \frac{1}{2} \left\| X - \sum_{k=1}^K Z_k * \boldsymbol{D}_k \right\|_2^2 + \lambda \left\| Z \right\|_1 \quad , \tag{1.1}$$

pour un signal $X \in \mathcal{X}_T^P$ et un dictionnaire $\boldsymbol{D} = \{\boldsymbol{D}_k\}_{k=1}^K \subset \mathcal{X}_W^P$ fixés. Lorsque tous les coefficients de Z sont fixés sauf le coefficient (k_0, t_0), ce problème a une solution explicite, notée $Z'_{k_0}[t_0]$. À chaque itération, la descente gloutonne par coordonnée choisit une coordonnée (k_0, t_0) telle que

$$(k_0, t_0) = \underset{(k,t) \in [\![1,K]\!] \times [\![0,T-1]\!]}{\operatorname{argmax}} \left| Z_k[t] - Z'_k[t] \right|$$

et la valeur de cette coordonnée (k_0, t_0) de Z est mise à jour à $Z'_{k_0}[t_0]$. Ces mises à jour peuvent être calculées efficacement en maintenant une variable auxiliaire $\beta \in \mathcal{X}_L^K$, moyennant une complexité numérique de l'ordre de $\mathcal{O}\left(KW\right)$. Nos contributions dans le Chapitre 5 sont les suivantes.

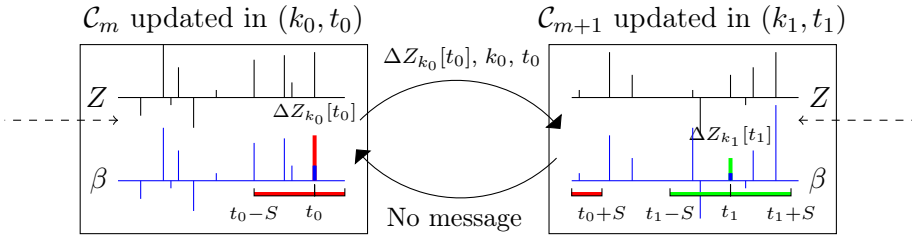


FIGURE 1.4: Illustration de l'algorithme DICOD. Le processeur $m+1$ met à jour $(k_1, t_1) \in \mathcal{C}_{m+1}$ indépendamment des autres processeur car le coefficient est hors de la zone d'interférences, c'est à dire qu'il ne change rien pour les valeurs optimales des coefficients hors du segment de $\mathcal{C}_{m+1}$. Le processeur m met à jour le coefficient $(k_0, t_0) \in \mathcal{C}_m$ qui est dans la zone d'interférence $[\![mL_M - S, mL_M]\!]$. Ainsi, il doit notifier le processeur voisin $m+1$ de la mise à jour, en lui envoyant un message contenant la valeur de la mise à jour $\Delta Z_{k_0}[t_0]$ et les coordonnées de celle-ci. Lorsque le processeur $m+1$ reçoit ce message, il met à jour sa version de β pour prendre en compte la mise à jour.

Algorithm 1.1 DICOD$_M$

Entrée : Dictionnaire $\boldsymbol{D} \subset \mathcal{X}_W^P$, signal $X \in \mathcal{X}_T^P$, paramètre $\epsilon > 0$
En parallèle pour $m = 1 \cdots M$
Pour tout (k, t) in $\mathcal{C}_m$, initialiser $\beta_k[t]$ et $Z_k[t]$
Répéter
  Prendre en compte les messages reçus pour mettre à jour β si besoin.
  Pour tout $(k, t) \in \mathcal{C}_m$, calculer $Z'_k[t]$ à partir de β
  Choisir $(k_0, t_0) = \underset{(k,t) \in \mathcal{C}_m}{\operatorname{argmax}} |\Delta Z_k[t]|$ $\qquad \left(\Delta Z_k[t] = Z_k[t] - Z'_k[t]\right)$
  Mettre à jour β pour tout $(k, t) \in [\![1, K]\!] \times [\![t_0 - W, t_0 + W]\!]$
  Mettre à jour la solution : $Z_{k_0}[t_0] \leftarrow Z'_{k_0}[t_0]$
  Si $t_0 - mL_M < W$ **alors**
    Envoyer $(k_0, t_0, \Delta Z_{k_0}[t_0])$ au processeur $m-1$
  Si $(m+1)L_M - t_0 < W$ **alor**
    Envoyer $(k_0, t_0, \Delta Z_{k_0}[t_0])$ au processeur $m+1$
jusqu'à ce que $|\Delta Z_{k_0}[t_0]| < \epsilon$ pour tous les coeurs
Retourner Z

- Nous proposons l'algorithme distribué DICOD (Algorithme 1.1), pour résoudre le problème du codage parcimonieux convolutif. Cet algorithme, basé sur la descente gloutonne par coordonnée, est asynchrone et efficace en terme de communications. Il repose sur un découpage du signal $X \in \mathcal{X}_T^P$ en segments continus $\mathcal{C}_m = \{(k, t) \in [\![1, K]\!] \times [\![mT/M, (m+1)T/M]\!]\}$. Lorsque des coordonnées sont mises à jour sur le bord d'un segment, le processeur voisin doit être notifié. La Figure 1.4 illustre le schéma de communication envisagé dans DICOD.
- Nous décrivons également un algorithme séquentiel, appelé SeqDICOD, qui opère les mêmes mises à jour que DICOD, mais de manière séquentielle. Les mises à jour sont donc localement gloutonne, ce qui réduit la complexité de calcul des itérations par rapport à l'algorithme glouton classique.
- Nous prouvons la convergence de l'algorithme DICOD distribué sous des hypothèses faibles.

FIGURE 1.5: (*Gauche*) Évolution de l'objectif en fonction du temps pour un long signal (120000 échantillons temporels). (*Droite*)Évolution de l'accélération de DICOD par rapport à CD en fonction du nombre de coeurs.

Théorème 1.3. *Si les hypothèses suivantes sont vérifiées :*

H1. *Les corrélations entre les atomes du dictionnaire sont strictement inférieures à 1 ;*

H2. *Les processeurs ne sont pas arrétés avant que la solution sur leur segment soit localement optimale ;*

H3. *Le délai de communication entre les coeurs est inférieur au temps de calcul d'une mise à jour,*

alors, l'algorithme DICOD converge vers la solution optimale Z^* du problème (1.1).

- Nous démontrons que l'accélération de DICOD_M distribué avec M coeurs par rapport à la version séquentielle SeqDICOD_M est sous-linéaire avec le nombre de coeurs utilisés mais l'accélération par rapport à la descente gloutonne par coordonnée (CD) est super-linéaire,

  Corollaire 1.4. *Soit $\alpha = W/T$ et $M \in \mathbb{N}^*$ le nombre de coeurs utilisés pour DICOD. Si $\alpha M < 1/4$ et si les coefficients non nuls du signal d'activation Z sont distribués de manière uniforme dans le temps, alors $\bar{A}$, l'espérance de l'accélération de DICOD_M par rapport à CD a pour borne inférieure quand α tend vers 0,*

$$\bar{A} \underset{\alpha \to 0}{\gtrsim} M^2(1 - 2\alpha^2 M^2 + \mathcal{O}(\alpha^4 M^4)) \ .$$

- Nous illustrons enfin les performances numériques de notre algorithme, distribué et non distribué, ainsi que son accélération sur de longs signaux. Dans la partie droite de la Figure 1.5, DICOD et SeqDICOD sont comparés aux algorithmes de l'état de l'art pour résoudre le problème (1.1). Ces différents algorithmes sont décrits dans la Section 3.3. La partie gauche de la Figure 1.5 confirme que l'accélération de DICOD est super-linéaire par rapport à la descente gloutonne par coordonnée.

Ensuite, dans la Partie II, nous étudions certains modèles d'apprentissage profond et leurs représentations internes dans le but d'améliorer leur interprétabilité.

Chapitre 6 : Interprétabilité des réseaux de neurones profonds. Les réseaux de neurones se sont imposés dans de nombreux domaines qui nécessitent de comparer des signaux, comme le traitement audio ou la reconnaissance d'image. Cependant, ces techniques sont souvent vues comme des boîtes noires, qui ne permettent pas de comprendre le mécanisme de décision. Ce manque d'interprétabilité vient notamment du fait qu'il est difficile d'étudier leurs représentations internes. Bien que chaque fonction définissant une couche soit simple, il est difficile de comprendre leurs interactions. Le Chapitre 6 rappelle tout d'abord le cadre général de l'apprentissage profond ainsi que

FIGURE 1.6: Erreur d'entraînement (*haut*) et de test (*bas*) sur CIFAR10. La courbe en pointillés bleus montre l'entraînement régulier. La valeur de la courbe pleine rouge pour l'itération i montre l'erreur après $i - 100$ itérations normales, suivies de 100 itérations de post-entraînement. Le post-entraînement limite donc le sur-apprentissage, puisque l'erreur de test est inférieure bien que l'erreur d'apprentissage soit un peu plus élevée.

ses principaux résultats théoriques. Les réseaux de neurones sans rétroaction, en anglais *feedforward network*, se décrivent comme la composition de plusieurs fonctions simples

$$\Phi_{\boldsymbol{\mathcal{W}}} = \phi_L \circ \phi_{L-1} \circ \cdots \circ \phi_1$$

où chaque fonction $\phi_l : \mathcal{X}_l \to \mathcal{X}_{l+1}$ envoie la l-ième représentation interne sur la l+1-ième et où $\boldsymbol{\mathcal{W}}$ correspond aux paramètres du modèle. Lorsqu'une couche est linéaire, ϕ_l peut être réécrit

$$\phi_l : x \mapsto \psi_l \left(W_l x \right) \ ,$$

où ψ_l est une fonction d'activation fixée. La suite du chapitre est dédiée à une revue des résultats récents sur l'interprétabilité de ces réseaux.

Chapitre 7 : Post-entraînement pour l'apprentissage profond. En utilisant l'idée d'interpréter les représentations internes dans les réseaux de neurones profonds, le Chapitre 7 propose de revisiter l'apprentissage des modèles bout-à-bout. Lorsque la dernière couche d'un réseau de neurones est linéaire, le problème d'apprentissage du réseau peut se réécrire

$$\min_{\Phi_{L-1}, \boldsymbol{W}_L} \mathbb{E}_{(x,y)\sim\mathcal{P}} \left[\ell \left(\psi_L \left(\boldsymbol{W}_L \Phi_{L-1}(x) \right), \ y \right) \right] \ .$$

où Φ_{L-1} est le réseau formé par les premières couches du réseau, ψ_L et $\boldsymbol{W}_L$ sont respectivement la fonction d'activation et le vecteur de poids de la dernière couche, ℓ est la fonction de coût de la tâche à résoudre et $\mathcal{P}$ est la distribution d'entrée. Lors de la phase d'entraînement d'un réseau, tous les poids sont mis à jour simultanément avec une estimation du gradient sur un ensemble d'entraînement. Pour les modèles bout-à-bout, cela permet de co-adapter la représentation interne du modèle, apprise par les premières couches du réseau Φ_{L-1}, et le modèle de résolution de la tâche, calculé dans la dernière couche avec $\boldsymbol{W}_L$. À la fin de la phase d'apprentissage, on considère que le réseau a appris une bonne représentation et un bon modèle pour résoudre la tâche. Dans le Chapitre 7, nous faisons les contributions suivantes.

- Nous proposons une étape supplémentaire pour l'apprentissage des réseaux de neurones, appelée le post-entraînement (*post-training* en anglais), où la représentation apprise au cours de l'entraînement est fixée et l'on optimise les poids de la dernière couche du réseau. Le problème est posé sous la forme

$$\boldsymbol{W}_L^* = \operatorname*{argmin}_{\boldsymbol{W}_L} \frac{1}{N} \sum_{i=1}^{N} \ell \left(\psi_L \left(\boldsymbol{W}_L \Phi_{L-1}(x_i) \right), y_i \right) + \lambda \|\boldsymbol{W}_L\|_2^2 \ , \tag{1.2}$$

  avec $\lambda > 0$ un paramètre de régularisation. Dans ce problème, Φ_{L-1} est fixé et on cherche uniquement les meilleurs paramètres possibles $\boldsymbol{W}_L$ pour la dernière couche. Dans de nombreux cas, ce problème est fortement convexe et il est très peu coûteux en temps de calcul de le résoudre.

- Nous donnons une justification de notre méthode basée sur une interprétation des réseaux de neurones comme des méthodes à noyaux. Dans le problème (1.2), une régularisation ℓ_2 est ajoutée pour les paramètres $\boldsymbol{W}_L$. Cet ajout est proposé en interprétant Φ_{L-1} comme une carte topologique (*feature map* en anglais) pour les données d'entrée et en utilisant la théorie des méthodes à noyaux sur le calcul de la fonction minimisant le risque moyen dans l'espace de Hilbert à noyau reproduisant, en anglais *Reproducing Kernel Hilbert Space* (RKHS).
- Nous montrons que cette étape supplémentaire permet d'obtenir de manière consistante un gain de performance sur de multiples architectures, pour des réseaux convolutifs ou des réseaux récurrents, avec différents jeux de données. La Figure 1.6 montre le gain obtenue en utilisant notre méthode sur un réseau convolutif avec le jeu de données CIFAR10 (Krizhevsky, 2009).

Chapitre 8 : Comprendre le modèle ISTA appris. Pour un dictionnaire $D \in \mathbb{R}^{p\times K}$ et un vecteur $x \in \mathbb{R}^p$, le problème du LASSO est défini comme suit :

$$z^* = \underset{z\in\mathbb{R}^K}{\operatorname{argmin}} \underbrace{\|x - Dz\|_2^2 + \lambda\|z\|_1}_{F(z)} , \tag{1.3}$$

avec $\lambda > 0$ un paramètre de régularisation fixé. Des travaux récents ont montré qu'il était possible d'accélérer la résolution numérique de ce problème en utilisant un réseau de neurones entraîné pour estimer la solution (Gregor & Lecun, 2010; Sprechmann et al., 2012). Ces résultats se basent sur une vision de l'algorithme *Iterative Soft-Threshosholding* (ISTA, Daubechies et al. 2004) sous forme d'un réseau de neurones récurrent, qui peut être déplié K fois pour calculer K itérations de l'algorithme. Les résultats empiriques montrent que l'utilisation de K itérations de ISTA est moins efficace que le réseau à K couches entraîné pour résoudre le problème et que ce réseau, appelé LISTA, se généralise bien. La compréhension des mécanismes qui expliquent ce résultat pourrait permettre de mettre en évidence le lien entre les modèles de représentation par dictionnaire et l'interprétabilité des représentations internes des réseaux de neurones. Dans le Chapitre 8, nos contributions sont les suivantes.

- Dans un premier temps, nous montrons qu'il est possible d'accélérer ISTA lorsque la matrice de Gram $B = D^\mathsf{T}D$ associée au problème admet une quasi-diagonalisation avec des espaces propres parcimonieux $B \approx A^\mathsf{T}SA$. Nous proposons une procédure basée sur des factorisations successives (A_q, S_q) de la matrice de Gram B pour résoudre le problème (1.3), telle que

$$z^{(q+1)} = A_q^\mathsf{T} \underset{u}{\operatorname{argmin}}\, Q_{S_q}\left(u, A_q z^{(q)} - S_q^{-1}A_q B(z^{(q)} - y)\right) , \tag{1.4}$$

  où $Q_S(v, w) := \frac{1}{2}(v-w)^\mathsf{T}S(v-w)+\lambda\|v\|_1$ et $y = (D^\mathsf{T}D)^{-1}D^\mathsf{T}x$. Le problème (1.4) est séparable et une solution être calculée explicitement. En particulier, si $A = \boldsymbol{I}_K$ et $S = \|B\|_2\boldsymbol{I}_K$, on retombe sur l'algorithme ISTA. Le Theorem 8.2 montre que cet algorithme généralisé a une vitesse de convergence de l'ordre de $\mathcal{O}\left(\frac{1}{q}\right)$, similaire à ISTA, avec un facteur constant qui peut être amélioré en fonction des factorisations utilisées.
- De plus, on peut montrer aisément que :

FIGURE 1.7: Performance de LISTA et de FacNet sur un problème générique (*gauche*) et sur un problème adverse (*droite*).

Proposition 1.5. *Pour une itération q donnée, si la matrice $R_q = A_q^\intercal S_q A_q - B$ est définie positive, et si $z^{(q+1)}$ est défini selon l'équation (1.4), alors nous avons*

$$F(z^{(q+1)}) - F(z^*) \leq \frac{1}{2}\|R_q\|\|z^{(q)} - z^*\|_2^2 + \delta_{A_q}\left(z^*\right) - \delta_{A_q}\left(z^{(q+1)}\right) , \qquad (1.5)$$

où $\delta_A(z) = \|Az\|_1 - \|Z\|_1$.

Nous démontrons ensuite dans le Theorem 8.7 que, sous certaines conditions, en moyenne sur les dictionnaires gaussiens normalisés (dit dictionnaires génériques), la borne Proposition 1.5 peut être meilleure pour une factorisation (A_q, S_q) de la matrice de Gram que pour ISTA. Ce théorème permet de montrer le corollaire suivant,

Corollaire 1.6 (Certificat d'accélération)**.** *Si la distribution d'entrée $\mathcal{P}$ et le paramètre de régularisation λ vérifient*

$$\frac{\lambda\sqrt{p}}{8} \leq \mathbb{E}_{z\sim\mathcal{P}}\left[\|z^*\|_1\right] ,$$

alors, pour toute résolution $\mathbb{E}_{z\sim\mathcal{P}}\left[\|z^{(q)} - z^*\|_2\right] = \epsilon > 0$ à l'itération q, la borne de la Proposition 1.5, pour notre algorithme basé sur les factorisations est meilleure que celle pour ISTA, en moyenne sur les dictionnaires génériques.

- Nous montrons ensuite que ces résultats sont suffisants pour expliquer l'accélération de ISTA, car l'algorithme basé sur les factorisations peut être réécrit comme une reparamétrisation de LISTA, nommée FacNet, avec des contraintes supplémentaires sur les poids. Cet algorithme est donc toujours moins efficace que LISTA. Cette observation est étayée par des expériences numériques, par exemple dans la Figure 1.7.
- Finalement, nous montrons que des exemples adverses, conçus pour empêcher l'accélération dans FacNet, ne permettent pas non plus d'accélération pour LISTA (*cf* la partie droite de la Figure 1.7). Ces résultats empiriques semblent montrer que notre analyse capture une partie du mécanisme à l'oeuvre dans l'accélération de ISTA par LISTA.

Par ailleurs, pendant toute la durée de mon doctorat, j'ai collaboré avec des médecins autour de la recherche clinique, par le développement d'outils pour les aider à analyser les signaux physiologiques. La collaboration a été centrée autour de deux projets : l'étude de la marche chez les adultes et l'étude des mouvements oculaires chez les enfants en bas âge. La Partie III du manuscrit présente les résultats d'application des méthodes développées dans les chapitres 3, 4 et 5 aux signaux physiologiques, ainsi que des résultats médicaux.

FIGURE 1.8: Représentation basée sur des gabarits de pas, extrait d'une base de 50 sujets sains. (*Haut*) Activations des motifs de pas au cours du signal. (*Bas*) Gabarits de pas extraits par apprentissage de dictionnaire convolutif.

Chapitre 9 : Étude de la marche. La quantification de la motricité chez l'humain à partir de capteurs inertiels présente un intérêt majeur pour le suivi des patients. Par définition, la marche est un mouvement répétitif, où le pas joue un rôle de composant atomique. L'extraction de la structure locale de la démarche permet d'étudier la régularité ou la symétrie du signal. Ainsi, la capacité à identifier et extraire les pas de manière robuste dans un signal de marche est cruciale pour l'analyse de la marche. Nos contributions dans le Chapitre 9 sont les suivantes.

- Nous utilisons les représentations convolutives, décrites dans le Chapitre 3 afin de résumer les exercices de marche sous forme d'un signal d'activation du pas et d'un gabarit de pas (*cf* Figure 1.8).
- Nous présentons un nouvel algorithme permettant de détecter les pas dans un signal de marche de manière robuste. Notre méthode se base sur la comparaison du signal avec des exemples de pas pour identifier la présence d'un pas. Notre algorithme détecte correctement les pas pour les patients sains comme les patients présentant des pathologies diverses et nous analysons l'effet de chacun des paramètres sur les performances de notre algorithme.
- Ces travaux ont été associés à l'analyse des signaux de marche dans plusieurs études médicales, comme Barrois et al. (2015) and Barrois et al. (2016).

Chapitre 10 : Étude des mouvements oculaires. La neuro-opthalmologie est l'étude des relations entre le système nerveux et le système oculaire. Dans ce contexte, l'étude des mouvements des yeux est très intéressante car elle permet de révéler les mécanismes de contrôle du système nerveux sur les yeux. Au cours de cette thèse, nous avons étudié le nystagmus, un mouvement oculaire pathologique chez les jeunes enfants. Ce type de mouvement peut être un symptôme de plusieurs maladies, qui peuvent être diagnostiquées lorsque le nystagmus est correctement identifié. Nous faisons les contributions suivantes dans le Chapitre 10.

- Nous montrons que la SSA peut être utilisée afin de pré-traiter les signaux oculométriques dans le but de séparer les mouvements liés au nystagmus des mouvements du regard.
- Nous proposons deux représentations des propriétés du nystagmus, qui peuvent aider le médecin afin de mieux caractériser le mouvement et d'affiner son diagnostique.
- Ces outils ont été utilisés pour trois études : une communication orale à la *Gordon Research Conference on Eye Movements* (Robert et al., 2015), une étude du nystagmus associé au gliome du nerf optique (Robert et al., 2016) et une étude sur le nystagmus chez les enfants atteints du syndrome de Down.

1.2.2 Développement *opensource*

Au cours de la deuxième et troisième année de mon doctorat, j'ai été impliqué dans un projet open-source, soutenu par le Centre for Data Science, financé par l'IDEX Paris-Saclay, ANR-11-IDEX-0003-02. L'objectif de ce projet était de fournir un backend pour la bibliothèque `joblib`. `joblib` est une bibliothèque python conçue pour paralléliser facilement les calculs scientifiques, car elle fournit une interface efficace pour le calcul simplement parallélisable, où chaque coeur peut effectuer des calculs indépendants et les résultats sont également renvoyés indépendamment.

En collaboration avec Olivier Grisel, j'ai développé `loky` pour fournir une implementation robuste et fonctionnelle, multi-plateforme et multi-version de la classe `ProcessPoolExecutor` de `concurrent.futures`. Cette librairie présente notamment :

- **Implémentation sans deadlock :** Une des principales préoccupations dans les bibliothèques standards `multiprocessing` et `concurrent.futures` sont les capacités du `Pool / Executor` à gérer les incidents dans les processus esclaves. Notre bibliothèque répare les blocages possibles et renvoie des erreurs significatives lorsqu'une erreur arrive dans la gestion des travaux.

- **Comportement de lancement cohérent :** Tous les processus sont lancés en utilisant fork / exec sur les systèmes POSIX. Cela garantit des interactions plus sûres avec les bibliothèques tierces.

- **Exécuteur résutilisable :** Notre bibliothèque propose une stratégie pour éviter de relancer un exécuteur complet à chaque fois. Une instance d'exécuteur peut être réutilisée (et redimensionnée dynamiquement si nécessaire) à travers les appels consécutifs pour éviter les opérations de lancement et d'arrêt répétées. Les processus de travail peuvent être arrêtés automatiquement après un délai configurable pour libérer les ressources du système lorsqu'ils ne sont pas utilisés.

- **Integration transparente de `cloudpickle` :** Cette intégration permet d'appeler en parallèle des fonctions et des expressions définies de manière interactive. Les voies de communication entre les processus peuvent aussi être reconfigurées simplement.

- **Plus besoin de `if __name__ == "__main__":` dans les scripts :** Grâce à l'utilisation de `cloudpickle` pour l'appel des fonctions définies dans le module `__main__`, il n'est pas nécessaire de protéger l'appel de fonctions parallèles.

1.3 Publications

L'ensemble des travaux présentés dans ce document ont donné lieu à diverses publications et communications :

- Moreau, T., Oudre, L., and Vayatis, N. Groupement automatique pour l ' analyse du spectre singulier. In *Proceedings of the Groupe de Recherche et d'Etudes en Traitement du Signal et des Images (GRETSI)*, 2015b

- Oudre, L., Moreau, T., Truong, C., Barrois-Müller, R., Dadashi, R., and Grégory, T. Détection de pas à partir de données d'accélérométrie. In *Proceedings of the Groupe de Recherche et d'Etudes en Traitement du Signal et des Images (GRETSI)*, Lyon, France, 2015

- Moreau, T., Oudre, L., and Vayatis, N. Distributed Convolutional Sparse Coding via Message Passing Interface (MPI). In *Proceedings of the NIPS Workshop on Nonparametric Methods for Large Scale Representation Learning*, 2015a

- Moreau, T. and Audiffren, J. Post Training in Deep Learning with Last Kernel. *arXiv preprint*, arXiv:1611(04499), 2016

- Moreau, T. and Bruna, J. Understanding Neural Sparse Coding with Matrix Factorization. In *Proceedings of the International Conference on Learning Representation (ICLR)*, 2017

- Moreau, T., Oudre, L., and Vayatis, N. Distributed Convolutional Sparse Coding. *arXiv preprint*, arXiv:1705(10087), 2017

- Barrois, R., Oudre, L., Moreau, T., Truong, C., Vayatis, N., Buffat, S., Yelnik, A., de Waele, C., Gregory, T., Laporte, S., and Others. Quantify osteoarthritis gait at the doctor's office: a simple pelvis accelerometer based method independent from footwear and aging. *Computer methods in biomechanics and biomedical engineering*, 18(Sup1):1880–1881, 2015

- Barrois, R., Gregory, T., Oudre, L., Moreau, T., Truong, C., Pulini, A. A., Vienne, A., Labourdette, C., Vayatis, N., Buffat, S., Yelnik, A., De Waele, C., Laporte, S., Vidal, P. P., and Ricard, D. An automated recording method in clinical consultation to rate the limp in lower limb osteoarthritis. *PLoS ONE*, 11(10):e0164975, 2016

- Robert, M., Contal, E., Moreau, T., Vayatis, N., and Vidal, P.-P. The Why and How of Recording Eye Movement from Very Early Childhood. Oral Presentation, Gordon Research Conference on Eye Movement, 2015

2
Introduction

"Any fool can know. The point is to understand."

– Albert Einstein

Contents

2.1 Thesis Motivations: Understanding Time Series

2.1.1 Continuous Monitoring with Temporal Signals

In the last decades, there has been an explosion of the number of sensors, which record more and more information about our environment, our activity and our behavior. When navigating the web, many companies record the visited URLs, the time spent on each page or even the movement of the mouse on these pages. In cities, sensors are installed to record quantities of information about the environment or the population. Public transit badges track the daily travels of millions of people each day and CCTV cameras record hours of video surveillance in the streets. After being recorded, all these data are stored in huge data centers around the world, but little information is extracted from it. These signals which trace the evolution of many aspects of our daily life are not well understood. For CCTV cameras, even if automated methods are able to extract the content of a scene, human intervention is still needed to understand it and to highlight potential threats. Indeed, the understanding of long multivariate signals is a very complex task. It requires real-time analysis of the events, characterization of normal behaviors and abnormalities detection. For all these tasks, studying the statistical properties of the recorded data is often not enough. The mean or the variance of temporal

series characteristics are often not sufficient to distinguish the differences between signals. There is a need for tools specially designed to understand the temporal structures in signals. The design of these tools is expected to come from an interdisciplinary effort as they shall combine techniques from machine learning, signal processing and pattern recognition augmented by the knowledge of experts, from marketing professionals, to climatologists or medical doctors.

The medical field is a good example of a domain where understanding the signals could lead to many applications. Human motion is a complex process, which requires the coordination of many muscles. Different pathologies impact the ability of a patient to walk and neurologists or ENT specialists are able to detect and distinguish these very subtle differences in gaits using their eyes. For instance, a neurologist is able to diagnose neural disorders – such as the Parkinson syndrome – by looking at a patient walking. He looks at different levels of details, from the easiness in the motion to the potential asymmetries between the sides. Finally, he follows the gait evolution during the exercise to quantify if the patient is getting tired. With this multi-scale observation, the doctor gets qualitative information about the patient condition. A trained specialist is able to judge a patient very quickly and to provide adequate care. With the development of inertial sensors, it is now possible to record the gait of patients walking. Being able to quantify the information perceived by the doctor using these recorded signals is a real challenge. The first step to tackle this challenge is to understand the signal. Intuitions of the doctors can rarely be transposed directly as signal properties. To allow the experts to find relevant characteristics of the signal based on their knowledge and understanding of the studied phenomenon, it is necessary to use comprehensive representations of the signal which highlights some parts related to physical movement. Representations based on the localization of local patterns, such as steps, are interpretable in the sense that the regularity of the representation can directly be linked to the regularity of the walk of the subject.

During my PhD, I have collaborated with Cognac-G, a research team that brings together machine learning researchers and medical doctors, to study the quantification of human and animal behavior. To that aim, several experimental protocols have been defined for a wide range of clinical problems from mice breathing or human locomotion to young infant eye movements. Each protocol provides an objective quantification of the phenomenon of interest through the use of sensors, that record univariate or multivariate time series known as physiological signals. Electrocardiogram (ECG) for the heart or electroencephalogram (EEG) for the brain are well known examples of such signals. The first challenge consists in extracting relevant information from these signals, in order to interpret them and to help understand the physiological, biological or bio-mechanical mechanisms that produced them. The second challenge is to automate the quantification process in order to provide tools that can be used by doctors for the longitudinal follow-up and the inter-individual comparison of their patients.

The aim of this thesis is to provide and study statistical tools that answer both these challenges.

2.1.2 Comparing Time Series: Discrepancy Quantification and Interpretability

No Natural Distance for Time Series

For vector data, most distances use coordinate-wise comparison of the data samples, as it is the case for the Euclidean distance. These similarity measures are not well suited for time series as there is no natural way to match the time samples from one series to another. Indeed, if the time series result from measurements acquired with setups that are not very constrained, the length and phase of the sequences are subject to many variations. This makes it unclear which time samples should be compared together.

The dynamic time wrapping technique, introduced by Sakoe & Chiba (1971) computes the best alignment between two signals, based on a given metric to compare each time sample. The advantage of this distance is that it automatically aligns the compared time series. It is also possible to compare signals with different length. DTW relies on dynamic programming and the Bellman's equations (Bellman, 1952) to compute efficiently the alignment which minimizes the discrepancy between the series. The minimum value defines a distance between signals which can be used to extend vector methods like the k-nearest neighbors or the SVM classifier. Also, relaxations of this problem – using smoothed Bellman's equations – have been proposed to define smoothed distances (Bahl & Jelinek, 1975) or kernels (Saigo et al., 2004).

This class of distances, based on the computation of an alignment between signals, can be used to extend classic machine learning technique to time series. However, the computational cost of the distance between two signals of length T_1 and T_2 is proportional to their product *i.e.* $\mathcal{O}\left(T_1 T_2\right)$. This cost becomes expensive for long series and hinders the usage of DTW. Also, as this distance is not differentiable, some vectorial distances cannot be adapted to use it. This motivated recent work by Cuturi & Blondel (2017) which proposed an algorithm to compute the gradient of the soft-DTW with the same complexity as the distance computation $\mathcal{O}\left(T_1 T_2\right)$, opening interesting research opportunities.

Another challenge is that time series can be very long. Considering them as vectors embeds them in a high-dimensional space. As the dimension increases, the pair-wise comparison distances between samples gets less informative as all points tend to be at the same distance, due to the averaging effect of the dimension. This observation, called curse of dimensionality, makes it less effective to compare long time series with time wise distance, even when it is possible to align them. Thus, it is necessary to use statistical tools specifically designed to compare signals. The existing approaches can be classified in two categories. The first approach is to compare the signals using properties that are chosen *a priori*. These methods, called here property-based comparison, are very interpretable as we know the compared properties. The discrepancy between two signals can be linked to different characteristics of these signals. The second approach is to use end-to-end models, such as neural networks. These models take the raw signal as an input and integrate the extraction of information in the model. Here, the comparison is based on unknown properties, which are chosen directly from the data. This type of models tends to have better performances in practice on given tasks but are less interpretable, as the compared properties are unknown.

Property-based Comparisons

A high level approach to signal comparison is to compare specific properties extracted from the signals. For instance, the periodicity or linearity of signals can be quantified and used to determine how similar they are to each other. Using these two global properties, a waveform will be closer to another waveform with similar frequency than to a linear signal. Many properties of the signal can be quantified and the distance between two signals is computed by comparing these features using property-wise distances. Examples of such simple features include the Fourier coefficients, the linear regression coefficients or global statistics such as the mean and variance of the signal.

More advance signal models such as Linear Dynamic Systems (LDS) or Hidden Markov Models (HMM) can also be used to compare time series. The model parameters estimated for a signal can be used as quantifiers of signal properties. While these models are different from classical feature extractors, they also quantify properties chosen in a data-agnostic setting and can thus be considered as property-based comparisons. In this case, the metric used to compare the model parameter needs to be adapted. For instance the parameter-space for LDS can be non-euclidean and the distance need to be carefully selected. The definition of suitable distances for such space has been studied since the 70s using the Riemannian framework (Krishnaprasad, 1983). Recently, computationally efficient approaches have been proposed. The Martin distance (Martin, 2000) is an algebraic metric between two processes which can be linked to subspace angles between the models (De Cock & De Moor, 2002). Other approaches in the machine learning communities have been proposed to compare LDS such as the Binet-Cauchy kernels (Vishwanathan et al., 2007), the alignment distance (Afsari et al., 2012) and the KL-divergence (Chan & Vasconcelos, 2005).

One challenging problem for the property-based comparison is to be able to reliably quantify the selected properties for the studied set of signals. Indeed, the estimation of global features for a set of signals can be unstable, especially when these signals are non-stationary, noisy or heterogeneous. Signal processing tools are rarely designed to handle heterogeneous populations of signals which appear in many applications and the selection of the correct parameters requires manual tweaking to work on all signals in the database. For statistical time series models, like ARMA, the presence of outliers in the set can also lead to unstable results, as they do not verify the model's hypothesis. The automatization of the property extraction can require domain knowledge and a lot of engineering.

Another challenge of this method is the selection *a priori* of the properties to include in the comparison. This decision has to be done manually and is critical for the performance of the statistical models based on these comparisons. In many cases, the comparison elements are unknown and the extracted properties are designed using some trial and error methods to find the features that best capture the intrinsic properties necessary for the considered task. This process, called *feature engineering*, can be a very long and tedious process and requires expert knowledge to get an intuition about the critical properties that need to be quantified.

For activity recognition from inertial sensors recordings, certain features have been developed such as measurements in certain frequency bands of the signals (Oudre et al., 2012). These features are specifically validated to distinguish the signals from a human walking from a human running or biking. Thus, they cannot be reused to classify the same type of data for different activity such as swimming or rowing without a

novel validation step. A better known example can be found in image processing. Local descriptors – such as the Scale-Invariant Feature Transform (SIFT; Lowe 1999) or the Histogram of Gradient (HOG; Dalal & Triggs 2005) – have been designed to measure the local variation of the intensity in the image in order to be invariant to specific image transformations. Using these metrics, it is possible to define similarity measures between local patches in images for object recognition. These features have been refined over the decade following their development by multiple advances in the fields – such as GLOH (Mikolajczyk et al., 2005), SURF (Bay et al., 2008) or GF-HOG (Hu et al., 2010). But these features, which are well suited for images, are not directly usable for other applications with non image signals. While some can be adapted for new usages – such as 3D-SURF developed for 3D data (Knopp et al., 2010) or image descriptors applied to time-frequency representations of audio signals (Zhu et al., 2010; Rakotomamonjy & Gasso, 2015) – the resulting features must be validated as new features. Even for the same kind of signals, the choice of features can be task-dependent. As the feature selection and validation is time consuming, property-based comparisons are less practical than methods which automatically select the comparison properties.

A possible solution to select characteristic properties when there is no insight on the properties of interest is to use the *bag-of-features* approach. This technique is inspired by the bag-of-word model (Harris, 1954) used in Natural Language Processing (NLP) and has been used for instance with images in Qiu (2002). The idea is to quantify the largest set of properties possible, and then proceed to feature selection with methods such as LASSO (Tibshirani, 1996). Here, the properties can be selected in a supervised setting, to be specifically adapted to the considered task, or in an unsupervised setting to capture enough information in the signals to be able to generate them. The selection process is critical for the interpretability of the comparison, as it selects few properties that are judged important for the task at hand. It relies on the same principles as the property-based comparison but does not manually select the features. This technique has been extended in the context of activity recognition in videos with the bag of spatio-temporal features (Schüldt et al., 2004).

End-to-end Models

Another efficient technique to compare signals is to design end-to-end models. This type of model is operating directly on raw signals and integrates a part that computes a representation for each data sample. During the training phase, the representation is learned simultaneously with the parameters for the model solving the task. Typical end-to-end models are the deep neural networks (see Goodfellow et al. 2016 and references therein). Neural networks compute successive internal representations of the data, which are then used by the last layer to solve the considered task. As these representations are learned simultaneously with the model parameters, they are adapted to the specific task to be solved. Another example of end-to-end technique has been proposed with the task-driven dictionary learning (Mairal et al., 2012). In their paper, the authors propose to learn a data-representation, based on a learned dictionary, simultaneously with a statistical model to solve the considered task, based on the learned representation. This approach adapts the representation to the problem we try to solve, as with neural network models. Other works also proposed to learned discriminative dictionary, adapted for the considered task such as Mairal et al. (2008), Zhang & Li (2010) and Jiang et al. (2011).

Figure 2.1: ECG signal for a human under anesthesia. (*top*) raw signal and manual annotation for the patterns localisation. (*bottom left*) extracted patterns (*bottom right*) mean pattern. The mean pattern is repeated almost exactly in the signal.

Figure 2.2: Convolutional dictionary learned from a accelerometer signal recorded during the walk of a human subject. (*top*) raw signal. (*bottom*) learned dictionary atoms. The elements learned with the convolutional dictionary learning can be interpreted as step patterns.

Figure 2.3: Convolutional dictionary learned with a eye position signal recorded from an infant with pathological nystagmus movement. (*top*) raw signal. (*bottom*) learned dictionary atoms. The elements learned with the convolutional dictionary learning can be linked to nystagmus movements, with upward slow phase and downward saccades.

These methods differ from the property-based comparison because the properties compared by these methods are not known *a priori* but learned from the data, in association with the considered task. The co-adaptation of the representation and the statistical model is one big factor in the success of such methods. Also, removing the need to automate property quantification allows faster and more efficient practical applications. But because the compared properties are unknown, these models tend to lose interpretability. Neural networks are not easily interpretable in the sense that the internal representations they compute are hard to link to signal properties in the original space even though recent works propose feature visualization methods (Olah et al., 2017; Montavon et al., 2018). Dictionary learning algorithms can have better interpretability, in particular for physiological signals. These signals are often composed of repeated patterns. For instance, electrocardiogram signals (ECG) can have very regular patterns, as presented in Figure 2.1. These patterns have a medical signification as they are linked to different phases of an heartbeat. In the figure, starting positions of the heartbeats are manually annotated and the bottom left part presents all the extracted patterns. We can see that the variation between the extracted patterns of heartbeat and the mean pattern in the bottom right part of the figure are small. Convolutional dictionary learning techniques could be used to extract these patterns automatically and robustly, in order to facilitate the study of the heart-rate. For this simple example, the extraction of the repeated patterns seems trivial but it remains challenging for full signals which can be heterogeneous, have noise or trends which alter the patterns and have different amplitude. For other physiological signals such as accelerometer data recorded from a human walking in Figure 2.2 or the eye position of an infant with a nystagmus presented in Figure 2.3, the patterns learned with convolutional dictionary learning can be interpreted as specific movement from the body. Here, the task is more complex than with ECG as the shapes have more variations, which can be due to different phases in a recording or to pathologies. In this sense, the convolutional dictionary learning extracts interpretable representations for physiological signals as it learns patterns that can be linked to specific physical phenomenons. The representation of a signal on this set of learn patterns permits to naturally study the regularity of these phenomenons as well as their local variations.

Another issue with end-to-end techniques is that most optimization problems for statistical learning become non-convex when the representation and the statistical model based on it are jointly learned. The theoretical guarantees for these models are not well understood and they do not always converge, but recent works show that it is possible to guarantee the convergence of the training under certain conditions (Agarwal et al., 2013; Haeffele & Vidal, 2015, 2017). In practice, these models can be trained when the training set is large enough, with limited noise on the labels. Recently, a lot of attention has been focused on understanding the generalization properties of neural networks, notably using the generalization properties of invariant classifiers (Sokolic et al., 2017), margin preservation properties of neural networks (Sokolic et al., 2016) or the regularization properties of learning algorithms (Neyshabur et al., 2015; Keskar et al., 2017; Neyshabur, 2017).

2.1.3 Extracting Information: Fixed Representations and Empirical Dictionaries

A representation is a visual way to summarize a series, in order to investigate its properties. Finding representations that highlight the main variance sources for a set of signals is very important, both for property-based comparisons and for end-to-end models. Indeed, discriminant representations help to select the relevant properties to extract to compare signals. For end-to-end models, the decision process can be studied with such representation, in order to make it more interpretable. We describe in the following different representation methods for time series.

Global Representations

The most common representation of a temporal signal is a plot where the values are displayed against time. This kind of plots are useful because they are very general and, as we are very used to it, we can easily detect specific properties of the signal. Indeed, we recognize many properties of the signals from their plots when we see them, such as linearity, periodicity, stationarity, recurrent patterns, artifacts or ruptures. Also, experts are able to extract a lot of information from this representation. For instance, cardiologists are able to diagnose some disease by observing at electrocardiogram (ECG) signals. But this canonical representation becomes less informative when no clear shape is present in the signal, for instance in presence of noise.

Another common representation for signals is the Fourier spectrum of the signal. This representation reveals the harmonic properties of the series and attenuates the noise. Figure 2.4 shows an example of 3 signals represented using temporal and Fourier representations. With the temporal representation, the two noisy signals $X^{(2)}$ and $X^{(3)}$ appear to be the most similar. But using the Fourier representation, it can be seen that $X^{(3)}$ has an harmonic component with the same frequency as $X^{(1)}$. With the Fourier representation, we can thus see that $X^{(3)}$ is the sum of the same harmonic component as in $X^{(1)}$ and a noise term similar to $X^{(2)}$. This example shows that it is important to carefully select the representation used to study a set of series as the compared properties are linked to this representation. Looking at these representations of the signal, we access global properties that can then be quantified to globally distinguish the signals and their similarity.

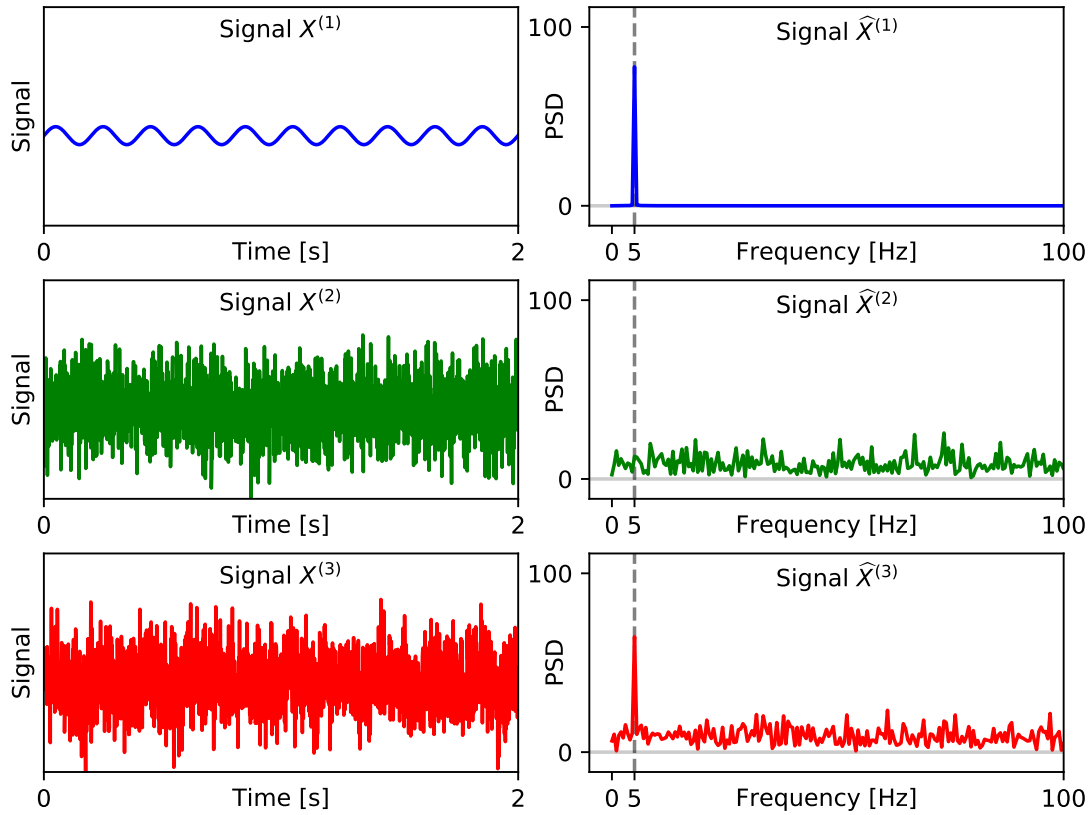


Figure 2.4: Comparison of 3 signals for temporal representation (*left*) and Fourier representation (*right*). With the temporal representation, signals $X^{(2)}$ and $X^{(3)}$ appear to be more similar but in the Fourier domain, $X^{(3)}$ is also close to $X^{(1)}$.

Extracting Local Structure

For non-stationary and noisy series, global properties of the signal are not very informative and might even be hard to estimate correctly. For instance, the estimation of the Fourier spectrum of non-stationary signal is unstable and it is hard to extract the relevant harmonics. The relevant information is thus contained in the local structures of the signal. To capture these structures, methods that analyze the signal locally are needed. A natural extension to the Fourier representation to local structure analysis was proposed in Gabor (1946), and later developed as the Short-Time Fourier Transform (STFT). This analysis uses Fourier Transform on windowed sub-series of the original signal. The information is not aggregated, but presented as a function of the time and the frequency and it reveals the transient structure in the signal. Figure 2.5 shows that this representation highlights the variation of the frequency structure of the series, which was not visible at all on its spectrum. The idea of using global analysis on portions of the signal has been a popular way to represent a signal. For instance, piecewise linear approximations (PLA) quantify the linearity of sub-segments of the original signal to reduce its complexity (see Keogh et al. 2001 and references therein). Another example of representation which study the local structure in the signal is the wavelet transform. The most common wavelet analysis computes a sparse representation of the signal, which concentrate the information around the discontinuities of the signal (see Mallat 2008 and reference therein). As this transform is multi-scale, it reveals phenomenons which have different time spans. Note that this transform have been

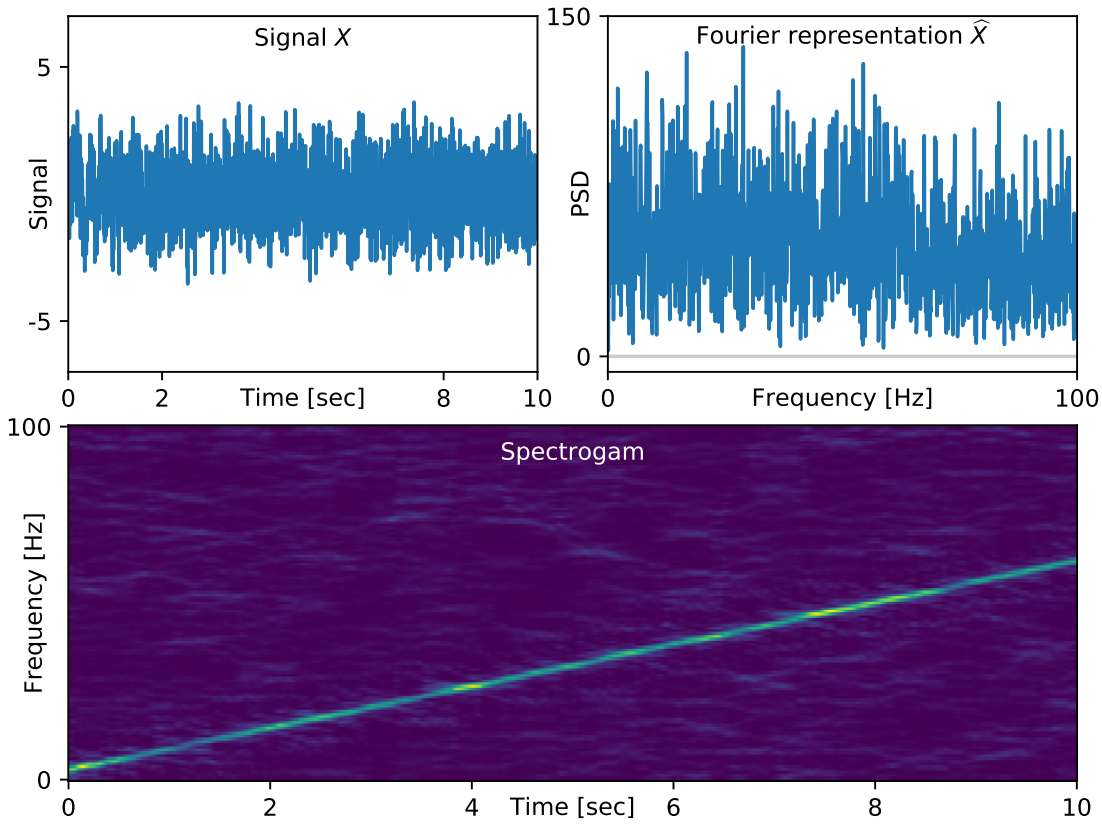


Figure 2.5: Different representations for signal $X[t] = \sin\left(\frac{(t+2)t}{2}\right) + \epsilon[t]$ with ϵ a white noise. (*top left*) Temporal representation, (*top right*) Fourier representation and (*bottom*) Spectrogram.

Figure 2.6: Illustration of the pattern based representation for a signal from human walking. (*top*) Temporal representation of an accelerometer signal on the vertical axis during the walk. The dashed lines indicate the localization of the steps, and highlight the regularity of the walk. (*bottom*) Three different steps from the signal, respectively the second step, the first step and the smaller center step and their activation signals computed with convolutional sparse coding. The activation signals indicate the position where the patterns are detected. This representation efficiently separates the local variations in the signal from their localization.

extended as a multi-layer analysis with the scattering transform (Mallat, 2012). But all these representations analyze specific properties, known *a priori*. Indeed, the use of Fourier analysis is designed to study harmonic properties, and the PLA quantifies the linearity of the segments in the series. If we do not know the structure of the signal, it is hard to chose a discriminant representation.

Pattern-based Representations

For series with unknown structure, the adaptivity of the representation method is critical. An idea to summarize the signal is to automatically extract the recurring shapes. The characteristic of the structures are learned from the data, making it possible to ex-

tract non-analytic behaviors. The local structures extracted are called *patterns*. Pattern representation was first developed for vector data as a way to reduce the variability of the points and the noise. Hotelling (1933) developed the popular Principal Component Analysis (PCA) to compute the vectors which explain the most the variance in the data. The principal components can be seen as pattern vectors, typical of the observed data. Another vector representation based on patterns is the K-means algorithm (Macqueen, 1967). This method assigns each vector to one of the K clusters and represents it as the centroïd of all the elements in the cluster. Many other classical vector techniques can be interpreted in the pattern-based representation framework such as the Independent Component Analysis (ICA, Naik & Kumar 2011), or the Non-negative Matrix Factorization (NMF, Gillis 2011). Olshausen & Field (1997) introduced the sparse dictionary learning, a method which learns patterns from the data, called atoms of a dictionary, and uses them to encode the original samples. This method can be seen as a very general framework to learn patterns.

For temporal signals, patterns are typically sub-series which are repeated in time. The series are encoded with the activation of a limited number of these patterns. This representation has a double advantage. First, the limited number of patterns ensure that we reduce the complexity of the variation of the series. Then, if the coding signal is sparse, this representation efficiently separates the variations of the signal from their localization in time. This kind of representation is quite natural for physiological signals which have some characteristic patterns such as ECG, EEG, or the vertical acceleration of a foot during the walk presented in Figure 2.6. By using insights from the dictionary learning for vector data, algorithms designed to extract typical local structures in signals have been developed recently. The advantages of this set of techniques are their adaptivity and their interpretability. Fixing the design of the learned dictionary allows to tweak for the size and scale of the atoms, and the resolution of the method to study the structure of the signal. Also, the split between the localization and the shape of the patterns makes the representation informative. For instance, in Figure 2.6, it is easier to study the regularity of the steps from the activation signal in dashed red than from the original signal in blue, as the variations are summarized by a unique pattern and the small variations from this pattern are discarded, making the analysis clearer and cleaner.

2.2 Thesis Contributions

2.2.1 Summary

During my Ph.D, I became interested in the issues of representation learning for time-series and interpretability of the learned representations. The convolutional dictionary learning for temporal signals are methods which allow to represent a signal in an intuitive and interpretable way. However, these methods can be complicated to use, due to the large number of parameters that influence them and to their computational cost. On the other hand, neural networks are very effective and solve many tasks in practice but it is very difficult to interpret the obtained results. The joint study of these two model classes and the ties between them can bring new perspectives to reduce the drawbacks of each of these methods.

In Part I, we study models based on convolutional representations and show how to improve their interpretability and computational cost.

Chapter 3: Convolutional Representations. The convolutional representation is used to represent time series by extracting patterns which are used to summarize the series variations. This model is interesting in the context of physiological signals which are quasi-periodic, with defined patterns. In Chapter 3, we present this model and its sparse version. Then we describe the state-of-the-art algorithms to compute the embedding with this models (see Section 3.3) and to update the patterns (see Section 3.4).

Chapter 4: Interpretability of the Singular Spectrum Analysis. The Singular Spectrum Analysis (SSA) is a technique used for short and noisy signal analysis. This technique extracts sub series from the original series and studies them using PCA. The principal components can be used to compute a decomposition of the signal with low-rank components, tied to the trend and seasonality of the studied signal. To improve the interpretability of the extracted components, the SSA requires a manual step which groups the raw components of the resulting decomposition. In Chapter 4, we make the following contributions.

- We show with Proposition 4.7 that this method solves a convolutional representation optimization problem, with dense activation, and we highlight the properties of the learned patterns. This shows that the SSA can be used to compute efficiently the solution of the non-convex problem of convolutional dense dictionary learning, for a certain class of orthogonal dictionaries.
- We describe a general unified framework to automate the grouping step in Section 4.5. In addition, we propose two novel similarity measures to compare the components (**GG3** and **HGS**) and a new group formation scheme based on the importance of each component, named hierarchical method (**HM**). These novel grouping strategies are compared to the methods proposed in the literature on generated signals.

Chapter 5: Distributed Convolutional Sparse Coding. The greedy coordinate descent can be used to solve the convolutional sparse coding. At each iteration, this algorithm updates the coordinate which is the farthest from its optimal value given all the other coordinates are fixed. It converges to the optimal solution and for large signal, numerical results show that it requires less iterations than its randomized counter part. We present the following contributions in Chapter 5.

- We introduce in Section 5.3, DICOD, a novel distributed algorithm, based on the greedy coordinate descent to solve the convolutional sparse coding. This algorithm is communication efficient and can run asynchronously.
- We also describe a sequential algorithm, called SeqDICOD. This algorithm is designed to run sequentially the updates made by DICOD. In this setting the updates are locally greedy. This reduces the computational cost of the updates compared to the greedy coordinate descent.

- In Section 5.4, we establish the convergence of DICOD with Theorem 5.2, under mild condition on the dictionary elements. We also show in Theorem 5.3 that the computational acceleration of DICOD is super-linear compared to the greedy coordinate descent. Theorem 5.5 shows that this acceleration is only sub-linear compared to our new locally greedy algorithm SeqDICOD.
- Finally, we demonstrate in Section 5.5 that these two algorithms work well in practice. We also confirm in Figure 5.6 that the computational acceleration of DICOD compared to greedy CD is quadratic when the number of cores is small enough.

Then, in Part II, we focus deep learning models and their internal representations aiming to improve their interpretability.

Chapter 6: Interpretability in Deep Learning Models. The Deep Learning models have improved the state-of-the-art performance for many tasks where signals are involved, such as images or audio signal processing. But these techniques are often seen as black boxes and provide little intuitions on their decision process. A key aspect is the lack of interpretability of their internal representation. Chapter 6 starts by recalling the general framework of deep learning and some of its theoretical properties. Then, we review recent results on neural network interpretability.

Chapter 7: Post-training for Deep Learning Models. During the training of a neural network, all the weights are updated together using an estimate of the gradient. For the end-to-end model, this adapts the representation learned by the first layers to the model solving the task at hand, which is computed in the last layers. At the end of the training, the model is considered to have learned both a good representation and a good model solving the task. The contributions made in Chapter 7 are the following.

- We propose in Section 7.2 an extra training step, called *post-training*, where the representation learned during training is fixed and we optimize the last layer. This extra step aims to improve the usage of the learned representation to solve the considered task.
- We propose a justification of our method based on the interpretation of neural network as a kernel method in Section 7.3.
- We show in Section 7.4 that this extra step provides a small performance boost for many network architectures, from convolutional networks to recurrent networks, and with different data sets.

Chapter 8: Understanding Trainable Sparse Coding. Some recent works have shown that it was possible to accelerate the resolution of the LASSO problem using a trained neural network to estimate the optimal solution. This study relies on the interpretation of the ISTA algorithm as a recurrent neural network, which can be unfolded K times to represent K iterations of the algorithm. The findings were backed by some interesting empirical results which showed that using the same number of ISTA iterations as the number of layers in the trained network was less efficient. In Chapter 8, we make the following contributions.

- We design in Section 8.2 an algorithm based on a factorization of the Gram matrix of the LASSO problem. This algorithm updates have the same computational cost as the iteration of ISTA. We show that the performances of this algorithm are linked to the sparsity of this factorization (see Proposition 8.1).
- We show with Theorem 8.2 that this algorithm has the same convergence rate as ISTA but with possibly better constant factors.
- In Section 8.3, we highlight with Theorem 8.7 the conditions under which the performance of our factorization based algorithm are better than ISTA, in expectation for generic dictionaries.
- Section 8.4 shows that our algorithm can be computed with a neural network called FacNet. This network is a re-parametrization of LISTA, with a more constrained parameter space. This shows that when FacNet is able to accelerate the resolution of LASSO, LISTA can also accelerate it. Thus, our results are sufficient to explain the acceleration of LISTA.
- Finally, we design in Section 8.5 an adversarial dictionary for which FacNet does not accelerate the resolution of the LASSO compared to ISTA. The results show that the performances of LISTA networks for this problem are also reduced. This empirical result suggests that our analysis captures part of the mechanism at work in the LISTA acceleration.

Part III presents some chosen results of physiological signal analysis. During my PhD, I have collaborated with medical doctors for clinical research purposes, developing tools to help them analyze physiological signals. This collaboration has been centered around two projects: the study of human walking and the study of nystagmus eye movements of young infants.

Chapter 9: Extracting Steps from Human Gait Signals. The quantification of human locomotion based on inertial sensors could change the way doctors follow their patients. By definition, walking is a repetitive movement, were the building block is the step. The extraction of the local structure in the signal enables the study of the regularity or the asymmetry of the signal. Thus, being able to robustly identify the steps in a walk exercise is critical to analyze the gait of the patient. In Chapter 9, we present the following contributions.

- In Section 9.3, we apply the convolutional representations described in Chapter 3 to signals of human walking. Preliminary results show that convolutional dictionary learning is able to identify local structures in the signals.
- We present a novel technique to robustly detect the steps in signals of human walking (Oudre et al., 2015). This technique relies on a base of steps templates to identify the start of a step. The algorithm identifies the steps robustly for healthy and pathological subjects.
- This study was associated to the analysis of signals from human walking with medical doctors in various studies, like Barrois et al. (2015) and Barrois et al. (2016). We briefly present our study in Barrois et al. (2015), which is included in the annex.

Chapter 10: Recording Eye Movements in Young Children. Neuro-ophthalmology is a field which studies the relation between the nervous system and the ocular system. The study of eye movements is particularly interesting as it sheds light on the control mechanisms between these two systems. In this thesis, we studied a particular type of eye movements, called the nystagmus, in early infancy. These movements are associated to various conditions which can be detected when the nystagmus is correctly classified. In Chapter 10, the following contributions are described.

- We showed that the Singular Spectrum Analysis (SSA) can be used to pre-process oculometric signals and to extract the eye movements linked to the nystagmus syndrome.
- We developed signal processing tools to analyze characteristics of the nystagmus syndrome to improve the doctor diagnosis.
- These tools were used for three studies: a communication at the Gordon conference on eye movement (Robert et al., 2015), a study on the nystagmus associated to optical path-way gliomas (Robert et al., 2016) and a study on the nystagmus for children with Down Syndrome.

2.2.2 Opensource development

During the second and third year of my PhD, I was involved in an open-source development projects, supported by the Center for Data Science, funded by the IDEX Paris-Saclay, ANR-11-IDEX-0003-02. The aim of the project was to provide a backend for the library `joblib`. `joblib` is a popular python library to easily parallelize scientific computations, as it provides simple support to embarrassingly parallel computation, where each process can perform independent computations and the results are also returned independently.

With Olivier Grisel, we developed `loky` to provide a robust, cross-platform and cross-version implementation of the `concurrent.futures.ProcessPoolExecutor` class. It notably features:

- **Deadlock free implementation:** one of the major concern in standard `multiprocessing` and `concurrent.futures` libraries is the ability of the `Pool/Executor` to handle crashes of worker processes. This library intends to fix possible deadlocks and send back meaningful errors in these situations.
- **Consistent spawn behavior:** All processes are started using fork/exec on POSIX systems. This ensures safer interactions with third party libraries.
- **Reusable executor:** strategy to avoid respawning a complete executor every time. A singleton executor instance can be reused (and dynamically resized if necessary) across consecutive calls to limit spawning and shutdown overhead. The worker processes can be shutdown automatically after a configurable idling timeout to free system resources.
- **Transparent `cloudpickle` integration:** to call interactively defined functions and lambda expressions in parallel. It is also possible to register a custom pickler implementation to handle inter-process communications.

- **No need for `if __name__ == "__main__":` in scripts:** thanks to the use of `cloudpickle` to call functions defined in the `__main__` module, it is not required to protect the code calling parallel functions under Windows.

2.3 Publications

The different work presented in this document resulted in various publications and communications:

- Moreau, T., Oudre, L., and Vayatis, N. Groupement automatique pour l ' analyse du spectre singulier. In *Proceedings of the Groupe de Recherche et d'Etudes en Traitement du Signal et des Images (GRETSI)*, 2015b
- Oudre, L., Moreau, T., Truong, C., Barrois-Müller, R., Dadashi, R., and Grégory, T. Détection de pas à partir de données d'accélérométrie. In *Proceedings of the Groupe de Recherche et d'Etudes en Traitement du Signal et des Images (GRETSI)*, Lyon, France, 2015
- Moreau, T., Oudre, L., and Vayatis, N. Distributed Convolutional Sparse Coding via Message Passing Interface (MPI). In *Proceedings of the NIPS Workshop on Nonparametric Methods for Large Scale Representation Learning*, 2015a
- Moreau, T. and Audiffren, J. Post Training in Deep Learning with Last Kernel. *arXiv preprint*, arXiv:1611(04499), 2016
- Moreau, T. and Bruna, J. Understanding Neural Sparse Coding with Matrix Factorization. In *Proceedings of the International Conference on Learning Representation (ICLR)*, 2017
- Moreau, T., Oudre, L., and Vayatis, N. Distributed Convolutional Sparse Coding. *arXiv preprint*, arXiv:1705(10087), 2017
- Barrois, R., Oudre, L., Moreau, T., Truong, C., Vayatis, N., Buffat, S., Yelnik, A., de Waele, C., Gregory, T., Laporte, S., and Others. Quantify osteoarthritis gait at the doctor's office: a simple pelvis accelerometer based method independent from footwear and aging. *Computer methods in biomechanics and biomedical engineering*, 18(Sup1):1880–1881, 2015
- Barrois, R., Gregory, T., Oudre, L., Moreau, T., Truong, C., Pulini, A. A., Vienne, A., Labourdette, C., Vayatis, N., Buffat, S., Yelnik, A., De Waele, C., Laporte, S., Vidal, P. P., and Ricard, D. An automated recording method in clinical consultation to rate the limp in lower limb osteoarthritis. *PLoS ONE*, 11(10):e0164975, 2016
- Robert, M., Contal, E., Moreau, T., Vayatis, N., and Vidal, P.-P. The Why and How of Recording Eye Movement from Very Early Childhood. Oral Presentation, Gordon Research Conference on Eye Movement, 2015

Part I

Pattern-based Time Series Representation

In this part, we focus on pattern-based representations. Part I is organized as follows. Chapter 3 introduces the convolutional representation model as well as state-of-the-art algorithms used to extract patterns from signals. Then, Chapter 4 introduces the Singular Spectrum Analysis (SSA) and shows that it actually corresponds to a convolutional representation with specific patterns. Then, a framework to automatically improve the interpretability of the components computed with the SSA is evaluated. Finally, Chapter 5 presents a novel algorithm based on greedy coordinate descent to solve the convolutional sparse coding. This algorithm can be distributed asynchronously to represent long signals in the convolutional representation model. It is proven to converge and to have a super-linear speedup compared to the classical greedy coordinate descent algorithm.

3

Convolutional Representations: a state-of-the-art

"Set patterns, incapable of adaptability, of pliability, only offer a better cage. Truth is outside of all patterns."

– Bruce Lee

Contents

In this chapter, we describe the convolutional representation framework and present state-of-the-art algorithms used to compute such representations. The convolutional representation has been used to compute unsupervised representations of signals in various fields and is well adapted to study physiological signals. There are multiple methods to compute such representation, due to the independence of the different computation blocks involved in embedding the data or updating the patterns. We present here a unified view of these different blocks.

3.1 Convolutional Representation

The convolutional representation is an efficient and adaptive model to describe the patterns composing a signal. Consider the P-dimensional signal $X \in \mathcal{X}_T^P$ of length T. Let $\boldsymbol{D} = \{\boldsymbol{D}_k\}_{k=1}^K \subset \mathcal{X}_W^P$ be a set of K multivariate patterns of length $W \ll T$ with the same dimension P and $\{Z_k\}_{k=1}^K \subset \mathcal{X}_L^1$ be a set of K scalar activation signals with length $L{=}T{-}W{+}1$. We denote $Z \in \mathcal{X}_L^K$ the K-dimensional signal of length L such that

$$Z[t] = \begin{pmatrix} Z_1[t] \\ \dots \\ Z_K[t] \end{pmatrix} .$$

Definition 3.1. *The convolutional representation models a multivariate signal X as the sum of K convolutions between a multivariate pattern $D_k \in \mathcal{X}_W^P$ and an activation signal $Z_k \in \mathcal{X}_L$ such that:*

$$X[t] = \sum_{k=1}^{K} (Z_k * \boldsymbol{D}_k)[t] + \mathcal{E}[t], \qquad \forall t \in [\![0, T-1]\!] \; . \tag{3.1}$$

with $\mathcal{E} \in \mathcal{X}_T^P$ representing an additive noise signal with the same length and dimension as X.

A univariate signal generated using this model is presented in Figure 3.1. This kind of model has first been introduced by Grosse et al. (2007), who showed this method can be useful for audio signal comparison.

A shift Invariant Model. This model is well suited for time series studies as it captures the patterns in a shift-invariant way. Indeed, with the convolution operator, the presence of the k-th pattern $\boldsymbol{D}_k$ at any time in the signal is encoded in the activation signal Z_k. Using the code signals, it is easy to study the regularity of a pattern occurrence or the correlation between patterns. The representation based on model (3.1) is particularly useful to separate the localization of the patterns, encoded in the activation signal Z, and their shapes captured in $\boldsymbol{D}$. The search for patterns shorter than the signal permits to compare easily the representation of signals with different lengths as it focuses on local structures in the signal.

Higher Order Extension. Note that this model can be extended to higher order signals such as images by using the proper convolution operator. In this thesis, the focus is set on 1D-convolution due to the application domain, with temporal physiological signals. Most of the algorithms described here can be easily adapted using 2D-convolutions and the focus of many works on convolutional dictionary learning was the image processing (Bristow et al., 2013; Chalasani et al., 2013; Kavukcuoglu et al., 2010).

3.1.1 Interpretability of the Dictionary

The choice of dictionary to encode the signal is critical as it conditions the interpretability of the coding signal. The dictionary can be chosen *a priori*, with analytical patterns which capture specific properties, or adapted to the data, to highlight specific structures present in the data.

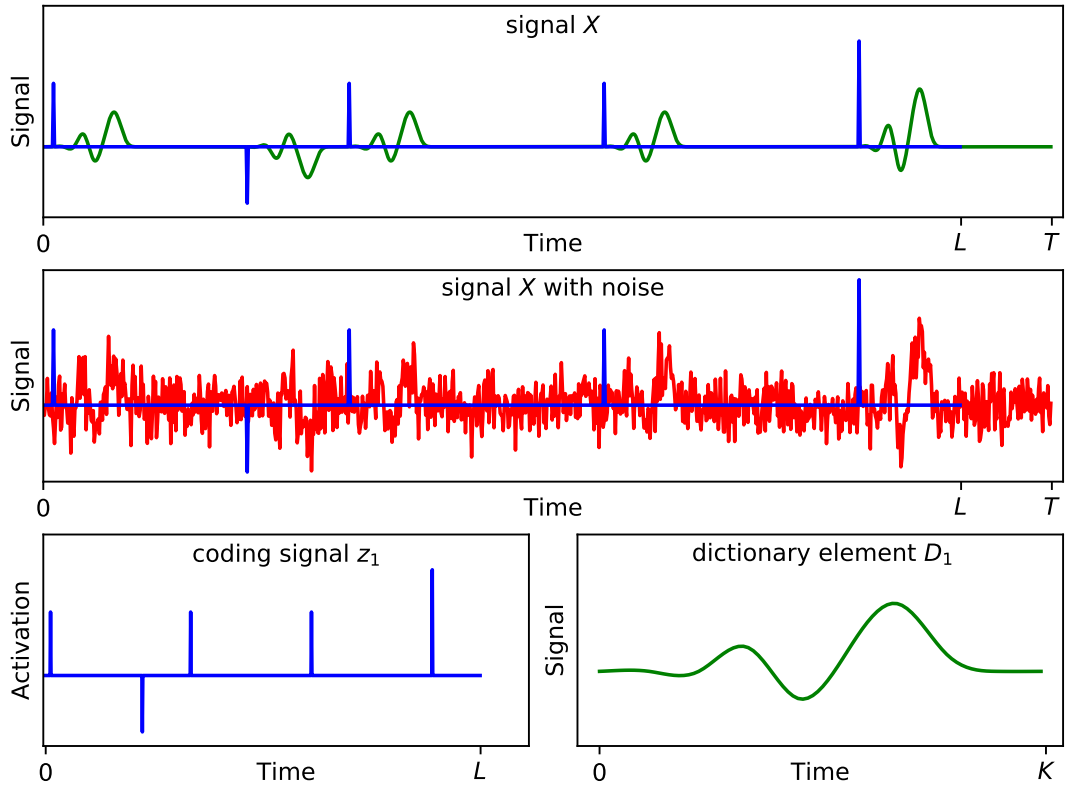


Figure 3.1: Representation of a scalar signal X generated from one dictionary element D_1 and the coding signal Z_1 using the model (3.1). The pattern D_1 is repeated at different time locations with different activation intensities in the signal. The coding signal Z_1 captures the apparitions of the pattern D_1 in all possible shifted time locations.

Analytic Dictionary

The analytic approach uses predefined functions to extract interpretable properties of the signal. For example, the dictionary can be set to be the Fourier basis. When this technique is used on the full signal, it gives information on oscillatory properties of the signal such as its harmonic spectrum. This dictionary can also be used to encode subsegments of the signal using the short-time Fourier Transform (STFT) (Gabor, 1946). This technique can be analyzed in the convolutional dictionary model as the resulting spectrogram can be interpreted as an activation signal for each harmonic. Other transformation can be used on windowed signals such as Discrete Cosine Transform (Ahmed et al., 1974), Hadamard Transform (Pratt et al., 1969) or Wavelet Transform (Mallat, 2008). Each of these transformations is linked to an analytic dictionary. The wavelet transform is of particular interest when interpreted as a convolutional representation. The idea is to use a base pattern, called mother wavelet, which is replicated with different scales (typically the powers of 2) and positions to extract information localized around signal discontinuities. Here, the dictionary is composed of a single pattern, scaled to multiple levels. Using wavelets brings two important advantages: the embedding of the signal on the dictionary can be done using a very efficient methods and the resulting activation signal is very sparse. With analytical dictionaries, the representation captures particular properties that are defined *a priori*. Thus, the choice of the dictionary have a large impact on the type of information highlighted.

Learned Dictionary

Another choice for the dictionary is to learn it from the observed signals. The shape of the dictionary elements are chosen with an optimization problem, leading to representations that capture the intrinsic information by analyzing unknown patterns. The dictionary elements, also called atoms, are chosen from a constraint set Ω which is defined by the user. The most common constraint imposed on the dictionary atoms is to have a unit norm. Indeed, for convolutional sparse coding – where we seek to minimize the ℓ_1-norm of Z and the reconstruction error – scaling $\boldsymbol{D}$ by α and Z by $1/\alpha$, for $\alpha > 1$, does not change the reconstruction error but decreases ℓ_1-norm of Z by a factor $1/\alpha$. Thus, without the unit norm constraint, Z tends to 0 and the norm of the atoms explodes. Other insights for dictionary design can be taken from matrix factorization and dictionary learning techniques for vectorial data. For instance, Principal Component Analysis uses the directions that best explain the variance in the data set as dictionary elements (Hotelling, 1933). A similar problem can be solved for signals with the Singular Spectrum Analysis (SSA) introduced by Vautard & Ghil (1989). It analyzes the recurrent patterns in a time series and extracts the ones that explain the variance of the signal in order to encode the signal using them as dictionary elements. This technique has been studied in details by Golyandina et al. (2001) and is used to analyze short and noisy time series. It notably decomposes the series as the sum of a trend pattern, a few seasonal components and a noise term. In Chapter 4, we will show how this method can be used to compute a convolutional representation. The same idea could be used to extend other matrix factorization techniques such as Independent Component Analysis (Naik & Kumar, 2011), which computes statistically independent dictionary elements. Another example of dictionary learning technique is the Empirical Wavelet Transform (EWT) developed by Gilles (2013). This method chooses the dictionary elements by adapting to the observed signals an analytic base constructed from wavelet multi-resolution, thus segmenting the spectral information of the signals. Finally, the dictionary can be learned directly by solving an optimization problem for a given constraint set Ω. This process is described in Section 3.2.

3.1.2 Interpretability of the Activation Signal

The choice of constraints for the signal embedding on the dictionary controls the properties of the representation. A possible constraint is to penalize with the rank of the representation as done in Liu et al. (2010) or Candes et al. (2011). This ensures a representation in a low dimensional manifold, compressing the information on a few patterns. The low-rank embedding ensures that if two signals can be represented with similar patterns, the distance between their representation should be small.

Another common hypothesis for the convolutional representation is to assume that the coding signals Z_k are sparse, in the sense that only few entries are nonzero in each signal. The sparsity property forces the representation to display localized patterns in the signal. This is very important to the interpretability of the coding signal. Indeed, a sparse signal can be seen as events, or spikes, occurring in time and it is easier to distinguish their regularity of their correlation than with a dense signal. This constraint can be enforced using the ℓ_0 regularization, with algorithms such as the matching pursuit (Mallat & Zhang, 1993) and the orthogonal matching pursuit (Pati et al., 1993). Other techniques ensure sparsity by solving a relaxed version of the ℓ_0-problem using norm L_p, $p \leq 1$, with algorithms like FOCUSS (Gorodnitsky & Rao, 1997). One noticeable sparsity inducing technique is the convolutional sparse coding with the sparsity

constraint enforced using a ℓ_1-regularization of the coding signals. The embedding is computed using an specialization of the LASSO problem with convolution product instead of matrix multiplication. Section 3.3 provides details on this method and the state-of-the-art algorithm to compute the signal embedding.

Finally, nonnegative constraints, similar to the one used in Nonnegative Matrix Factorization (NMF) (Gillis, 2011), can be used to improve interpretability. Indeed, forcing the nonnegativity of the activation signals avoids having multiple patterns being mixed together to represent the same phenomena by canceling some parts of each other. This improves interpretability as it makes it easier to see the effect of each separate components.

3.1.3 Link to Classic Dictionary Learning

The convolutional representation is an extension of the matrix factorization problem, which can be retrieved for $W = T = 1$. In this case, the multivariate signal Z is just a coding vector as it is of length $L = 1$ and the sum of convolution can be summarized as a matrix multiplication such that

$$X[0] = Z[0]\boldsymbol{D}[0] + \mathcal{E}[0] \ .$$

Here, $X[0]$ and $Z[0]$ are vectors, respectively of size P and K and $D[0]$ is a matrix of size $K \times P$. Also, using vector forms for the signals, it is possible to re-write the convolutional model as a vectorial model. For a signal $X \in \mathcal{X}_T^P$, we define the vector $\bar{x} \in \mathbb{R}^{PT}$ representing X as

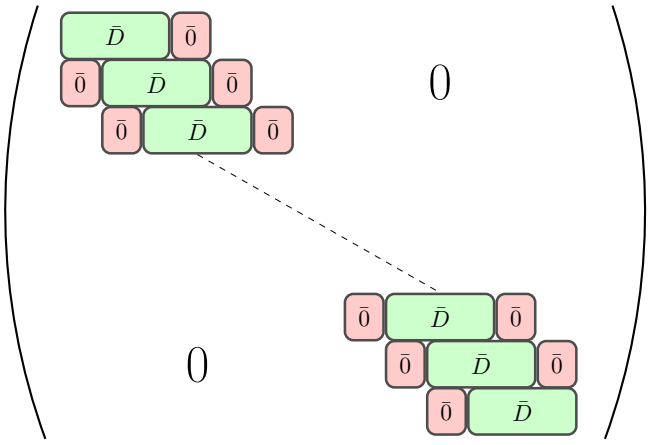
$$\bar{x}_{t*P+k} = X_k[t] \qquad \forall (k,t) \in [\![1, K]\!] \times [\![0, T-1]\!] \ .$$

We define a band circulant matrix $D \in \mathbb{R}^{KL \times PT}$ constructed by repeating the blocks $\bar{D} \in \mathbb{R}^{K \times PW}$ at every time positions, *i.e.* for the block dictionary

$$\bar{D} = \begin{bmatrix} d_{1,1}[0] & \dots & d_{1,P}[0] & \dots & d_{1,1}[W] & \dots & d_{1,P}[W] \\ d_{2,1}[0] & \dots & d_{2,P}[0] & \dots & d_{2,1}[W] & \dots & d_{2,P}[W] \\ \vdots & & \vdots & & \vdots & & \vdots \\ d_{K,1}[0] & \dots & d_{K,P}[0] & \dots & d_{K,1}[W] & \dots & d_{K,P}[W] \end{bmatrix} ,$$

the band circulant matrix D associated to the convolutional dictionary $\boldsymbol{D}$ repeats $\bar{D}$ $L = T - K + 1$ times a shift of P columns.

$D =$



where the $\bar{0}$ here denotes matrices in $\mathbb{R}^{K\times P}$ filled with 0. This dictionary contains all the patterns from the convolutional dictionary $\boldsymbol{D}$ replicated in all shifted positions. Using this band circulant matrix D and denoting $\bar{z} \in \mathbb{R}^{KT}$, the model (3.1) can be re-written as

$$\bar{x} = \bar{z}D + \bar{\varepsilon} \tag{3.2}$$

where $\bar{x}, \bar{\varepsilon} \in \mathbb{R}^{PT}$ and $\bar{z} \in \mathbb{R}^{KL}$ are vectors representing the signals $X, \mathcal{E}$ and Z. Thus, convolutional dictionary learning is equivalent to dictionary learning for band-circulant matrices and the classical algorithms can be used. However, due to the dimension of (3.2), the algorithms are not efficient enough as they do not make use of the specific structure of the problem. Moreover, if the signals to encode do not have the same length, this formulation cannot be used for multiple signals without zero-padding. The use of the more compact representation (3.1) is thus preferred for convolutional representation.

3.2 Learning Dictionary via Alternate Minimization

Multiple algorithms have been designed to learn a dictionary suitable to represent a set of vectors, such as the Method of Optimal Directions (MOD; Engan et al. 1999), the K-SVD (Aharon et al., 2006), the stochastic gradient descent (Aharon & Elad, 2008) or the online dictionary learning (Mairal et al., 2010). These techniques all rely on an alternated procedure which computes the embedding of the data point on the dictionary and then update the dictionary to improve the representation. In the case of the ℓ_1-penalized coding signal, the process to learn a dictionary that can describe a given set of signals $\left\{X^{[1]}, \dots, X^{[N]}\right\}$ can be posed as an optimization problem such that

$$\underset{\boldsymbol{D}\in\Omega}{\operatorname{argmin}} \underbrace{\frac{1}{N}\sum_{n=1}^{N} \underset{Z^{[n]}}{\operatorname{argmin}} \frac{1}{2}\left\| X^{[n]} - \sum_{k=1}^{K} Z_k^{[n]} * \boldsymbol{D}_k \right\|_2^2}_{G_N(\boldsymbol{D})} + \lambda\Psi\left(Z^{[n]}\right) , \tag{3.3}$$

for a certain set of constraints Ω for the dictionary $\boldsymbol{D}$ and a regularization function Ψ for the coding signal. For sparse dictionary learning, the regularization function Ψ is usually chosen chosen to be either the ℓ_1 or ℓ_0-norm of the coding signal (Grosse et al., 2007; Yellin et al., 2017). The learned dictionary can be adapted to the intended usage by imposing different sets of constraints Ω to the dictionary's elements. Finding both $\boldsymbol{D}$ and Z is a non-convex problem. However, estimates of the solution can be obtained using alternate minimization. This method requires to be able to solve two independent steps. The first one is to infer the activation signal associated to a given signal and dictionary. This step is named convolutional sparse coding and is described in Section 3.3. It computes the embedding $Z^{[n]}$ for all signals $X^{[n]}$ and the current dictionary $\boldsymbol{D}^{(q)}$ for $n \in [\![1, N]\!]$. Then, the second step is to be able to update the dictionary, to find the atoms that best describes the given signal population. Given a constraint set Ω, the dictionary is updated to improve the modeling of the signals $X^{[n]}$ given the current coding signals $Z^{[n]}$. This step and algorithms to update the dictionary are described in Section 3.4.

3.2.1 Online Learning

One of the drawbacks of the alternate minimization algorithm is its complexity. At each iteration, it is necessary to compute convolutional sparse coding for all signals. This step is computationally expensive. Moreover, if the dictionary needs to be updated

to take into account a new signal, it is also necessary to recompute the codes for all the other signals to be able to perform the update. One way to improve the efficiency of the alternate minimization is to use online dictionary learning. This technique – proposed by Mairal et al. (2010) for vectorial data – has recently been extended for convolutional dictionary learning in Liu et al. (2017). The core idea of this procedure is to approximate at each iteration q the function G_N by a surrogate function $\widehat{G}_q$ defined as

$$\widehat{G}_q(\boldsymbol{D}) = \frac{1}{q}\sum_{p=1}^{q}\frac{1}{2}\left\|X^{[p]} - \sum_{k=1}^{K} Z_k^{[p]} * \boldsymbol{D}_k\right\|_2^2, \tag{3.4}$$

with $Z^{[p]}$ the coding signal for signal $X^{[p]}$ for dictionary $\boldsymbol{D}^{(p)}$ at iteration p. Note that with this surrogate function, only one signal $X^{[p]}$ is encoded at each iteration.

Multiple variants of this algorithm have been proposed to improve its convergence. A very natural variant is to use mini-batch updates for the surrogate function. At each step q, instead of only selecting one signal, a batch of Q signals are added in the surrogate function. In order to give more importance to the samples with more accurate coding signals, computed with more recent dictionary, Mairal et al. (2010) proposed to add a forgetting factor to the previous samples. The surrogate function becomes

$$\overline{G}_{q,\gamma}(\boldsymbol{D}) = \frac{1}{q}\sum_{p=1}^{q}\frac{\gamma^p}{2}\left\|X^{[p]} - \sum_{k=1}^{K} Z_k^{[p]} * \boldsymbol{D}_k\right\|_2^2, \tag{3.5}$$

where $0 < \gamma < 1$ is a forgetting factor, controlling the importance of history in the new updates. Another online approach to learn convolutional dictionary was described in Kavukcuoglu et al. (2010). In their paper, they rely on the stochastic gradient descent approach to update their dictionary. At each iteration q, they draw randomly one of the signal $X^{[q]}$ and compute its coding signal $Z^{[q]}$. The dictionary is then updated by considering the surrogate function g_q, defined as

$$g_q(\boldsymbol{D}) = \frac{1}{2}\left\|X^{[q]} - \sum_{k=1}^{K} Z_k^{[q]} * \boldsymbol{D}_k\right\|_2^2. \tag{3.6}$$

3.2.2 Theoretical Guarantees for Convolutional Representation

The alternate minimization approach for dictionary learning is not guaranteed to converge to a good solution in general. Indeed, the non-convexity of the problem makes it hard to determine a convergence at all. The first theoretical studies of alternate minimization for dictionary learning were for vectorial data. In their paper, Agarwal et al. (2014) show that if data is generated using a dictionary $\boldsymbol{D}^0$, there exists a polynomial time algorithm which permits estimating this dictionary, given that there are enough samples and that the observed signals do not have noise or outliers. The algorithm relies on initialization schemes proposed by previous works (Arora et al., 2013; Agarwal et al., 2013) which ensures that the initial estimate of the dictionary is in the neighborhood of the solution. Gribonval et al. (2015) show the sample complexity of dictionary retrieval methods under presence of noise and outlier points. This paper does not provide an algorithm to find the dictionary but quantifies the effect of the assumptions made in the model. One key quantity in their analysis is the *cumulative coherence* which quantifies the over-completeness of the dictionary elements with the sparsity of the coding vectors.

As we have seen in Subsection 3.1.3, the convolutional setting is equivalent to the vectorial case. Thus, these previous works can be directly applied for convolutional dictionary learning. However, due to the particular structure of the problem, these results can be improved to better handle the temporal structure. Recent work from Papyan et al. (2017) introduces quantities which extend the different concepts used in sparse coding literature to convolutional settings and highlights the properties of dictionary elements critical for the uniqueness of the coding signal. The core of their analysis is to study the properties of stripes of the original signal, avoiding the very large dimension of the whole signal. Their work is completed in a companion paper which studies the recovery capacities of classical convolutional sparse coding algorithms for noisy observations (Papyan et al., 2016).

3.3 Convolutional Sparse Coding

The convolutional sparse coding refers to the computation of the embedding of a signal X on a fixed dictionary $\boldsymbol{D}$ with a sparsity inducing regularization Ψ, solving

$$\underset{Z}{\operatorname{argmin}} \frac{1}{2} \left\| X - \sum_{k=1}^{K} Z_k * \boldsymbol{D}_k \right\|_2^2 + \lambda \Psi\left(Z\right) \ . \tag{3.7}$$

The choice of the regularization function Ψ has an impact on the sparsity of the estimated coding signal and the performance of the algorithms used to solve (3.7). The ℓ_0-norm is a natural choice for Ψ as it is directly measuring the sparsity of the solution, but the problem (3.7) in this case is non-convex and NP-hard to solve. Greedy algorithms such as the Matching Pursuit (MP; Mallat & Zhang 1993) and the Orthogonal Matching Pursuit (OMP; Pati et al. 1993) efficiently compute an approximate solution to this problem, and can give good results in practice (Yellin et al., 2017). A convex relaxation of this problem is obtained by taking Ψ to be the ℓ_1-norm of the coding signal and efficient algorithms can compute the minimal solution. Under some assumptions on the sparsity of the solution and the design of the dictionary, this relaxation can be shown to consistently estimate the solution of the ℓ_0 problem (Donoho & Elad, 2002; Fuchs, 2004). In the following, we focus on sparse coding with ℓ_1-regularization as its guaranteed convergence can improve the accuracy compared to approximate ℓ_0 minimization algorithms. Note that ℓ_1-regularized methods tend to be slower than the greedy ℓ_0 approaches so the choice of the method results of tradeoff between computational power and accuracy.

The convolutional sparse coding refers to the embedding of a signal on a dictionary with a ℓ_1 regularization. Given a dictionary of patterns $\bar{\boldsymbol{D}}$, convolutional sparse coding aims to retrieve the activation signals Z^* associated to the signal X by solving the following optimization problem,

$$Z^* = \underset{Z}{\operatorname{argmin}}\, E(Z) \triangleq \underbrace{\frac{1}{2} \left\| X - \sum_{k=1}^{K} Z_k * \boldsymbol{D}_k \right\|_2^2}_{h_1(Z)} + \underbrace{\lambda \left\| Z \right\|_1}_{h_2(Z)} \ , \tag{3.8}$$

for a given regularization parameter $\lambda > 0$. (3.8) can be interpreted as a special case of the LASSO problem with the band circulant matrix $\bar{\mathcal{D}}$ and the model (3.2). Therefore, classical optimization techniques designed for LASSO can be applied to solve it with the

Methods	Original paper for sparse coding	Convolutional adaptation
FSS	Grosse et al. 2007	Lee et al. 2007
FISTA	Beck & Teboulle 2009	Chalasani et al. 2013
ADMM	Gabay & Mercier 1976	Bristow et al. 2013
CD	Friedman et al. 2007	Kavukcuoglu et al. 2010

Table 3.1: Algorithms for ℓ_1-regularized convolutional sparse coding

same convergence guarantees. However, the dimension of the problem is too large for the algorithms to be efficient. The expression (3.8) using convolution allows the use of more efficient computation for the gradient of h_1 and it is possible to design. State-of-the-art algorithms for (3.8) and their link to original algorithm in the optimization literature are recalled in Table 3.1 and the following subsections describe them in details.

3.3.1 Feature Sign Search (FSS)

The Feature Sign Search algorithm (FSS) was introduced by Lee et al. (2007). This algorithm is developed to solve traditional sparse coding problem, without the convolution but can be easily adapted to solve the convolutional case. Knowing the signs of the non-zero coefficients of the optimal solution, we can replace the ℓ_1-norm by a linear operator. If Θ^* denotes the sign of the values of Z^* such that, $\Theta^*_k[t] = \text{sign}(Z^*_k[t])$, then

$$\|Z^*\|_1 = \langle \Theta^*, Z^* \rangle \ . \tag{3.9}$$

If Θ^* is known, (3.8) is equivalent to the differentiable problem

$$\underset{Z \in \mathcal{X}_L^K}{\text{argmin}} \frac{1}{2} \|X - \sum_{k=1}^{K} Z_k * \boldsymbol{D}_k\|_2^2 + \lambda \langle Z, \Theta^* \rangle \ . \tag{3.10}$$

Here, the non-differentiable ℓ_1-norm has been replaced by a linear scalar product. Grosse et al. (2007) showed that the FSS algorithm, designed to solve the LASSO problem, could be efficiently used for the convolutional sparse coding.

Algorithm 3.1 describe the pseudo code of the FSS algorithm in details. The idea of FSS is to estimate the signs of the coefficients in Z^* and to solve the resulting quadratic program (QP) sub-problem (3.10). Each iteration refines the sign estimation, and the algorithm converges to a global solution. Due to the high dimension of the problem, the algorithm is designed as a working set algorithm, putting coefficients which are estimated to be non-zero in the active set and then computing the solution of the resulting QP. From this new solution, the sign of Z^* is estimated again and the working set is updated accordingly.

Solving the QP Sub-problems

In order to clarify its steps, it is necessary to use a vector form for (3.10), which has a closed form solution and helps understand the benefit of the working set. We recall that for a signal $X \in \mathcal{X}_T^P$, $\bar{x}$ is defined as $\bar{x}_{t*P+p} = X_p[t]$. Re-using the vectorized form of the convolutional representation from (3.2), with the vector $\bar{x} \in \mathbb{R}^{PT}$ representing X and the vectors $\bar{z}, \bar{\theta}^{(q)} \in \mathbb{R}^{KL}$ representing Z and $\text{sign}(Z)$, (3.10) can be rewritten as

$$\underset{\bar{z}}{\text{argmin}} \underbrace{\frac{1}{2} \|\bar{x} - D\bar{z}\|_2^2}_{F(\bar{z})} + \lambda \langle \bar{\theta}^{(q)}, \bar{z} \rangle \ . \tag{3.11}$$

Algorithm 3.1 Feature Sign Search Algorithm

Input $X, \boldsymbol{D}$ and λ
We recall that $\bar{x}$ is the vector notation for $X \in \mathcal{X}_T^P$, *i.e.* $x_{t*P+p} = X_p[t]$
And D is the band cirucant matrix associated to $\boldsymbol{D}$ (see Subsection 3.1.3)
Initialize $\bar{z} = 0, \bar{\theta} = 0$ and $\mathcal{I} = \{\}$
repeat
  Compute the gradient vector $G = \nabla h_1(Z)$
  Update the active set $\mathcal{I} = \mathcal{I} \cup \left\{ t * K + k \;\middle/\; \left|G_k[t]\right| > \gamma \right\}$
  Estimate the signs on the active set *s.t.*

$$\Theta_k[t] = \begin{cases} \text{sign}(G_k[t]) & \text{if } Z_k[t] = 0 \\ \text{sign}(Z_k[t]) & \text{if } Z_k[t] \neq 0 \end{cases} \quad \forall (k,t) \;\; s.t \;\; (t * K + k) \in \mathcal{I}$$

  Solve the QP sub-problems:
  repeat
    Extract matrix $\widetilde{A} = D_{[\mathcal{I},\mathcal{I}]}$;
    Extract vectors $\widetilde{z} = \bar{z}_{\mathcal{I}}; \quad \widetilde{\theta} = \bar{\theta}_{\mathcal{I}}; \quad \widetilde{x} = \bar{x}_{\mathcal{I}}$;
    Compute solution of (3.11) with $\widetilde{z}_{new} = (\widetilde{A}^\mathsf{T}\widetilde{A})^{-1}(\widetilde{A}^\mathsf{T}\widetilde{x} - \lambda\widetilde{\theta}/2)$
    Discrete line search:
      Set $\mathcal{S}_d = \left\{ \widetilde{z}^\alpha; \alpha \in \left\{ \frac{\widetilde{z}_{new,i}}{\widetilde{z}_{new,i} - \widetilde{z}_i} \right\} \cap [0,1] \text{ and } \widetilde{z}_j^\alpha = \begin{cases} 0 \text{ if } \widetilde{\theta}_j \widetilde{z}_j^\alpha < 0 \\ \alpha \widetilde{z}_j + (1-\alpha)\widetilde{z}_{new,j} \end{cases} \right\}$
      Set $\widetilde{z}^* = \underset{\widetilde{z}^\alpha \in \mathcal{S}_d}{\text{argmin}} \|\widetilde{x} - \widetilde{A}\widetilde{z}^\alpha\|^2 + \lambda \widetilde{\theta}^\mathsf{T} \widetilde{z}^\alpha$
    Set $\bar{z}_{\mathcal{I}} = \widetilde{z}^*$ and $\bar{\theta} = \text{sign}(\bar{z})$
  until $\bar{g}_j + \lambda \bar{\theta}_j = 0 \quad \forall j \in \left\{ j \in \mathcal{I}; \bar{z}_j \neq 0 \right\}$
  Set $\mathcal{I} = \left\{ i | \bar{z}_i \neq 0 \right\}$;
until $|\bar{g}_j| \leq \lambda \quad \forall j \in \left\{ j \in \mathcal{I}; \bar{z}_j = 0 \right\}$
Return Z

This minimization problem has a closed form solution

$$\bar{z} = \left(D^\mathsf{T} D\right)^{-1} \left(D^\mathsf{T} \bar{x} - \lambda \bar{\theta}^{(q)}/2\right) \; . \tag{3.12}$$

The computational cost of (3.12) is too expensive to be computed for the whole problem, as its complexity is $\mathcal{O}\left(K^3 T^3\right)$. The FSS algorithm reduces the computational cost of solving the QP by only considering a working set of coefficient $\mathcal{I}^{(q)}$ at each iteration q. By only considering non-zero variable in $\mathcal{I}^{(q)}$, (3.12) can be reduced to

$$\bar{z}_{\mathcal{I}^{(q)}} = \left(D^\mathsf{T} D\right)^{-1}_{[\mathcal{I}^{(q)},\mathcal{I}^{(q)}]} \left((D^\mathsf{T} \bar{x})_{[\mathcal{I}^{(q)},\mathcal{I}^{(q)}]} - \lambda \bar{\theta}^{(q)}_{\mathcal{I}^{(q)}}/2\right) \; . \tag{3.13}$$

The complexity of the iteration is reduced from $\mathcal{O}\left(K^3 T^3\right)$ to $\mathcal{O}\left(|\mathcal{I}^{(q)}|^3\right)$. In practice, due to the sparsity of the searched solution, the size of the working set is manageable and the complexity of the iteration does not explode.

Sign Estimation and working set extension

At each iteration q, the estimated sign of the solution is updated. For coefficient i which are non-zero in the current solution $Z^{(q-1)}$, the estimation of the sign is set to be coherent with the solution estimate, *i.e.*

$$\bar{\theta}_i^{(q)} = \text{sign}\left(\bar{z}_i^{(q-1)}\right) \ .$$

Then, the working set is updated to include extra coefficient in $\mathcal{I}^{(q)}$. To avoid overly growing the working set, a fixed number of coefficient are chosen from the zero coefficients to be added to the working set. The selected coefficients are the one with maximal gradient, such that

$$\mathcal{I}^{(q)} = \mathcal{I}^{(q-1)} \cup \left\{ i \ \middle/ \ |\nabla F(\bar{z}^{(q-1)})_i| \geq \max\left(\lambda, \nabla F(\bar{z}^{(q-1)})_j|\right) \quad \forall j \notin \mathcal{I}^{(q-1)} \right\} \ . \tag{3.14}$$

For these coefficients, the sign is estimated from the value of the gradient and

$$\theta_i = \text{sign}\left(\nabla F(z^{(q-1)})_i\right) \quad \forall i \in \mathcal{I}^{(q)} \ s.t. \ \bar{z}_i^{(q-1)} = 0 \ . \tag{3.15}$$

Discrete Line Search

An important point of this algorithm is that the current solution should stay coherent with the sign estimate. This property ensures that the cost function will always decrease with the algorithm iterations. The solution computed with (3.12) is not guaranteed to be coherent. To cope with this, a line search is used to find a point, coherent with the current sign estimate $\theta^{(q)}$ which decreases the objective function. As the objective (3.11) is convex, this line search can be conducted by looking at a discrete number of point, where coefficients are zeroed. First, the QP sub problem solution $\bar{y}$ is computed with (3.12). We define the set of coefficients $\mathcal{J}$ which are not coherent, such that

$$\mathcal{J} = \left\{ i \ \middle| \ \bar{\theta}_i^{(q)} \bar{y}_i < 0 \right\} \ .$$

Then, we find the points on the segment from $\bar{z}^{(q-1)}$ to $\bar{y}$ where the coefficients are null, *i.e.* for $j \in \mathcal{J}$, we define $\alpha_j \in [0, 1]$ as the number such that $\left(\alpha_j \bar{z}^{(q-1)} + (1 - \alpha_j)\bar{y}\right)_j = 0$. These scalars have a closed form,

$$\alpha_j = \frac{\bar{z}_j^{(q-1)}}{\bar{z}_j^{(q-1)} - \bar{y}_j} \ .$$

The discrete line search is performed for all these value $\{\alpha_j\}_{j \in \mathcal{J}}$. We define a set of possible coherent solutions $\{\bar{y}^{[j]}\}_{j \in \mathcal{J}}$ such that

$$\bar{y}_i^{[j]} = \begin{cases} \alpha_j \bar{z}^{(q+1)} + (1 - \alpha_j)\bar{y}, & \text{if} \quad \bar{y}_i^{[j]} \bar{\theta}_i^{(q)} > 0 \text{ and } i \in \mathcal{I}^{(q)} \ , \\ 0, & \text{elsewhere.} \end{cases}$$

The computed $\bar{y}^{[j]}$ are all coherent with $\bar{\theta}^{(q)}$. The coefficients which flip signs on the segment $[0, \alpha_j]$ are set to 0 to keep the coherence. The next solution estimate is chosen such that

$$\bar{z}^{(q+1)} = \underset{\left\{\bar{y}^{[j]}\right\}_{j \in \mathcal{J}}}{\text{argmin}} \ \frac{1}{2} \|\bar{x} - \boldsymbol{\mathcal{D}} \bar{y}^{[j]}\|_2^2 + \lambda \langle \bar{\theta}^{(q)}, \bar{y}^{[j]} \rangle \ .$$

Algorithm 3.2 Windowed FSS

Input $X, \boldsymbol{D}$ and λ
Initialize $Z = 0$
for $q = 1 \dots N_{pass}$ **do**
  for $w = 0 \dots \frac{L-W+1}{W}$ **do**
    Select sub-signal $\mathcal{W} = \left\{ t \middle| (w-1) * 2W < t < (w+1) * 2W \right\}$
    Solve the sub-signal coefficients by restricting $\mathcal{I} \subset \mathcal{W}$ in Algorithm 3.1
  end for
end for
return Z

Convergence and Complexity

This algorithm converges to the optimal solution of (3.8). The proof of convergence was derived by Lee et al. (2007) in the vectorial case and can be adapted easily to the convolutional case. We refer the reader to this paper for details about the proof. The proof starts by showing that the solution at each iteration of the feature sign search step is guaranteed to strictly reduce the objective cost if the current solution, coherent with the support set $\mathcal{I}^{(q)}$ and sign estimate $\bar{\theta}^{(q)}$, is not optimal for (3.11). Then, they show that no pair of sign estimate and active set can be repeated during the algorithm. As there is only a finite set of these pairs, the algorithm is guaranteed to converge. This proof does not state any convergence rate and the convergence can be very slow as the number of pairs grows exponentially with the dimension. However, the pairs that can be visited have to result in energy strictly lower than the current energy and in practice, this algorithm is able to solve reasonable scale problems.

The most computationally expensive operation for the FSS iteration is to compute the solution of (3.11) using the closed form solution (3.12). With the working set technique, this operation complexity is reduced to $\mathcal{O}\left(|\mathcal{I}^{(q)}|^3\right)$. The complexity of this algorithm is thus highly dependent of the solution sparsity. If the solution is very sparse, the size of the working set should not grow much and thus each iteration of FSS should be fast. Wohlberg (2016) showed in practice that FSS was efficient for very sparse signals or short signals. For large signals, the number of coefficients in the active set might grow bigger and the resolution of the quadratic sub problem becomes computationally too expensive. In their original work, Grosse et al. (2007) propose an more efficient extension for longer signals called Windowed FSS and described in Algorithm 3.2. This extension selects a sub-part of the signal of length $2W$ at each step and calls the FSS algorithm on this sub-signal. Then, the next windowed signal is selected by shifting the selection window by W time samples. This algorithm can be related to a cyclic block coordinate descent and converges in practice. It is necessary to make multiple pass over all the windows to ensure that the results are good enough. Empirically, Grosse et al. (2007) showed that after 2 passes, the results was slightly worse than the optimal value.

3.3.2 Fast Iterative Soft Thresholding Algorithm (FISTA)

Iterative Soft-Thresholding Algorithm

The most classical algorithm to solve ℓ_1-regularized problems such as LASSO is the Iterative Soft Thresholding Algorithm (ISTA). It was designed by Daubechies et al. (2004) and relies on a proximal gradient descent. It is straight forward to adapt this

Algorithm 3.3 Iterative Soft-Thresholding Algorithm (ISTA)

1: **Input:** dictionary D, regularization parameter λ, and tolerance ϵ
2: Initialization: $Z_k^{(0)}[t] = 0$ and $L = \max_\omega \|\widehat{\boldsymbol{D}}[\omega]\widehat{\boldsymbol{D}}[\omega]^\mathsf{T}\|_2^2$
3: **repeat**
4: **Gradient step:** Update for all $(k, t) \in \mathcal{C}$

$$U_k[t] = Z_k^{(q)}[t] - \frac{1}{L}\nabla f(Z^{(q)})_k[t]$$

5: **Soft-thresholding:** point with proximal operator for U

$$Z^{(q+1)} = \mathrm{Sh}\left(U, \frac{\lambda}{L}\right)$$

6: **until** $\left\|Z^{(q+1)} - Z^{(q)}\right\|_\infty < \epsilon$
7: **Return** $Z^{(q)}$

algorithm to the convolutional setup. The algorithm updates the current estimate $Z^{(q)}$ at iteration q using a proximal descent step for (3.8) *i.e.*

$$Z^{(q+1)} = \mathrm{Sh}\left(Z^{(q)} - \alpha\nabla h_1(Z^{(q)}), \alpha\lambda\right) \tag{3.16}$$

with $\alpha > 0$ a learning rate parameter and Sh the soft-thresholding operator. The soft-thresholding operator is defined as a coordinate-wise operator, such that applying it to the scalar $u \in \mathbb{R}$ gives

$$\mathrm{Sh}(u, \lambda) = \mathrm{sign}(u)\max(|u| - \lambda, 0) \ . \tag{3.17}$$

It is the closed form formula for the proximal operator associated to $\lambda\|\cdot\|_1$. The proximal operator extends gradient descent for convex, non-differentiable functions. For differentiable convex functions, the operator corresponds to a gradient descent step. As h_2 is a convex function, its proximal operator in Z is defined as

$$\mathrm{prox}_{h_2}(Z) = \underset{Y}{\mathrm{argmin}}\ \frac{1}{2}\|Z - Y\|_2^2 + h_2(Y) \ .$$

This minimization problem is separable on each coordinate and its solution is given by the coordinate-wise function Sh.

Accelerating ISTA with Momentum

This algorithm can be accelerated via the momentum method. In their paper, Beck & Teboulle (2009) derive an algorithm called Fast ISTA (FISTA), based on ISTA with an extra step which accelerates the convergence of the algorithm to the optimal solution of (3.8). This extra step has been developed by Nesterov (1983) and is called the Nesterov's momentum. It defines an auxiliary point Y^q by continuing in the direction of the update between points at iterations $q-1$ and q, such that

$$Y^{(q)} = Z^{(q)} + \frac{\gamma^{(q)} - 1}{\gamma^{(q+1)}}\left(Z^{(q+1)} - Z^{(q)}\right)$$

Algorithm 3.4 Fast Iterative Soft-Thresholding Algorithm (FISTA)

Input: dictionary D, regularization parameter λ, and tolerance ϵ
Initialization: $Z_k^{(0)}[t] = Y_k^{(0)}[t] = 0, L = \max_\omega \|\widehat{d}[\omega]\widehat{d}[\omega]^\mathsf{T}\|_2^2$
repeat
  Proximal gradient step: compute ISTA like update from $Y^{(q)}$

$$Z^{(q+1)} = \text{Sh}\left(Y^{(q)} - \frac{1}{L}\nabla f(Y^{(q)}), \frac{\lambda}{L}\right)$$

  Update momentum coefficient $\gamma^{(q+1)} = \dfrac{1+\sqrt{1+4\gamma^{(q)^2}}}{2}$
  Nesterov momentum step:

$$Y^{(q+1)} = Z^{(q+1)} + \frac{\gamma^{(q)}-1}{\gamma^{(q+1)}}\left(Z^{(q+1)} - Z^{(q)}\right)$$

until $\left\|Z^{(q+1)} - Z^{(q)}\right\|_\infty < \epsilon$
Return $Z^{(q)}$

with scalar $\gamma^{(q)}$ following the recursion with $\gamma^{(0)} = 1$ and $\gamma^{(q+1)} = \frac{1+\sqrt{1+4\gamma^{(q)^2}}}{2}$. The design of the γ term was derived to maximize the acceleration given by this extra step. The proximal descent update is then computed starting from this new point, using the same mechanism as in ISTA. Algorithm 3.4 summarizes this algorithm.

The explanation of why this algorithm is able to accelerate the convergence of gradient descent is complicated. An intuition of what happens can be seen when analyzing this algorithm as a dynamical system. If we consider the function we want to minimize as a bowl and assimilate our current point to a ball, the minimization can be seen as the movement of the ball toward an equilibrium point. In the gradient descent, the ball is moved as if it was starting each time with zero speed and only gravity helps it moves to the next spot. The momentum technique adds the speed of the ball in the equation and speed up the movement of the ball toward the equilibrium point, which is the minimal point of the surface defined by the cost function. A formal link with second order differential equations is established by Su et al. (2016).

Convergence and Complexity

Both ISTA and FISTA were proven to converge to the optimal solution of the LASSO in Beck & Teboulle (2009). The convergence rate of ISTA is $\mathcal{O}\left(\frac{1}{q}\right)$ and its accelerated version has a convergence rate of $\mathcal{O}\left(\frac{1}{q^2}\right)$. Their extension to the convolutional cases is really straightforward. The only change is the formula for the gradient computation. The proof of convergence and the convergence rates do no depend on the particular structure of h_1 and can also be proven for (3.8). Using FISTA to solve the convolutional sparse coding was proposed by Chalasani et al. (2013). They show that when using convolution to compute the gradient of h_1, it is possible to efficiently solve convolutional sparse coding (3.8).

Algorithm 3.5 Alternating Direction Method of Multipliers (ADMM)

1: **Input:** functions h_1, h_2, matrix $\boldsymbol{A}, \boldsymbol{B}$, vector C, parameter μ and tolerance ϵ
2: Initialization: $Y^{(0)}, \Theta^{(0)}$
3: **repeat**
4: $\quad X^{(q+1)} = \underset{X}{\operatorname{argmin}}\, h_1(X) + \frac{\mu}{2}\left\| \boldsymbol{A}X + \boldsymbol{B}Y^{(q)} - C + \frac{\Theta^{(q)}}{\mu} \right\|_2^2$
5: $\quad Y^{(q+1)} = \underset{Y}{\operatorname{argmin}}\, h_2(Y) + \frac{\mu}{2}\left\| \boldsymbol{A}X^{(q+1)} + \boldsymbol{B}Y - C + \frac{\Theta^{(q)}}{\mu} \right\|_2^2$
6: $\quad \Theta^{(q+1)} = \Theta^{(q)} + \boldsymbol{A}X^{(q+1)} + \boldsymbol{B}Y^{(q+1)} - C$
7: **until** $\max\left(\|AX^{(q+1)} + BY^{(q+1)} - C\|_2, \|\mu A^\mathsf{T} B(X^{(q+1)} - X^{(q)})\|_2 \right) < \epsilon$

The most computationally expensive operation for the FISTA updates is to compute the gradient ∇h_1 . An interesting idea proposed by Wohlberg (2016) and by Haeffele et al. (2017) is to use fast Fourier Transform (FFT) to compute it quickly. Indeed, using the Parseval theorem,

$$\left\| X - \sum_{k=1}^{K} Z_k * \boldsymbol{D}_k \right\|_2^2 = \left\| \widehat{X} - \sum_{k=1}^{K} \widehat{Z}_k \widehat{\boldsymbol{D}}_k \right\|_2^2 \tag{3.18}$$

This technique accelerates the computations for the updates at each step in FISTA. The most expensive computation is the FFT performed to obtain the Fourier transform of the elements. It has a computational cost of $\mathcal{O}\left(KT \log T\right)$.

3.3.3 Alternating Direction Method of Multiplier (ADMM)

General ADMM Algorithm

An algorithm which received much attention recently for ℓ_1 optimization is the alternating direction method of multiplier (ADMM). It was introduced for general problems by Gabay & Mercier (1976). The paper considers solving problem of the form

$$\begin{aligned} &\text{minimize } h_1(X) + h_2(Y) \\ &\text{subject to } \boldsymbol{A}X + \boldsymbol{B}Y = C \end{aligned} \tag{3.19}$$

with $X \in \mathbb{R}^{P_1}, Y \in \mathbb{R}^{P_2}, C \in \mathbb{R}^{P_3}$ and $\boldsymbol{A} \in \mathbb{R}^{P_3 \times P_1}, \boldsymbol{B} \in \mathbb{R}^{P_3 \times P_2}$ for two convex functions h_1, h_2. The resolution of this constraint optimization problem is performed using the augmented Lagrangian, defined as

$$\mathcal{L}(X, Y, \Theta, \mu) = h_1(X) + h_2(Y) + \Theta^\mathsf{T}(\boldsymbol{A}X + \boldsymbol{B}Y - C) + \frac{1}{2}\mu \|\boldsymbol{A}X + \boldsymbol{B}Y - C\|_2^2 \ . \tag{3.20}$$

with $\Theta \in \mathbb{R}^{P_3}$ the dual variable of the problem. Algorithm 3.5 describes the steps of the ADMM algorithm. In a nutshell, the updates are performed alternatively on each variable of the Lagrangian to reach the optimum. Updates line 4 and line 5 minimize the Lagrangian $\mathcal{L}$ in the first two arguments and then, in line 6, the dual variable Θ is updated in order to maximize $\mathcal{L}$.

Fast Convolutional Sparse Coding (FCSC)

Bristow et al. (2013) built on this method to propose a new algorithm to solve (3.8). The idea is to re-write the minimization problem by splitting the two parts of the cost function with an auxiliary variable Y, such that

$$\begin{aligned} &\text{minimize} \quad \underbrace{\frac{1}{2}\left\| X - \sum_{k=1}^{K} Z_k * \boldsymbol{D}_k \right\|_2^2}_{h_1(Z)} + \underbrace{\lambda \left\| Y \right\|_1}_{h_2(Y)} , \\ &\text{subject to } Z = Y \ . \end{aligned} \tag{3.21}$$

The augmented Lagrangian for problem (3.21) is given by

$$\mathcal{L}(Y, Z, \Theta, \mu) = h_1(Z) + h_2(Y) + \Theta^\mathsf{T}(Y - Z) + \frac{\mu}{2}\|Y - Z\|_2^2 \tag{3.22}$$

The updates are then computed using the same steps as described in Algorithm 3.5. The steps line 5 and 6 are easy to compute in this setup. The updates can be computed using the following

$$Y^{(q+1)} = \text{Sh}\left(Z^{(q+1)} + \frac{\Theta^{(q)}}{\mu}, \frac{\lambda}{\mu} \right) \tag{3.23}$$

$$\Theta^{(q+1)} = \Theta^{(q)} + \left(Z^{(q+1)} - Y^{(q+1)} \right) \tag{3.24}$$

where Sh is the soft thresholding operator, defined in (3.17). The most expensive part is to compute the update line 4,

$$Z^{(q+1)} = \underset{Z}{\text{argmin}} \frac{1}{2}\left\| X - \sum_{k=1}^{K} Z_k * \boldsymbol{D}_k \right\|_2^2 + \frac{\mu}{2}\left\| Y^{(q)} + \frac{\Theta^{(q)}}{\mu} - Z \right\|_2^2 . \tag{3.25}$$

Using the same idea as the one to accelerate the gradient in FISTA in (3.18), we can rewrite (3.25) using the Parseval theorem

$$\widehat{Z}^{(q+1)} = \underset{Z}{\text{argmin}} \frac{1}{2}\left\| \widehat{X} - \sum_{k=1}^{K} \widehat{Z}_k \widehat{\boldsymbol{D}}_k \right\|_2^2 + \frac{\mu}{2}\left\| \widehat{Y}^{(q)} + \frac{\widehat{\Theta}^{(q)}}{\mu} - \widehat{Z} \right\|_2^2 .$$

The solution to this problem is given by the solution $\widehat{Z}$ of the linear system

$$\left(\widehat{\boldsymbol{D}}^H \widehat{\boldsymbol{D}} + \mu \boldsymbol{I} \right) \widehat{Z} = \widehat{\boldsymbol{D}}^H \widehat{X} + \mu \left(\widehat{Y}^{(q)} + \frac{\widehat{\Theta}^{(q)}}{\mu} \right) . \tag{3.26}$$

This system is composed of T independent system, which correspond to each frequency computed by the FFT and the solution of (3.25) can be retrieved using the inverse Fourier transform. The full algorithm to solve the convolutional sparse coding based on ADMM is described in Algorithm 3.6.

Algorithm 3.6 Fast convolutional Sparse Coding (FCSC)

Input: Signal X, Dictionary $\boldsymbol{D}$, parameter μ and tolerance ϵ
Initialization: $Y^{(0)} = X, \Theta^{(0)} = 0$
Precompute $\widehat{\boldsymbol{D}}$ with FFT of $\boldsymbol{D}$ with zero-padding to length T
repeat
  Compute $\widehat{Y}^{(q)}, \widehat{\Theta}^{(q)}$ with FFT of $Y^{(q)}, \Theta^{(q)}$ with zero-padding to length T
  Solve the linear system for $l \in [\![0, \frac{T}{2}[\![$ for $\widehat{Z}^{(q+1)}$

$$\left(\widehat{\boldsymbol{D}}[l]^H \widehat{\boldsymbol{D}}[l] + \mu \boldsymbol{I}_K\right) \widehat{Z}^{(q+1)}[l] = \widehat{\boldsymbol{D}}[l]^H \widehat{X}[l] + \mu \left(\widehat{Y}^{(q)}[l] + \frac{\widehat{\Theta}^{(q)}[l]}{\mu}\right)$$

  Compute inverse FFT of $\widehat{Z}^{(q+1)}$
  Update $Y^{(q+1)} = \mathrm{Sh}\left(X^{(q+1)} + \frac{\Theta^{(q)}}{\mu}, \frac{\lambda}{\mu}\right)$
  Update $\Theta^{(q+1)} = \Theta^{(q)} + \left(X^{(q+1)} - Y^{(q+1)}\right)$
until $\max\left(\|X^{(q+1)} - Y^{(q+1)}\|_2, \|X^{(q+1)} - X^{(q)}\|_2\right) < \epsilon$
Return: $Y^{(q)}$

Convergence and complexity

Gabay & Mercier (1976) showed that the ADMM algorithm converges to the optimal solution of (3.8). A detailed study of the properties of this algorithm is given in Boyd et al. (2010). This algorithm often gives an estimate with sufficient accuracy for dictionary learning within tens of iterations. Indeed, with alternate minimization, each iteration does not need to find an optimal point, but a point with medium accuracy. However, ADMM can be slow to converge to high accuracy.

For convolutional sparse coding, the computational complexity of each iteration of the ADMM is driven by the update of $Z^{(q)}$. The updates are performed with FFT, costing $\mathcal{O}\left(KT \log T\right)$, and the resolution of the $T/2$ linear systems (3.26), with cost $\mathcal{O}\left(TK^3\right)$ using the cholesky decomposition.

The value of μ in the ADMM algorithm controls the enforcement of the constraint $X = Y$. A natural extension is to have this parameter vary at each iteration, with the goal to improve the practical convergence to a good solution and to make the algorithm more robust to initialization. Rockafellar (1976) showed that for strongly monotone operator, having $\mu^{(q)} \xrightarrow[q \to +\infty]{} +\infty$ implied super-linear convergence of method of multiplier. As the convergence proof for ADMM relies on a fixed μ, it is sufficient to consider that μ becomes fixed after a certain number of iterations. The most classical scheme to scale μ is the following:

$$\mu^{(q+1)} = \min(\mu_{\max}, \tau \mu^{(q)}) \ ,$$

Algorithm 3.7 Greedy Coordinate Descent

Input: $\bar{\boldsymbol{D}}, X$, parameter $\epsilon > 0$
$\mathcal{C} = [\![1, K]\!] \times [\![0, L-1]\!]$
Initialization: $\forall (k,t) \in \mathcal{C}$,
$Z_k[t] = 0, \quad \beta_k[t] = \left(\widetilde{D}_k * X\right)[t]$
repeat
  $\forall (k,t) \in \mathcal{C},\ Z'_k[t] = \dfrac{1}{\|D_k\|_2^2} \mathrm{Sh}(\beta_k[t], \lambda)$,
  Choose $(k_0, t_0) = \underset{(k,t)\in\mathcal{C}}{\operatorname{argmax}} |\Delta Z_k[t]|$
  Update β using (3.29) and $Z_{k_0}[t_0] \leftarrow Z'_{k_0}[t_0]$
until $|\Delta Z_{k_0}[t_0]| < \epsilon$

for a given $\tau > 1$ and $\mu_{\max}$ usually fixed to 10^5. Another scaling proposed for μ is to adapt it to balance the two parts of the function. He et al. (2000) proposed to use the update rule

$$\mu^{(q+1)} = \begin{cases} \tau\mu^{(q)}, & \text{if } \|AX^{(q+1)} + BY^{(q+1)} - C\|_2 > \nu \|\mu A^\mathsf{T} B(X^{(q+1)} - X^{(q)})\|_2 \ , \\ \dfrac{1}{\tau}\mu^{(q)}, & \text{if } \|\mu A^\mathsf{T} B(X^{(q+1)} - X^{(q)})\|_2 > \nu \|AX^{(q+1)} + BY^{(q+1)} - C\|_2 \ , \\ \mu^{(q)}, & \text{elsewhere,} \end{cases}$$

where $\nu > 1$ and $\tau > 1$. Typical choices for these parameters are $\nu = 10$ and $\tau = 2$. The idea between this penalty is to balance the residuals of the primal and dual problems, keeping them withing a factor ν of one another. Once one residual becomes bigger than this factor, the weight of the associated problem is increased by properly scaling the multiplier ν. This simple scheme works well in practice.

3.3.4 Convolutional Coordinate Descent (CD)

Another classical algorithm for sparse coding is the coordinate descent (CD). This method was first proposed specifically for LASSO problem by Wang et al. (2007) and then described in a unified framework by Friedman et al. (2007). Based on this seminal work, multiple extensions and variant have been developed. The core idea for this algorithms is the following framework:

1. Select a coordinate to update,
2. Update only this coordinate.

The different choices for steps 1 and 2 are critical and should be chosen based on the optimization problem at hand. The key idea in coordinate descent is that updating one coordinate of the solution is computationally cheap and if the solution is sparse, only few coordinates should be updated as most of them are 0.

Coordinate Update

For the second step of the procedure, two schemes have been proposed. Given a coordinate (k_0, t_0) to update in the current solution, the simplest update is to use a proximal gradient descent step for the cost function reduced to this coordinate. This

update strategy uses a learning rate $\alpha > 0$ given as a parameter to perform the gradient descent and performs the following step

$$Z_{k_0}[t_0] = \mathrm{Sh}\left(x_{k_0}[t_0] - \alpha\left(\nabla h_1(Z^{(q)})\right)_{k_0}[t_0]\right) \tag{3.27}$$

This strategy allows the practitioner to adapt the step size algorithm, which can be critical in some application. In most cases, using the inverse of the coordinate-wise Lipschitz constant is a good choice. In the case of the convolutional sparse coding, it is also possible to compute the optimal value of a given coordinate if all the other are fixed. The problem in coordinate (k_0, t_0) admits a closed form solution, which can be used to replace the value of the updated coefficient,

$$Z'_{k_0}[t_0] = \frac{1}{\|D_{k_0}\|_2^2}\mathrm{Sh}(\beta_{k_0}[t_0], \lambda) \tag{3.28}$$

with

$$\beta_k[t] = \left(\widetilde{D}_k * \left(X - \sum_{\substack{k'=1 \\ k' \neq k}}^{K} Z_{k'} * D_{k'} - \Phi_t\left(Z_k\right) * D_k\right)\right)[t] \ .$$

The coefficient is then updated to its optimal value $Z'_{k_0}[t_0]$.

Coordinate Selection

The first coordinate selection strategy proposed for this algorithm is to cycle through all the coordinates (Friedman et al., 2007). The coordinates are all updated, once at a time, before a new pass is made. Shalev-Shwartz & Tewari (2009) proposed another selection scheme picking coordinates at random. Different sampling strategies have been proposed but the most common one is the uniform strategy. For these two methods, the choice of coordinate is computationally inexpensive as it can be made independently of the current point.

Osher & Li (2009) proposed another idea for the selection step, aiming to maximize the cost function descent. A good proxy to the cost change induced by one coefficient update is to choose the coefficient which would be changed the most by the update, *i.e.* if for any coefficient (k, t), the current value $Z_k[t]$ would be updated to $U_k[t]$, then the chosen update coefficient (k_0, t_0) is

$$(k_0, t_0) = \underset{(k,t) \in [\![1,K]\!] \times [\![0,T-1]\!]}{\mathrm{argmin}} \left|Z_k[t] - U_k[t]\right| \ .$$

If the updates are computed using the proximal gradient descent (3.27), this method chooses the coefficient with the maximal gradient $\left|\left(\nabla h_1(Z^{(q)})\right)_k[t]\right|$, up to the soft thresholding border effect. When the update is done using the optimal value of the coefficient, the updated coefficient is the one the farthest from its optimal value. This strategy, tagged as greedy, is efficient in the context of sparse coding as it focuses on coordinates which have high chances to be non-zero. The drawback is that computing the updates is more expensive than the previous methods. Moreover, if the greedy selection is computed naively, the cost of one update can be as expensive as computing the full gradient. In this case, this method is obviously less efficient than gradient based method as the full gradient is computed but only one coordinate is updated, leading to slower algorithm.

Convergence and Complexity

Tseng (1988) shows the convergence of coordinate-wise maximization to the optimal solution of concave maximization problems of the form

$$\max_{x \in \mathbb{R}^n} f(x) + \sum_{i=1}^{n} g_i(x_i)$$

with f concave and differentiable and the g_i concave using cyclic updates of the coordinates x_i. Osher & Li (2009) give a proof of the algorithm convergence for greedy updates and Nesterov (2012) for the randomized updates. In addition, the latter also shows that the convergence rate of both coordinate selection schemes is $\mathcal{O}\left(\frac{1}{q}\right)$ for general convex and differentiable function f. In their work, Nutini et al. (2015) discuss the convergence rate of greedy algorithm in several settings and show that for strongly convex function f, the greedy updates converge faster to the solution than their randomized counter part, with better constants. It is not clear whether their finding can be extended to non strongly convex f, as it is the case in the convolutional sparse coding setting.

Another important aspect of comparison between those methods is the complexity of each iteration. Computing the new value for the updated coordinate, for both (3.27) and (3.28), has the same complexity of $\mathcal{O}\left(KW\right)$, obtained by maintaining the auxiliary variables ζ or β after each update (see below). For coordinate selection, the computational cost of choosing a random coordinate is $\mathcal{O}\left(1\right)$ whereas selecting the maximal coordinate is $\mathcal{O}\left(KT\right)$. When choosing a variant of the coordinate descent, there is a tradeoff between the computational cost of each update, larger for the greedy coordinate selection, and the convergence rate of the coordinate selection, slower for random coordinate selection. For convolutional coordinate descent with very sparse coding signals, the size of the problem is very large and randomized coordinate descent have low chance of selecting coordinates that are relevant compared to greedy coordinate descent. In practice, we observe that a greedy coordinate descent is quicker for these problems with a convergence to sparser solution.

When dealing with sparse problem, only a few coordinates are really important. A line of methods have been developed to take advantage of this fact by screening out coefficients that are supposed to be 0 at the optimal solution. These methods are called screening and are very efficient to improve the computational complexity of coordinate descent methods. The screening idea was introduced in the seminal work of El Ghaoui et al. (2012), which proposes *safe-rules* to screen out variables assured to be 0 in the optimal solution. Tibshirani et al. (2012) proposed less safe rules, tagged as *strong-rules* which are more aggressive and might wrongly disregard some coordinates that needs to be recovered in a post-processing step. Recently, Fercoq et al. (2015) proposed another set of rules which screens out more coordinates than the safe-rules but is still assured to only screen out coordinates that are null at the optimum.

Computing Greedy Updates Efficiently

The success of the greedy updates highly depends on the efficiency to compute the coordinate update. For problem (3.8), Kavukcuoglu et al. (2010) show that if at iteration q, the coefficient (k_0, t_0) is updated from $Z_{k_0}[t_0]$ to a value $Z'_{k_0}[t_0]$, by denoting $\Delta Z^{(q)} =$

Methods	Original paper	Dictionary adapation
APGD	Nesterov 1983	Kavukcuoglu et al. 2010
Block CD	Luo & Tseng 1993	Mairal et al. 2010
K-SVD	Aharon et al. 2006	Yellin et al. 2017
ADMM	Gabay & Mercier 1976	Bristow et al. 2013

Table 3.2: Algorithms for dictionary update

$Z_{k_0}[t_0] - Z'_{k_0}[t_0]$, β is updated with

$$\beta_k^{(q+1)}[t] = \beta_k^{(q)}[t] - \mathcal{S}_{k,k_0}[t - t_0] \left(Z_{k_0}[t_0] - Z'_{k_0}[t_0] \right), \qquad \forall (k,t) \neq (k_0, t_0) \tag{3.29}$$

with $\mathcal{S}_{k,l}[t] = (\widetilde{D_k} * D_l)[t]$. For all $t \notin [\![-W+1, W-1]\!]$, $\mathcal{S}[t]$ is zero. Thus, only $\mathcal{O}\left(KW\right)$ operations are needed to maintain β up to date with the current estimate Z. Finally, the complexity of an iteration of CD is dominated by the $\mathcal{O}\left(KT\right)$ operations needed to find the maximum of $|\Delta Z_k[t]|$.

Note that for the updates using proximal gradient descent for one coordinate, it is also possible to maintain the current gradient value with the same complexity. We will denote $\zeta^{(q)} = \nabla h_1(Z^{(q)})$, the gradient at iteration q. If coordinate (k_0, t_0) is changed from $Z_{k_0}[t_0]$ to $Z'_{k_0}[t_0]$, ζ is updated by

$$\zeta_k^{(q+1)}[t] = \zeta_k^{(q)}[t] - \mathcal{S}_{k,k_0}[t - t_0] \left(Z_{k_0}[t_0] - z'_{k_0}[t_0] \right), \qquad \forall (k,t) \in \mathcal{C} \tag{3.30}$$

This update rule is very close to the one in (3.29) except that all the coordinates are updated this time, even (k_0, t_0). After a coordinate update, the gradient can be maintained using the same number of operation $\mathcal{O}\left(KW\right)$.

3.4 Dictionary updates

Let $\left(X^{[1]}, \dots, X^{[N]}\right)$ be a set of signals in $\mathcal{X}_T^P$ and $\left(Z^{[1]}, \dots, Z^{[N]}\right)$ the associated convolutional sparse codes. The problem of learning a dictionary on this data set is posed using the following minimization problem,

$$\boldsymbol{D}^* = \underset{\boldsymbol{D}_k \in \Omega}{\operatorname{argmin}} \underbrace{\frac{1}{N} \sum_{n=1}^{N} \frac{1}{2} \left\| X^{[n]} - \sum_{k=1}^{K} Z_k^{[n]} * \boldsymbol{D}_k \right\|_2^2}_{G_N(\boldsymbol{D})}, \tag{3.31}$$

for a given constraint set Ω. Conversely to the sparse coding, here the codes are fixed and we update the dictionary elements. The most common constraint imposed on the dictionary atoms is to have a unit norm. Indeed, scaling $\boldsymbol{D}$ by α and Z by $1/\alpha$, for $\alpha > 1$, does not change the reconstruction cost but the ℓ_1-norm is decreased by a factor $1/\alpha$. Thus, without the unit norm constraint, Z tends to 0 and the norm of the atoms explodes. Other constraints have also been proposed, such as smoothness constraints enforced by regularizing the gradient with its ℓ_2-norm. Problem (3.31) is smooth and convex if Ω is convex. Table 3.2 summarize three algorithms usually used to compute $\boldsymbol{D}^*$.

Algorithm 3.8 Proximal Gradient Descent

Input: initial dictionary $\boldsymbol{D}^{(0)}$, regularization parameter λ, signals X, Z and tolerance ϵ
repeat
  $\boldsymbol{D}' = \boldsymbol{D}^{(q)} - \frac{1}{L}\nabla G_N(\boldsymbol{D}^{(q)})$ ▶Gradient step
  $\boldsymbol{D}^{(q+1)} = \text{proj}_\Omega(\boldsymbol{D}')$ ▶Proximal operator
until $\max_{k,t\in\mathcal{C}} |\boldsymbol{D}_k^{(q+1)}[t] - \boldsymbol{D}_k^{(q)}[t]| < \epsilon$
Return: $\boldsymbol{D}^{(q)}$

3.4.1 Proximal Gradient Descent

If Ω is convex, it is possible to solve (3.31) using a proximal gradient descent. Indeed, if $\mathcal{I}_\Omega$ denotes the indicator function of the constraint set Ω, (3.31) is equivalent to

$$\underset{D_k}{\text{argmin}} \quad G_N(\boldsymbol{D}) + \mathcal{I}_\Omega(\boldsymbol{D})$$

The proximal operator of $\mathcal{I}_\Omega$ is the projector proj_Ω on Ω. If Ω is the unit ball, this proximal operator is separable for each atom and can be computed using the following close form,

$$\text{proj}_\Omega(\boldsymbol{D}_k) = \frac{\boldsymbol{D}_k}{\max\left(\|\boldsymbol{D}_k\|_2, 1\right)} \ .$$

The proximal gradient descent is recalled in Algorithm 3.8. At each iteration, a gradient step is performed for the smooth and convex function G_N. Then, we use the proximal operator of $\mathcal{I}_\Omega$ to compute the next point.

Algorithm 3.9 Accelerated Proximal Gradient Descent

Input: initial dictionary $\boldsymbol{D}^{(0)}$, regularization parameter λ, signals X, Z and tolerance ϵ
Initialize: $A^{(0)} = \boldsymbol{D}^{(0)}$ and $\gamma^{(0)} = 1$
repeat
  Gradient step:

$$\boldsymbol{D}^{(q+1)} = \text{proj}_\Omega(\boldsymbol{A}^{(q)} - \frac{1}{L}\nabla G_N(\boldsymbol{A}^{(q)}))$$

  Update momentum coefficient $\gamma^{(q+1)} = \dfrac{1 + \sqrt{1 + 4{\gamma^{(q)}}^2}}{2}$
  Nesterov momentum step:

$$\boldsymbol{A}^{(q+1)} = \boldsymbol{D}^{(q+1)} + \frac{\gamma^{(q)} - 1}{\gamma^{(q+1)}}\left(\boldsymbol{D}^{(q+1)} - \boldsymbol{D}^{(q)}\right)$$

until $\max_{k,t\in\mathcal{C}} |\boldsymbol{D}_k^{(q+1)}[t] - \boldsymbol{D}_k^{(q)}[t]| < \epsilon$
Return: $\boldsymbol{D}^{(q)}$

Like ISTA, this algorithm can be accelerated using the Nesterov momentum (as described in Subsection 3.3.2). Algorithm 3.9 summarizes the Accelerated Proximal Gradient Descent (APGD). This acceleration uses an auxiliary point $\boldsymbol{A}$ computed by continuing in the direction of the update between two iterations.

Algorithm 3.10 Block Coordinate Descent

Input: initial dictionary $\boldsymbol{D}^{(0)}$, signals A and B defined in (3.32) and tolerance ϵ
Pre-compute $L_k = A_{k,k}[0]$
repeat
  for k $= 1 \dots$ K **do**
    Compute $\frac{1}{L_k}\nabla G_N(\boldsymbol{D}^{(q)})_k$ using (3.33)
    $\boldsymbol{D}'_k[t] = \boldsymbol{D}_k^{(q)}[t] - \frac{1}{L_k}\nabla G_N(\boldsymbol{D}^{(q)})_k[t]$ ▶Gradient step on coordinate k
    $\boldsymbol{D}_k^{(q+1)} = \text{proj}_\Omega(\boldsymbol{D}'_k)$ ▶Proximal operator for $\boldsymbol{D}'_k$
  end for
until $\max_{k,t\in\mathcal{C}} |\boldsymbol{D}_k^{(q+1)}[t] - \boldsymbol{D}_k^{(q)}[t]| < \epsilon$
Return: $\boldsymbol{D}^{(q)}$

3.4.2 Block coordinate Descent

Mairal et al. (2010) proposed another algorithm to solve (3.31) based on the block coordinate descent. The block coordinate descent updates at each iteration one of the dictionary atoms to minimize the objective function relatively to this element with all the other fixed. The atoms are updated using the coordinate-wise proximal gradient descent step. The main difference with Algorithm 3.8 is that it is only necessary to compute the gradient for one atom at a time, making the iteration more efficient. The proposed method uses cyclic updates for the atoms but can easily be extended to randomly choose the atom. Algorithm 3.10 describe the algorithm.

An important observation for this algorithm is the method to compute the coordinate-wise gradient. The gradient of G_N can be easily computed when using these two constants

$$\begin{aligned} A_{k,l}[t] &= \frac{1}{N}\sum_{n=1}^{N}\left(\widetilde{Z}_k^{[n]} * Z_l^{[n]}\right)[t] \\ B_k[t] &= \frac{1}{N}\sum_{n=1}^{N}\left(\widetilde{Z}_k^{[n]} * X^{[n]}\right)[t] \end{aligned} \qquad \forall t \in [\![-(W-1), W-1]\!] \tag{3.32}$$

These two constants are simply the sum of the auto-correlation and cross-correlation of X and Z, for a given shift t. Using this constants, it is possible to compute the gradient without using the signals Z and X, *i.e.*

$$\nabla G_N(\boldsymbol{D})_k[t] = B_k[t] - \sum_{l=1}^{K}\left(A_{k,l} * \boldsymbol{D}_l\right)[t] \ . \tag{3.33}$$

3.4.3 K-SVD

When the constraint set Ω is the ℓ_2 ball, it is possible to compute the dictionary updates using the K-SVD algorithm. In their paper, Aharon et al. (2006) propose a technique based on the computation of K Singular Value Decomposition to update the dictionary. This algorithm can be seen as an extension of the K-Means algorithm and it has been adapted for convolutional dictionary learning in Yellin et al. (2017). For each dictionary

Algorithm 3.11 K-SVD

Input: estimate of the sparse code Z, initial dictionary $\boldsymbol{D}^{(0)}$
Initialize $\boldsymbol{D}$ to 0
for k = 1 ... K **do**
  Compute $R_k = X - \sum_{\substack{l=1 \\ l \neq k}}^{K} \boldsymbol{D}_l^{(0)} * Z_l$.
  For $t \in [\![0, L-1]\!]$ such that $Z_k[t] \neq 0$,
  collect the sub-series $(R_k[t], \dots, R_k[t+W-1])$ in a matrix A_k. such that $Z_k[t] \neq 0$,
  Compute the first singular vectors (u, v) of A_k
  Set $\boldsymbol{D}_k = v$
end for
Return: $\boldsymbol{D}$

element $\boldsymbol{D}_k$, the residual signal

$$R_k^{[n]} = X^{[n]} - \sum_{\substack{l=1 \\ l \neq k}}^{K} \boldsymbol{D}_l * Z_l^{[n]}$$

is computed without the atom $\boldsymbol{D}_k$. Using this notation, the problem of minimizing G_N with respect to $\boldsymbol{D}_k$ can be re-written as

$$\underset{\|\boldsymbol{D}_k\|_2=1}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} \left\| R_k^{[n]} - \boldsymbol{D}_k * Z_k^{[n]} \right\|_2^2$$

The idea in K-SVD is to use the fixed support of $Z_k^{[n]}$ and to solve this problem using the SVD. To that purpose, we select the segments in the residuals which are activated in the signals $Z_k^{[n]}$, *i.e.* the segment $(R_k^{[n]}[t], \dots R_k^{[n]}[t+W-1])$ for $(t, n) \in [\![0, L-1]\!] \times [\![1, N]\!]$ such that $Z_k^{[n]}[t] \neq 0$. All these segments are used to create a matrix A_k where each line is one of the segment. Then, the first singular vectors (u, v) of A_k are computed using the SVD of A_k and the value of $\boldsymbol{D}_k$ can be updated to v. Note that the value of u can also be used to update the non-zero values of Z_k. The algorithm is summarized in Algorithm 3.11.

One advantage of this method is that the dictionary can be updated simultaneously with the non-zero coefficients of Z. This ensures that the cost function can only decrease when using an ℓ_0 penalization of the coding signal. For the ℓ_1-norm, this property is not verified anymore, but the sparsity of the activation vectors cannot be reduced.

3.4.4 Alternate Direction Method of Multiplier (ADMM)

Finally, Bristow et al. (2013) proposed a method for the dictionary updates based on the ADMM. The basic idea behind this method is to split the cost function between two variables and to constrain these variables to be equal as described in Subsection 3.3.3. For problem (3.31), this gives

$$\underset{\boldsymbol{D}=\boldsymbol{D}'}{\operatorname{argmin}} G_N(\boldsymbol{D}) + \mathcal{I}_\Omega(\boldsymbol{D}') \tag{3.34}$$

Algorithm 3.12 ADMM for dictionary update

Input: initial dictionary $\boldsymbol{D}^{(0)}$, signal $X^{[n]}$ and $Z^{[n]}$, parameter μ_d and tolerance ϵ
Initialize $\Theta_d^{(0)}$ to 0 and $\boldsymbol{D}'^{(0)} = \boldsymbol{D}^{(0)}$
repeat
  Compute $\boldsymbol{D}^{(q+1)}$ with (3.35)
  Compute $\boldsymbol{D}'^{(q+1)}$ with (3.36)
  Compute $\Theta_d^{(q+1)}$ with (3.37)
until $\max\left(\|\boldsymbol{D}^{(q+1)} + \boldsymbol{D}'^{(q+1)} - C\|_2, \|\boldsymbol{D}^{(q+1)} - \boldsymbol{D}^{(q)})\|_2\right) < \epsilon$
Return: $\boldsymbol{D}'^{(q)}$

The augmented Lagrangian of (3.34) is

$$\mathcal{L}(\boldsymbol{D}, \boldsymbol{D}', \Theta_d, \mu_d) = G_N(\boldsymbol{D}) + \mathcal{I}_\Omega(\boldsymbol{D}') + \Theta_d^\mathsf{T}(\boldsymbol{D} - \boldsymbol{D}') + \mu_d \|\boldsymbol{D} - \boldsymbol{D}' + \Theta_d\|_2^2$$

Then, the ADMM algorithm optimizes $\mathcal{L}$ for each variable iteratively. In the case of (3.31), the following updates are performed,

$$\boldsymbol{D}^{(q+1)} = \underset{\boldsymbol{D}}{\operatorname{argmin}}\, G_N(\boldsymbol{D}) + \mu_d \left\| \boldsymbol{D} - \boldsymbol{D}'^{(q)} + \frac{\Theta_d^{(q)}}{\mu_d} \right\|_2^2 \tag{3.35}$$

$$\boldsymbol{D}'^{(q+1)} = \operatorname{proj}_\Omega \left(\boldsymbol{D}^{(q+1)} + \frac{\Theta_d^{(q)}}{\mu_d} \right) \tag{3.36}$$

$$\Theta_d^{(q+1)} = \Theta_d^{(q)} + \boldsymbol{D}^{(q+1)} - \boldsymbol{D}'^{(q+1)} \tag{3.37}$$

As for the convolutional sparse coding, the update (3.35) is the most expensive to compute. Using the same idea as for the computation of the of the updates of Z in (3.25), we can use Fourier domain to show that $\boldsymbol{D}^{(q+1)}$ is the inverse Fourier transform of the solution $\widehat{\boldsymbol{D}}^{(q+1)}$ of the linear system

$$\left(\sum_{n=1}^{N} \widehat{Z}^{[n]}[l]^H \widehat{Z}^{[n]}[l] + \mu_d \boldsymbol{I}_K \right) \widehat{\boldsymbol{D}}[l] = \left(\sum_{n=1}^{N} \widehat{Z}^{[n]}[l]^H \widehat{X}^{[n]}[l] + \mu_d \widehat{\boldsymbol{D}'}[l] + \widehat{\Theta}_d[l] \right) ,$$

for l in $[\![0, T/2]\!]$ and for Fourier Transforms computed with zero-padding to length T of Z and $\boldsymbol{D}_k$. As $\boldsymbol{D}'$ is projected on the constraint set, it is the one which should be returned at the end of the algorithm.

One advantage of this algorithm is that it is easy to use with Algorithm 3.5. Indeed, the ADMM algorithm can be used for the full dictionary learning problem (3.3). The updates are performed following the equations (3.23),(3.24), (3.25), (3.35), (3.36) and (3.37) sequentially. This algorithm can be reduced to using one iteration of Algorithm 3.6 followed by one iteration of Algorithm 3.12.

4

Interpretability of the Singular Spectrum Analysis

"This is not a pipe."

– René Magritte

Contents

In this chapter, we describe the Singular Spectrum Analysis (SSA) in the convolutional representation framework and discuss the properties of the learned dictionary. This novel description of the SSA shows that it can be used to solve the non-convex optimization problem for dictionary learning with null regularization and orthonormal dictionary elements. Then, we derived a general framework to automatize SSA grouping step, which is crucial for interpretability of the components extracted by this method. This framework is used to compare different grouping strategies and to highlight their properties. These grouping techniques are used in Chapter 10 to capture the trend of oculometric signals.

4.1 Analyzing Short and Noisy Time Series

In their paper, Vautard & Ghil (1989) introduced Singular Spectrum Analysis (SSA), a technique based on the study of sub-series of a signal to decompose it as a sum of meaningful components. These components can be linked to the trend and seasonality of the studied series. The technique presents several advantages for the treatment of heterogeneous series. We will present the main steps involved in the decomposition and give hints on its usage as a time series representation.

The main idea behind the SSA is to extract from the signal a family of patterns explaining the variation of the sub-series of the signal. Then, the signal is decomposed as a sum of components linked to these extracted patterns. Vautard & Ghil (1989) proposed to use Singular Value Decomposition (SVD) to extract patterns explaining the variance of the signal. They studied the resulting components, which have interpretable role in the signal. The analyses of the components and their Fourier spectrum reveal that the components are linked to either the trend, the seasonality or the noise in the original signal. The resulting components have notably been used to study meteorological data in the original paper.

In practice, it computes a low-rank approximation of the sub-series of the studied signal. This low-rank approximation is computed using the same technique used inthe Principal Component Analysis (PCA, Hotelling 1933). The pattern extracted to construct the low-rank approximation form a good basis to represent the signal as they capture its main variation sources in the signal.

The patterns extracted with the SSA can be interpreted using the convolutional representation presented in Section 3.1. The study of the sub-series in the signal reveals the local structures of the signal, based on the extraction of patterns from the signal. The link between the two representations is made clear in Section 4.4 and the particular constraints imposed on the dictionary are explained. By design, the SSA does not find a sparse representation and the atoms are not always interpretable. But this efficient method can be used to initialize a convolutional dictionary learning algorithm.

The rest of this chapter will be organized as follows. The SSA method is presented in Section 4.2 and its known properties are exposed in Section 4.3. Section 4.4 highlights the link between the SSA and convolutional representation and propose to use the SSA as an initialization for convolutional dictionary learning. Based on the properties of the SSA, we also investigate in Section 4.5 strategies to automatize the grouping of the SSA components, making the representation learned with SSA more interpretable.

4.2 Singular Spectrum Analysis (SSA)

The SSA is composed of three steps. First, low-rank patterns that can capture the variance of the signal are computed. Then, the signal is decomposed as a sum of series linked to these patterns. Finally, a grouping step is used to clean up the decomposition in order to obtain informative components.

4.2.1 Embedding the Series in a Low-rank Space

Trajectory matrix

We consider a discrete signal $X \in \mathcal{X}_T^1$ in $\mathbb{R}$. The W-lagged matrix $\boldsymbol{X}^{(W)} \in \mathbb{R}^{L \times W}$ is defined for $W < T/2$, with $L = T - W + 1$ such that

$$\boldsymbol{X}^{(W)} = \begin{bmatrix} X[0] & X[1] & \dots & X[W-1] \\ X[1] & X[2] & \dots & X[W] \\ \dots & & & \\ X[T-W-1] & X[T-W] & \dots & X[T-1] \end{bmatrix} .$$

Remark that the rows of this matrix contain all the sub-sequences of length W which can be extracted from x. The matrix $\boldsymbol{X}^{(W)}$ contains L examples of patterns of the signal and it can be analyzed with the statistical tools to extract patterns of interest. The idea of the SSA is to learn from these L sub-series the patterns which best explain the variance of the signal. The analysis of the singular vectors of $\boldsymbol{X}^{(W)}$ determines the patterns suited to approximate the patterns with low-rank, capturing most of the variations within these samples. The order in this matrix plays an important role in the analysis as it determines the number of patterns needed to correctly approximate the signal.

Matrix Decomposition

The patterns to encode the series are extracted from $\boldsymbol{X}^{(W)}$ using the Singular Value Decomposition (SVD). This decomposition factorizes $\boldsymbol{X}^{(W)}$ as

$$\boldsymbol{X}^{(W)} = U \Lambda V^{\mathsf{T}}$$

with $U \in \mathcal{O}_L, V \in \mathcal{O}_W$ two orthogonal matrices and Λ a diagonal matrix in $\mathbb{R}^{L \times W}$ containing on its first diagonal the singular values of $\boldsymbol{X}^{(W)}$, $\{\lambda_1 \geq \dots \geq \lambda_W\}$. By construction, $W \leq L$ and this decomposition can be re-written as a sum of W rank 1 matrices, such that

$$\boldsymbol{X}^{(W)} = \sum_{k=1}^{W} \lambda_k U_k V_k^{\mathsf{T}} \tag{4.1}$$

with V_k the rows of V in $\mathbb{R}^W$ and U_k the rows of U in $\mathbb{R}^L$. This sum of low-rank matrices is linked to the best low-rank approximation of the W-lagged trajectory matrix. The optimal approximation of rank $1 \leq K \leq W$ of $\boldsymbol{X}^{(W)}$ is given by the sum of the K first terms of this sum, *i.e.*

$$\underset{\text{rank}(Y^{[K]})=K}{\text{argmin}} \|\boldsymbol{X}^{(W)} - Y^{[K]}\|_2 = \sum_{k=1}^{K} \lambda_k U_k V_k^{\mathsf{T}} .$$

Another property of this decomposition is that the variance of $\boldsymbol{X}^{(W)}$ explained by each pattern V_k is directly linked to its associated singular value λ_k. In this sense, λ_k is a measure of the importance of pattern V_k to explain the variation of the sub-sequences of the signal X and it gives a natural ordering of the patterns. These computations can be linked to the Principal Component Analysis (PCA) introduced by Hotelling (1933).

In practice, the patterns are computed using a eigenvalue decomposition. Indeed, as $U \in \mathcal{O}_L$, the value of V can be computed by diagonalizing the matrix ${\boldsymbol{X}^{(W)}}^{\mathsf{T}} \boldsymbol{X}^{(W)}$, as

$${\boldsymbol{X}^{(W)}}^{\mathsf{T}} \boldsymbol{X}^{(W)} = V^{\mathsf{T}} \Lambda^2 V .$$

The singular values are then obtained by taking the square root of the computed eigenvalues and the matrix U is computed using the relation

$$U\Lambda = \boldsymbol{X}^{(W)}V \ .$$

With this computation, the complexity of this decomposition is reduced to $\mathcal{O}\left(W^3\right)$ instead of $\mathcal{O}\left(W^2L\right)$.

Note that the singular spectrum of the trajectory matrix has been largely studied in the Linear Dynamical System Literature (LDS). Indeed, for an autoregressive model (AR) of order p, the trajectory matrix is of rank 1 and its first singular vector can be used to estimate the coefficients of the AR model. For more complex models, the trajectory matrix is used in the N4SID system identication method (Van Overschee & De Moor, 1994). Also, the SSA components satisfy the Linear Recurrence Relation (LRR) models which are associated with AR models. But these two models should not be confused as they have different models for the noise (Golyandina & Korobeynikov, 2014).

4.2.2 Reconstruction

Hankelization Operator

By construction, the signal X can be recovered perfectly from the trajectory matrix $\boldsymbol{X}^{(W)}$ by averaging its values along the anti-diagonal. This is due to the specific structure of the trajectory matrices, which have their anti-diagonals constant. Matrices verifying this property are called Hankel matrices. There is a natural bijection Φ from $H_{L,W}$, the space of Hankel matrices of size $L \times W$, to the signal space $\mathcal{X}_T^1$ for $T = W + L - 1$. The approximation done by the factorization of the trajectory matrix does not preserve the Hankel property. For a given matrix $\boldsymbol{Y} \in \mathbb{R}^{L\times W}$, we would like to associate a signal $Y \in \mathcal{X}_T^1$ whose W-lagged trajectory matrix $\boldsymbol{Y}^{(W)}$ minimize the Frobenius norm with $\boldsymbol{Y}$, *i.e.*

$$\underset{\boldsymbol{Y}^{(W)}\in H_{L,W}}{\operatorname{argmin}} \ \|\boldsymbol{Y} - \boldsymbol{Y}^{(W)}\|_2^2 \ .$$

As $\boldsymbol{Y}^{(W)}$ is the trajectory matrix of Y, we have by definition $\boldsymbol{Y}^{(W)}_{i,j} = Y[t]$ for all $i \in [\![1, L]\!]$ and $j \in [\![1, W]\!]$ such that $i + j = t$. With this relation

$$\begin{aligned}\|\boldsymbol{Y} - \boldsymbol{Y}^{(W)}\|_2^2 &= \sum_{i=1}^{W}\sum_{j=1}^{L}(\boldsymbol{Y}_{i,j} - \boldsymbol{Y}^{(W)}_{i,j})^2 \ , \\ &= \sum_{t=0}^{T}\sum_{i+j=t}(\boldsymbol{Y}_{i,j} - Y[t])^2 \ .\end{aligned}$$

The first order conditions for the minimization problem lead to:

$$Y[t] = \frac{1}{W_t}\sum_{i+j=t}\boldsymbol{Y}_{i,j}$$

where W_t is the number of pair $(i, j) \in [\![1, L]\!] \times [\![1, W]\!]$ such that $i + j = W$, *i.e.*

$$W_t = \min(W, t + 1, T - t) \ .$$

From this result, we can define the operator $\mathcal{H} : \mathbb{R}^{L\times W} \to \mathcal{X}_T^1$ to recover a signal from any matrices in $\mathbb{R}^{L\times W}$, even without Hankel structure. The signal is obtained by averaging the values of the matrix along the anti-diagonals and thus this operator is linear. This operator is called the Hankelization operator (Buchstaber, 1994).

Low-rank Signals

The component obtained from the SVD decomposition of the trajectory matrix $\boldsymbol{X}^{(W)}$ can be associated to temporal signal with the reconstruction operator $\mathcal{H}$. The two previous steps provide a decomposition of the trajectory matrix as a sum of rank one matrices. Let $Y^{(k)}$ denotes the component obtained by reconstructing the signal associated to the k-th rank one matrix of the SVD decomposition

$$Y^{(k)} = \mathcal{H}(\lambda_k U_k V_k^{\mathsf{T}}) \ . \tag{4.2}$$

By linearity of the Hankelization operator, the decomposition (4.1) can be re-written as a sum of temporal signals linked to the spectrum of the trajectory matrix *i.e.*

$$X = \mathcal{H}(\boldsymbol{X}^{(W)}) = \sum_{k=1}^{W} \mathcal{H}(\lambda_k U_k V_k^{\mathsf{T}}) = \sum_{k=1}^{W} Y^{(k)}$$

The signal X is decomposed into a sum of additive component with rank one trajectory matrices that are linked to the singular values λ_k of the matrix.

Remark 4.1. *Note that this procedure is split in two distinct steps. In the first one, a low rank approximation of the sub-series subspace is computed and then, the obtained components are projected on the space of Hankel matrices. A possibly more efficient procedure would be to directly compute low-rank components under the Hankel constraint. For instance, using the ADMM would allow simply require to use the hankelization procedure and the low rank approximation with the SVD with iterative matrices in order to compute a matrix which is both low-rank and has Hankel structure.*

4.2.3 Grouping

A reconstruction based on the W computed components retrieves perfectly the trajectory matrix $\boldsymbol{X}^{(W)}$ and thus the signal x. But to be informative, the decomposition needs to select the interpretable information. The last components of the decomposition, tied to the smallest eigenvalues λ_k, are likely to be noise present in the series. Indeed, the noise is not repeated in the W sub-series and thus, capturing the noise should explain less variance of the series than capturing the other patterns. Moreover, the learned patterns V_k can be redundant as it can be seen in Figure 4.1. After the decomposition is computed, an extra step called grouping is necessary to clean and select the relevant information and make the representation interpretable.

The grouping step consists in applying a chosen heuristic to create a partition $(I_1, \ldots, I_M)$, with M set $I_m \subset [\![1, W]\!]$ such that $\cup_{m=1}^{M} I_m = [\![1, W]\!]$, for which all the components $Y^{(k)}$ with $k \in I_m$ should be interpreted together. The component linked to the group I_m is denoted $X^{(m)}$ and is computed as the sum of the components in this group, *i.e.*

$$X^{(m)} = \sum_{k \in I_m} Y^{(k)} \ .$$

As said earlier, some components are tied to the noise of the series and are not useful for the interpretation of the signal representation. The last group I_M is used to regroup these components, to avoid having them polluting the other groups. To discard non-useful elements, a natural measure of the contribution of a component in the signal is the share of the corresponding eigenvalue linked to component k,

$$\frac{\lambda_k}{\sum_{l=1}^{W} \lambda_l} \ .$$

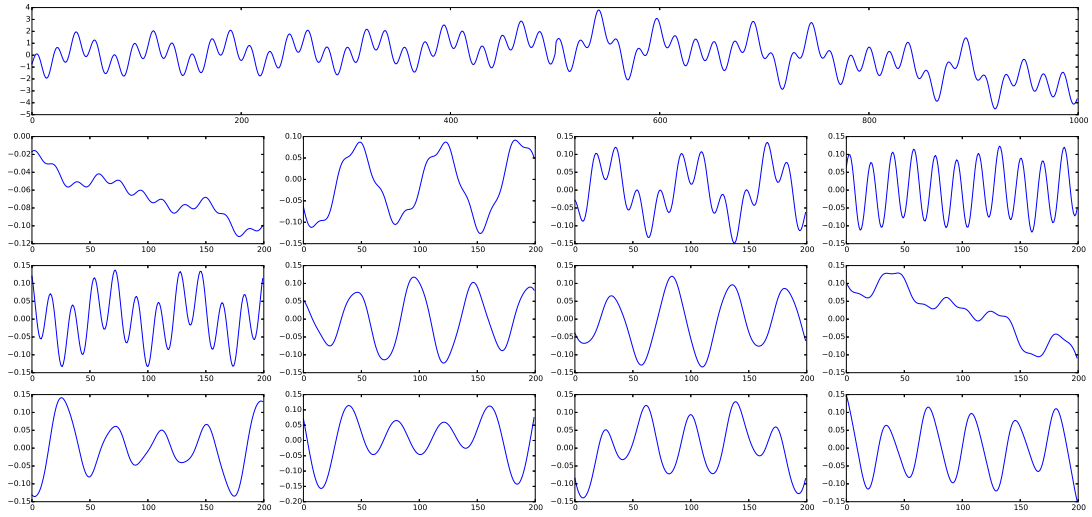


Figure 4.1: First elements of the dictionary of patterns extracted from the raw signal *(top)* ordered from left to right and top to bottom. The first and eighth components are both linked to the trend of the series. Some other components like the sixth and seventh or the ninth and tenth result from very similar phenomenons and should be interpreted together.

This measure shows the part of the series variance explained by this component. Components with a very small contribution can be discarded as they have a negligible effect on the signal, and are often associated to noise.

This step is very sensitive as it controls the quality of the decomposition. If the grouping is too wide, it would lead to components mixing different local structures of the signal, making it complicated to identify them. If it is too tight, the decomposition would fail to group components which account for the same patterns and the obtained decomposition would be less informative. In Golyandina et al. (2001), some insights are proposed to perform the grouping step, but they all require a manual selection in the end. We propose in the Section 4.5 automatized strategies to perform the grouping step in this procedure.

4.3 Properties of the SSA

In this section, we review the known separability properties of the SSA, mostly derived by Golyandina et al. (2001). It sheds light on the characteristics of the retrieved component and dictionary.

4.3.1 Notion of Separability

In Golyandina et al. (2001), the authors introduce the mathematical concept of separability to study the SSA decomposition. The decomposition obtained from the SSA is closely related to this notion. Let $X^{(1)}$ and $X^{(2)}$ denote two univariate series that we would like to separate.

Definition 4.2 (Separability)**.** *Two components $X^{(1)}$ and $X^{(2)}$ are said to be separable by SSA with window length W if there exists two subsets I_1 and I_2 of $[\![1, W]\!]$ such that $I_1 \cap I_2 = \emptyset$ and*

$$X^{(m)} = \sum_{k \in I_m} Y^{(k)} = \sum_{k \in I_m} \mathcal{H}(\lambda_k U_k V_k^{\mathsf{T}})$$

where (λ_k, U_k, V_k) are the singular values and associated singular vectors of the W-lagged trajectory matrix of $X = X^{(1)} + X^{(2)}$ and $\mathcal{H}$ is the Hankelisation operator.

The components are separable when they can be retrieved as the sum of basic components computed with the W-lagged trajectory matrix SVD. When it is not the case, there is no way to perfectly recover the original components from the one obtained with the SSA.

The $(V_k)_{k=1,\dots,W}$ form an orthogonal basis of the trajectory space $\mathcal{L}^{(W)}$, the space containing all W-lagged trajectory matrices of time series of length T. Choosing I_1, I_2 is equivalent to split this family into two groups of orthogonal basis vectors. As the patterns V_k are orthogonal to each other, the trajectory matrices of each signal have to live in the space spanned by the patterns in I_m , *i.e.* $\mathcal{L}^{(W,m)} = \text{span}\left\{V_k / k \in I_m\right\}$. The orthogonality of these two spaces implies that all sub-series of length W of $X^{(1)}$ have to be orthogonal to the sub-series of size W of $X^{(2)}$. By symmetry of the parameters W and L, the same property holds for sub-series of size L too. All the rows and columns of the W-lagged matrices linked to the separable series are orthogonal.

Proposition 4.3. *Two signals $X^{(1)}$ and $X^{(2)}$ are separable by SSA with W-length window if and only if all sub-series of length W in $X^{(1)}$ are orthogonal to sub-series of length W in $X^{(2)}$.*

This property indicates that the notion of separability is highly dependent of the choice of the parameter W. Using this characterization of the separability, a relaxed notion of the separability is defined.

Definition 4.4 (ϵ-Approximate separability)**.** *For $\epsilon > 0$, two series are said ϵ-approximately separable if all the Pearson correlation coefficients for the rows (or columns) of the trajectory matrices are close to zero i.e. for $W' \in \{W, L\}$:*

$$\frac{\sum_{l=0}^{W'-1} X^{(1)}[t_0 + \tau] X^{(2)}[t_1 + \tau]}{\sqrt{\sum_{\tau=0}^{W'-1} X^{(1)}[t_0 + \tau]^2} \sqrt{\sum_{\tau=0}^{W'-1} X^{(2)}[t_1 + \tau]^2}} < \epsilon \qquad 1 \leq t_0, t_1 \leq T - W'$$

If the two series $X^{(1)}$ and $X^{(2)}$ are separable, small perturbations can result in series which are only approximately separable. This notion keeps information on the separability of two series in a less constrained way. Also, the stability property is not stable with the window length W and the approximate separability is more robust to the choice of W.

Separability with Harmonics

These notions grasp the kind of signals that SSA can separate and the effects of the different parameters. An important case is to know under which conditions a non-zero signal $X^{(2)}$ is separable by SSA with window length W from a harmonic $X^{(1)}[t] = \cos(2\pi\omega t + \phi)$ with $0 < \omega < \frac{1}{2}$?

Proposition 4.5. *A harmonic component $X^{(1)}$ can be separated with the W-windowed SSA from a signal $X^{(2)}$ if $X^{(2)}$ has a null Fourier spectrum in ω and if the window length is a multiple of the period of $X^{(1)}$ and the period of $X^{(2)}$.*

Using Proposition 4.3, for all $0 \leq t_0, t_1 \leq T - W$,

$$\sum_{\tau=0}^{W-1} X^{(1)}[t_0+\tau]X^{(2)}[t_1+\tau] = \sum_{\tau=0}^{W-1} \cos(2\pi\omega(t_0+\tau)+\phi_1)X^{(2)}[t_1+\tau] = 0 \ .$$

By combining these equations for t_0 and t_0+1, we get

$$\begin{aligned}
& \textstyle\sum_{l=0}^{W-1} X^{(1)}[t_0+\tau]X^{(2)}[t_1+\tau] - \sum_{l=0}^{W-1} X^{(1)}[t_0+\tau+1]X^{(2)}[t_1+\tau] && = 0 \\
\Leftrightarrow \quad & \cos(2\pi\omega t_0+\phi)X^{(2)}[t_1] - \cos(2\pi\omega(t_0+W)+\phi)X^{(2)}[t_1+W] && = 0 \\
\Leftrightarrow \quad & \cos(2\pi\omega t_0+\phi)\left[X^{(2)}[t_1] - \cos(2\pi\omega W)X^2[t_1+W]\right] - \\
& \sin(2\pi\omega t_0+\phi)\sin(2\pi\omega W)X^{(2)}[t_1+W] && = 0
\end{aligned}$$

As this equality holds for all $1 \leq t_0 \leq T - W$,

$$\cos(2\pi\omega W)X^{(2)}[t_1+W] = X^{(2)}[t_1], \qquad \sin(2\pi\omega W)X^{(2)}[t_1+W] = 0, \quad \forall t_1 \in [\![0, L-1]\!] \ .$$

The second equality is valid if $2\omega W$ is an integer, as $X^{(2)}$ is non-zero. With this first hypothesis verified, *i.e.* $2\omega W$ is an integer, the first equation is reduced to a periodicity condition. The separability is only possible if $X^{(2)}$ is periodic with period T_0 and T_0 divides W. Another property which comes from the orthogonality of the length W sub-series is seen from the Fourier spectrum of component $X^{(2)}$. A simple computation gives

$$\sum_{\tau=0}^{W} e^{2i\pi\omega k} X^{(2)}[\tau] = 0 \qquad i.e. \qquad \widehat{X^{(2)}}(\omega) = 0 \ .$$

This condition is quite natural as it means that components sharing the same harmonic are not separable with the SSA.

Approximate Separability of Sums of Harmonics

In their book, Golyandina et al. (2001, chap. 6) show that the notion of approximate separation is more robust to the choice of window length W.

Proposition 4.6. *Two oscillatory signals $X^{(1)}$ and $X^{(2)}$ defined for $N_1, N_2 \in \mathbb{N}$, for $\{\omega_n^{(1)}\}_{n=1}^{N_1}, \{\omega_n^{(2)}\}_{n=1}^{N_2} \subset \mathbb{R}^+$ and $\{\phi_n^{(1)}\}_{n=1}^{N_1}, \{\phi_n^{(2)}\}_{n=1}^{N_2} \subset [0, 2\pi[$ by*

$$X^{(m)}[t] = \sum_{n=1}^{N_m} \cos(2\pi\omega_n^{(m)}t + \phi_n^{(m)})$$

are ϵ-approximately separable by SSA with window length W if they have disjoint sets of frequencies $\left\{\omega_n^{(m)}\right\}_{n=1}^{N_m}$ and if

$$\min(W, T-W) > \max\left(\frac{1}{\epsilon}, \max_{\substack{1\leq n\leq N_1,\\ 1\leq n'\leq N_2}} \frac{1}{|\omega_n^{(1)} - \omega_{n'}^{(2)}|}\right) \ .$$

Their demonstration relies on the notion of asymptotic separability, when the length of the series and the size of the window tend to infinity. They show that oscillatory components with disjoint Fourier spectrum are asymptotically separable and that the correlation between their components tends to 0 with a rate $\frac{1}{W}$.

4.3.2 Selecting a Window Length W

Resolution of SSA. The size of the window used to build the trajectory matrix has a strong impact on the decomposition provided by the SSA, as seen in Subsection 4.3.1. It determines the resolution for the algorithm. If the window is too small, all the variations would be grouped together, resulting in one component, very close to the full signal. If the size of the window is too big, the components computed with the SSA would not have an interesting meaning. Indeed, as the number of component increases, the oscillation and noise in the original signal are spread across them, making it difficult to interpret the results without a good grouping step. Also, the number of noisy components increases.

In Figure 4.2, we construct an artificial series X with a sampling rate of 1Hz as the sum of a quadratic trend component t^2 and a sinusoidal component of period $T_0 = 25$ time samples such that and use SSA with two different window lengths to decompose it.

$$X[t] = t^2 + \cos\left(2\pi\frac{t}{T_0}\right)$$

When the window length is smaller than the period T_0, the harmonic component is grouped in the same component as the trend component by the decomposition. The dictionary being learned on the windowed signal, when the window is smaller than the period, the dictionary's elements can not capture the effect in one element and are seen as global variation, linked to the trend of the signal. When W is greater than T_0, the learned patterns separate the two components in the signal and we retrieve the right decomposition.

The reason behind the link between the window length parameter W and the resolution can be seen with Proposition 4.5 and Proposition 4.6. The length of the window influences the capacity of SSA to separate harmonic components from one another. Particularly, Proposition 4.6 shows that the resolution of SSA is intimately tied to the selection of W, as harmonic components not too close with $|\omega_1 - \omega_2| > \epsilon$ are ϵ-separable if $\frac{1}{\epsilon} < W$.

Complexity. Because of the symmetry of the problem in W and $L = T - W + 1$, the window length has to be chosen in $[\![2, T/2]\!]$. Choosing a bigger window does not yield better results as the SVD of the trajectory matrix is the same for W and L. Increasing the window length also increases the resolution of the SSA and thus the separability power of the method. Also, with larger window length, the algorithm is more stable as small variations of the window size have less impact on the results. But the drawback of a large window is that it increases the computational cost of the method as the SVD is performed with a complexity $\mathcal{O}(W^3)$. Moreover, increasing the window length also creates more component in the SVD. This requires a better grouping step to ensure the interpretability of the results. There is no good heuristic to automatically tune the window length for a particular signal. The main criterion of choice is the size of the details we want to be able to detect coupled with the choice of a grouping heuristic

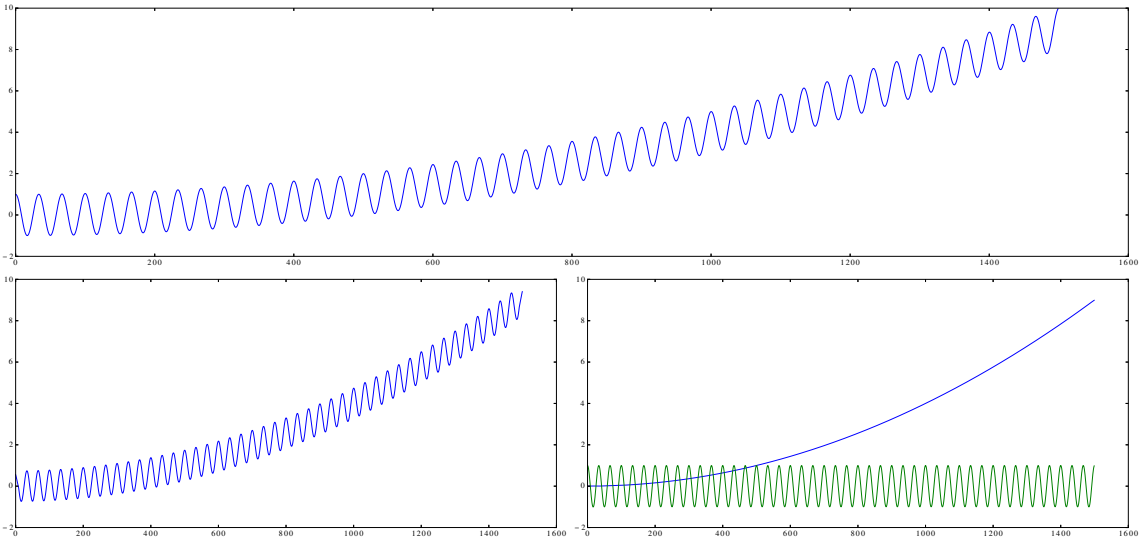


Figure 4.2: *(top)* Original serie, *(bottom)* Components extracted with a window length of *(left)* $W = 10$ and *(right)* $W = 100$

permitting to get back the meaningful components. The window length should be chosen larger than the wider high level pattern we want to be able to separate.

4.4 Initialization of the Convolutional Dictionary Learning with SSA

In this section, we show that patterns learned with the SSA can be used to initialize the convolutional dictionary learning.

4.4.1 A Convolutional Representation

The SSA computes a set of W patterns V and the associated coding vector U to represent the sub-series of length W in the signal. These patterns and code vectors can be used as a convolutional representation of the signal.

Proposition 4.7. *Let $X \in \mathcal{X}_T^1$ an univariate signal and $\{\lambda_k, U_k, V_k\}_{k=1}^W \in \mathbb{R} \times \mathbb{R}^L \times \mathbb{R}^W$ be the eigen-triple obtained with SSA. Up to the W first and last time sample, SSA computes a convolutional representation of the signal X, i.e. for all $t \in [\![W, T-W]\!]$,*

$$X[t] = \sum_{k=1}^{W} \left(Z_k * \boldsymbol{D}_k\right)[t] \ ,$$

where $Z_k[t] = \frac{\lambda_k U_{k,t+1}}{W}$ and $\boldsymbol{D}_k[t] = V_{k,t+1}$.

Proof. Using the properties of the SVD, we can write the coefficients of the trajectory matrix such that

$$X_{t,l}^{(W)} = \sum_{k=1}^{W} \lambda_k U_{k,t} V_{k,l} \ .$$

the initial signal can be recovered from the trajectory matrix $X^{(W)}$ using the hankelization operator $\mathcal{H}$, *i.e.*

$$\begin{aligned} X[t] = \mathcal{H}(X^{(W)})[t] &= \frac{1}{W_t} \sum_{i+j=t} X^{(W)}_{i,j} \ , \\ &= \frac{1}{W_t} \sum_{\tau=1}^{W_t} X^{(W)}_{1+t-\tau,\tau} \ , \\ &= \frac{1}{W_t} \sum_{\tau=1}^{W_t} \sum_{k=1}^{W} \lambda_k U_{k,1+t-\tau} V_{k,\tau} \ , \end{aligned}$$

where $W_t = \min(t, T-t, W) = \Big|\{(i,j) \in [\![1,L]\!] \times [\![1,W]\!]; i+j=t\}\Big|$ count the number of term in the t-th anti-diagonal. For $t \in [\![W, T-W]\!]$, $W_t = W$ and we can write

$$X[t] = \sum_{k=1}^{W} \sum_{\tau=1}^{W} \frac{\lambda_k U_{k,1+t-\tau}}{W} V_{k,\tau} = \sum_{k=1}^{W} \left(Z_k * \boldsymbol{D}_k\right)[t] \ ,$$

with $Z_k[t] = \frac{\lambda_k U_{k,t+1}}{W}$ and $\boldsymbol{D}_k[t] = V_{k,t+1}$. □

The set of patterns V computed with the SSA can thus be used as a dictionary to represent signals. This property shows the link between SSA and convolutional representations. This representation is dense and most of the coefficients are nonzero but the extracted patterns are suited to encode the signal. It is possible to use the extracted patterns as an initial dictionary for convolutional dictionary learning. This initialization strategy can be summarized as taking the principal components of the set of all sub-signals of length W in the training set. This strategy is similar to the one used in non-convolutional dictionary learning where the dictionary can be initialized using PCA (see for instance Mairal et al. 2012). This initialization strategy is tested in Subsection 9.3.3.

Remark 4.8. *As the computed convolutional representation is dense, the specific shapes in the original signal are lost. This is typically due to the fact that the SVD only imposes a low rank constraint on the coding components. Looking directly for sparse and low-rank activation would permit to retrieve more interesting components. In particular, recent results by Elhamifar et al. (2012) propose a method to approximate a matrix as a product of itself with a sparse and low rank coding matrix. In particular, given all the sub-series of length W, this method permits to select some of them to encode the others as sparse linear combination of the selected sub-series. The big advantage of this method is that the shape of the computed patterns is not altered by the learning and it is directly a part of the analyzed signals.*

4.4.2 Properties of the SSA Dictionary.

The SSA atoms form an orthonormal family, *i.e.* for $i, j \in [\![1, K]\!]$,

$$\sum_{\tau=0}^{W-1} D_i[\tau] D_j[\tau] = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases} \tag{4.3}$$

Using the equivalence of the temporal scalar product with its counter part in the frequency domain, the spectrum $\{\widehat{D}_k\}_{k=1,..K}$ are also orthogonal.

These patterns are the ones that best capture the variance in the length W sub-series of the original signal if they are considered independently. They can be used to initialize convolutional dictionary learning algorithms. Indeed, these patterns can be computed efficiently and they are a solution of the convolutional dictionary learning (3.3) for one signal and $\lambda = 0$. When λ is increased, it can be expected that the solution is not too unstable and that the computed dictionary is not far from a good solution for (3.3).

4.5 Automatizing the Grouping Process: a General Framework

The grouping step is a very delicate process as it controls the quality of the extracted components. Empirical results show that some learned dictionary's elements repeat the same harmonic with different phases. The ability to identify these elements is necessary to be able to interpret the representation. Several grouping elements are proposed by Golyandina et al. (2001), but all requires manual selection of the groups of components. Some recent works propose to automatize this process and this section proposes a unified framework to analyze and compare automated grouping strategies. These strategies are used in Section 10.2 to detrend eye tracker recordings in order to separate particular movements of infants' eyes from the regular gaze movement.

4.5.1 General Formulation

Automated grouping strategy can be described with 3 phases.

1. Select components $Y^{(k)}$ of interest from the SSA,
2. Compute a similarity matrix between these components. The choice of similarity measure is critical for the performance of the grouping strategy and is discussed in Subsection 4.5.2.
3. Create groups $I_m \subset [\![1, W]\!]$ of components based on the similarity matrix. It controls the relative importance of the components being grouped together. The different strategies are discussed in Subsection 4.5.3.

The final components $X^{(m)}$ are obtained by summing all components in each group I_m

$$X^{(m)} = \sum_{k \in I_m} Y^{(k)} \ . \tag{4.4}$$

Selecting the Components

The components are selected from the SVD components based on the associated singular value λ_k. The component $Y^{(k)}$ is selected if $\lambda_k > \tau$ for a given threshold $\tau > 0$. Indeed, as stated in Subsection 4.2.3, the singular value is linked to the part of variance explained by this component. A low singular value indicates a pattern less important to the interpretation of the signal. In the following, we adapt the threshold value with the singular spectrum of the trajectory matrix, such that

$$\tau = \tau_l \lambda_2 \ .$$

The second singular value is preferred to the first one. Indeed, the first singular value can be several orders of magnitude higher for some trend patterns and cause most of the components to be rejected. The second singular value λ_2 is more informative of the magnitude of the signal. Based on empirical observation, a reasonable choice for the multiplicative threshold is $\tau_l = 0.001$.

4.5.2 Similarity Measures

Correlation-based Measures

The aim of grouping is to gather components which are produced by the same phenomena. As these components are not independent, their correlation can be a good similarity indicator.

Correlation (AG1). Abalov & Gubarev (2014) propose to grouped components based on their correlation. Using this measure, two components are considered as adjacent if their correlation is greater than a threshold value ρ_c. The correlation is here taken in the sense of the Pearson coefficient, based on the inner product defined for two components $Y^{(k)}$ and $Y^{(l)}$ as

$$\langle Y^{(k)}, Y^{(l)} \rangle = \sum_{t=0}^{T-1} Y^{(k)}[t] Y^{(l)}[t] \ .$$

The norm associated to this inner product is $\|Y^{(k)}\|_2 = \sqrt{\langle Y^{(k)}, Y^{(k)} \rangle}$ for the univariate signal $Y^{(k)}$.

Definition 4.9 (Pearson correlation coefficient)**.** *For two scalar signals $Y^{(k)}$ and $Y^{(l)} \in \mathcal{X}_T^1$, the Pearson correlation coefficient is defined using the value of the inner product compared to the ℓ_2-norms of the components i.e.*

$$\text{corr}(Y^{(k)}, Y^{(l)}) = \frac{< Y^{(k)}, Y^{(l)} >}{\|Y^{(k)}\|_2 \|Y^{(l)}\|_2}$$

This similarity is computed directly using the signals reconstructed from the low-rank components. The main drawback of using correlation is that it is weak to noise. Indeed, if two components are correlated but with a lot of noise, the Pearson correlation might be small. To avoid this situation, the estimation of the correlation is performed using the full component $Y^{(k)}$ and not the associated pattern V_k . The estimation of the Pearson coefficient is more stable this way as the impact of the noise is smaller for longer series. Also, the Pearson correlation of a component of interest and a noise component can be very high, due to scale effects, as the Pearson correlation does not take the magnitude of the signal into account. To avoid including noise in the groups, components are considered similar only if they have the same order of magnitude. To control this, we compare the ratio of their associated singular value to a threshold $\rho_1 > 0$,

$$\frac{\min(\lambda_k, \lambda_l)}{\max(\lambda_k, \lambda_l)} \geq \rho_1$$

Finally, the adjacency matrix $\boldsymbol{A}$ is then defined using two parameters $\rho_c, \rho_1 > 0$ such that

$$\boldsymbol{A}_{k,l} = \begin{cases} 1 & \text{if} \quad \dfrac{\min(\lambda_k, \lambda_l)}{\max(\lambda_k, \lambda_l)} \geq \rho_1 \quad \text{and} \quad \text{corr}(Y^{(k)}, Y^{(l)}) \geq \rho_c \\ 0 & \text{elsewhere} \end{cases} .$$

W-Correlation (wCG). In the context of SSA, Golyandina et al. (2001) introduced the concept of w-correlation. This notion is based on the w-inner product

$$< Y^{(k)}, Y^{(l)} >_w = \sum_{t=0}^{T-1} W_t Y^{(k)}[t] Y^{(l)}[t]$$

with $W_t = \min(t+1, T-t, W)$. We denote $\|\cdot\|_w$ the norm associated to this inner product, such that $\|y\|_w = \sqrt{< y, y >_w}$ for a scalar signal y of length T.

Definition 4.10 (w-correlation)**.** *For two scalar signals $Y^{(k)}$ and $Y^{(l)} \in \mathcal{X}_T^1$, the w-correlation is defined similarly to the Pearson coefficient by replacing the scalar product by the w-inner product, i.e.*

$$\text{corr}_w(Y^{(k)}, Y^{(l)}) = \frac{< Y^{(k)}, Y^{(l)} >_w}{\|Y^{(k)}\|_w \|Y^{(l)}\|_w}$$

This quantity is useful as it reduces the border effect of the Pearson coefficient. When T tends to infinity, the border effect become negligible and this quantity tends to the correlation between $Y^{(k)}$ and $Y^{(l)}$. The W first and last terms of each series have less impact than the other term. This quantity is linked to the notion of separability introduced in Subsection 4.3.1. Indeed, by rewriting the w-inner product, we can see

$$\begin{aligned} < Y^{(k)}, Y^{(l)} >_w &= \sum_{t=1}^{T} \sum_{l=1}^{W_t} Y^{(k)}[t] Y^{(l)}[t] \\ &= \sum_{t=1}^{L} \sum_{l=1}^{W} Y^{(k)}[t+l] Y^{(l)}[t+l] \end{aligned}$$

In the last equation, the inner sum is the scalar product between 2 sub-series of size W. When $Y^{(k)}$ and $Y^{(l)}$ are separable, all these inner products are null, as stated in Proposition 4.3. Thus, if two series are separable using SSA, the w-correlation between them is null.

Figure 4.3 shows the correlation and w-correlation matrix between the first 20 components extracted using the SSA on a vertical accelerometer signal during the walk. The w-correlation on the right part of the figure is much more contrasted than the Pearson coefficient on the left. This is due to the border effect reduction obtained with the w-inner product.

The aim of grouping is to find components that are independent from each other, and thus that are separable, in the sense that they do not share common structures. The link between w-correlation and separability shows that using the w-correlation instead of the Pearson coefficient in **AG1** can define a good similarity metric. The similarity measure is tagged **wCG** and the adjacency matrix $\boldsymbol{A}$ is defined using the same two parameters $\rho_c, \rho_1 > 0$ such that

$$\boldsymbol{A}_{k,l} = \begin{cases} 1 & \text{if} \quad \dfrac{\min(\lambda_k, \lambda_l)}{\max(\lambda_k, \lambda_l)} \geq \rho_1 \quad \text{and} \quad \text{corr}_w(Y^{(k)}, Y^{(l)}) \geq \rho_c \\ 0 & \text{elsewhere} \end{cases}.$$

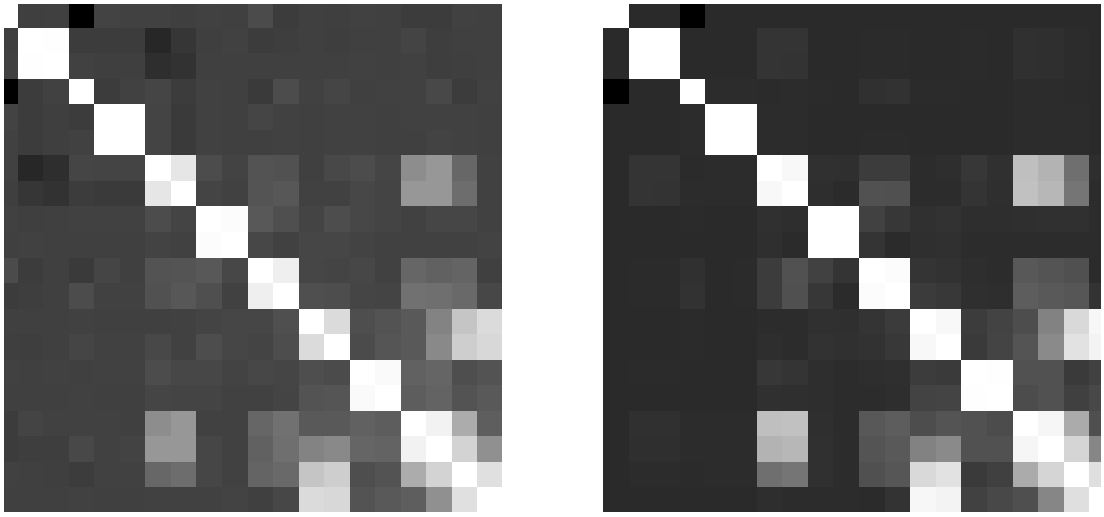


Figure 4.3: Correlation (*left*) and w-correlation (*right*) for the 20 first components of a walk signal. The w-correlation is more contrasted and it is easier to decide which components are similar.

Periodogram-based Measures

In Subsection 4.3.1, we have shown that the separability notion is tied with a certain notion of partition of the Frourier spectrum. Following this observation, Golyandina et al. (2001) recommend to compare the Fourier spectrum of the components or the eigen-vectors associated to it to derive similarity metrics, as components resulting from the same oscillatory phenomena share structures in their periodogram. In the following, for $X \in \mathcal{X}_T^1$, we denote $(\Pi_X[j])_{j=0..T-1}$ its normalized periodogram, computed using

$$\Pi_X[j] = \frac{1}{C_0} \left| \sum_{t=0}^{T-1} X[t] e^{-it\omega_j} \right|^2 ,$$

where $\omega_j = 2\pi \frac{j}{T}$ and C_0 is a normalization constant such that $C_0 = \sum_{j=0}^{T-1} \Pi_X[j] = 1$. Note that this definition does not take into account the sampling frequency as the periodogram is only used to compare signals with fixed sampling rate and length. The correct frequency in rad/seconds could be retrieved if the sampling rate T_s is known, by scaling ω_j by $\frac{T}{T_s}$.

Harmonic Grouping (HG). In their work, Alexandrov & Golyandina (2005) design an automatic grouping strategy based on the study of the periodogram to recover exponentially modulated harmonic. In their paper, they propose to use the following similarity metric between signals

$$\text{spike}\left(Y^{(k)}, Y^{(l)}\right) = \max_{0 \leq j < T} \frac{\Pi_{Y^{(k)}}[j] + \Pi_{Y^{(l)}}[j]}{2}$$

to compare the components. The intuition behind this metric method is that if two components represent the same sinusoidal waveform, their periodogram will be spiked, with the spike located at the same frequency. Thus, comparing the location of the

maximal values of the periodogram can yield good results. If the two components represent the same wave length with frequency $\omega = \frac{2\pi j_0}{T}$, then their periodogram are concentrated and $\Pi_{Y^{(k)}}[j_0] = 1$, leading to spike$\left(Y^{(k)}, Y^{(l)}\right) = 1$. However, if the spectrum supports of the two waveforms $Y^{(k)}, Y^{(l)}$ are disjoint, then for any $j \in [\![0, T-1]\!]$,

$$\frac{\Pi_{Y^{(k)}}[j] + \Pi_{Y^{(l)}}[j]}{2} < .5 \ .$$

despite their spectrum being very concentrated and thus their similarity is low.

For automated grouping, Alexandrov & Golyandina (2005) propose the following strategy. First, they consider only consecutive components. This choice is motivated by the fact that analytically, it can be shown that an exponentially modulated harmonic signal will generate two eigentriples with eigenvector similar to exponentially modulated harmonic with the same frequency, a phase shift close to $\frac{\pi}{2}$ and their associated eigenvalues are very close. Moreover, as the eigenvectors themselves are harmonic, and as they contain less noise, they use the similarity measure spike on the eigenvectors U_k and U_l associated to the components $Y^{(k)}$ and $Y^{(l)}$. The adjacency matrix $\boldsymbol{A}$ is defined by

$$\boldsymbol{A}_{k,l} = \begin{cases} 1 & \text{if spike}\left(U_k, U_l\right) \geq \rho_0 \text{ and } |l-k| = 1 \\ 0 & \text{elsewhere} \end{cases}$$

with a threshold $\rho_0 > 0$ given as a parameter of the grouping strategy.

Support Similarity (AG2). Abalov & Gubarev (2014) introduce a more flexible comparison of periodogram of the reconstructed components. With the previous similarity measure, when the components are not waveform or if the frequency of the waveform is not on the regular grid with step $\frac{1}{W}$, spike might decrease a lot because the periodograms of the eigenvectors are not concentrated enough. The similarity measure is defined based on the set of non-negligible frequencies, defined for a component $Y^{(k)} \in \mathcal{X}_L^1$ as the ordered set

$$F_{Y^{(k)}} = \left\{ 0 \leq j_1 < \cdots < j_M \leq L/2 \ \middle| \ \forall 1 \leq m \leq M,\ \Pi_{Y^{(k)}}[j_m] \geq \rho_p \|\Pi_{Y^{(k)}}\|_\infty \right\} \tag{4.5}$$

for a threshold $\rho_p \in [0, 1]$ given as a parameter of the grouping strategy. The frequencies $\frac{2\pi j_m}{T}$ are the part of the spectrum with a high fraction of the spectrum energy. The threshold is also selected adaptively from the maximal value in the periodogram, similarly to the selection of the eigenvalue threshold for the component selection. The measure of similarity between two components $Y^{(k)}$ and $Y^{(l)}$ is the computed using

$$\text{d\_supp}\left(Y^{(k)}, Y^{(l)}\right) = \max_{0<h<M} \frac{|F_{Y^{(k)}}[h] - F_{Y^{(l)}}[h]|}{L/2}$$

where $F_{Y^{(k)}}[h]$ denotes the h-th element in the ordered set $F_{Y^{(k)}}$ and M is the minimum of the cardinal of the two sets $F_{Y^{(k)}}$ and $F_{Y^{(l)}}$. If the support of the components $Y^{(k)}$ both contain M spikes, then this measure evaluates the distance between those spikes and the similarity is taken relatively to the maximal value. It is more robust than the metric **HG** as if the components to group have two modes in their spectrum, then the **HG** can reject them because their spectrum is not concentrated enough whereas **AG2** will be able to detect the fact that these two modes are in the same location. However, this metric can be unstable is the spectrum is not composed of clear spikes. If the spikes

are not clear, the estimation of $F_{Y^{(k)}}$ is unstable and some extra values are included in it in presence of noise. In this case, the order of the spikes can be misaligned, resulting in an overestimation of the distance.

Building on this similarity, Abalov & Gubarev (2014) propose to group the components when their distance d_supp is lower than a threshold $\rho_2 > 0$ given as a parameter of the method. Note that unlike **HG**, the spectrum is not estimated on the eigenvectors U_k but directly on the components $Y^{(k)}$ computed with SSA. They use the singular value ratio, introduced with **AG1**, to avoid grouping components that have very different variance impact. They define the adjacency matrix $\boldsymbol{A}$ with

$$\boldsymbol{A}_{k,l} = \begin{cases} 1 & \text{if } \dfrac{\min(\lambda_k, \lambda_l)}{\max(\lambda_k, \lambda_l)} \geq \rho_1 \text{ and } \mathrm{d\_supp}\left(Y^{(k)}, Y^{(l)}\right) \leq \rho_2 \\ 0 & \text{elsewhere} \end{cases}$$

Harmonic Support Grouping (HSG). One of the drawbacks of **HG** is that it is designed exclusively to retrieve harmonic components. When SSA components have a wide spectral support, the energy of the spectrum is spread and no frequency concentrate enough energy to be used for similarity comparison. The comparison of the spectrum values makes it impossible to group wider support components, because of the normalization. In **AG2**, the full spectral supports are compared. With this method, it is possible to group components that are not spiked. But slight mistakes in the spectrum estimation can make this method unstable. If one spike is missed in one component and not in the other, the similarity is not correctly computed anymore.

To improve the grouping of wide spectrum components using the periodogram, we propose a more robust comparison based on the spectral support of the components. The similarity metric is defined by considering the fundamental frequency of the spectrum – taken as the center of the spectrum support – and the width of their support. For a component $Y^{(k)}$, its fundamental frequency $h_{Y^{(k)}}$ and its spectrum width $w_{Y^{(k)}}$ are defined as

$$h_{Y^{(k)}} = \frac{F_{Y^{(k)}}[0] + F_{Y^{(k)}}[M]}{2} \quad \text{and} \quad w_{Y^{(k)}} = \frac{F_{Y^{(k)}}[M] - F_{Y^{(k)}}[0]}{2}$$

with $F_{Y^{(k)}}$ defined as in (4.5) with parameter ρ_p. These two characteristics can be used to detect components which are not spiked, but with a support centered around a fundamental frequency. We consider that two components should be grouped together when their fundamental frequencies are close. The estimation of the fundamental frequency is more stable than the ordered spike used in **AG2**, as small mistakes in the estimation of the support bound do not create misalignment. But this estimation might be off for components which have multiple modes. Indeed, if the modes are far away, the fundamental frequency can match the one of another component which is concentrated around this frequency. To avoid this instability, we use the spectral width to select components with small spectrum width. The estimation of our two characteristics are computed using the pattern U_k associated to $Y^{(k)}$, in order to be more robust to small variations. Finally, the adjacency matrix is defined as

$$\boldsymbol{A}_{k,l} = \begin{cases} 1, & \text{if } \left|\dfrac{h_{U_k} + h_{U_l}}{W/2}\right| \leq \rho_f \text{ and } \dfrac{2w_{U_W}}{W} \leq \rho_s \text{ for } i = \{k, l\} \\ 0 & \text{elsewhere} \end{cases}$$

with parameters $\rho_f, \rho_s \in [0, 1]$. This similarity measure selects components with a spectrum which is not too wide, and groups them when their fundamental frequency are close. This similarity metric is useful when the components to groups are uni-modal but not spiked.

K-means (KM). Instead of defining the spectrum comparison properties and the grouping level, it is possible to use statistical methods to form the groups. In their paper, Álvarez-Meza et al. (2013) propose to use K-means algorithm to automatically create the group of component. The clustering algorithm is used with the ℓ_2-norm to form clusters based on the periodogram of the eigenvectors U_k. Using this method, the parameter and threshold used to compare the components are directly defined by the K-means. The method computes C clusters from the set

$$\left\{ \Pi_{U_k} \;\middle/\; k \in [\![1, W]\!], \quad \lambda_k > \tau_l \lambda_2 \right\}$$

The selection part is the same as the one defined in Subsection 4.5.1. We denote M the number of component selected in this set. The number of clusters C is a given parameter for the clustering method. It is critical for the performance of this grouping strategy. An estimation of C can be made using the rank of the matrix $\mathbf{\Pi}$

$$\mathbf{\Pi} = \begin{bmatrix} \Pi_{U_1} & \dots & \Pi_{U_M} \end{bmatrix}^\intercal .$$

by computing the singular values $\sigma_1 \geq \cdots \geq \sigma_M$ of Π and take $C = \operatorname{argmin}_k \sigma_k < \rho_r \sigma_1$, with $\rho_r \in [0, 1]$ given as the parameter of **KM** similarity. This similarity directly estimates the number of components in the decomposed signal, by estimating the number of different Fourier spectrum for the eigenvectors. The adjacency matrix $\boldsymbol{A}$ is defined using the clusters $\mathcal{C}_m$ formed with the K-means algorithm such that

$$\boldsymbol{A}_{k,l} = \begin{cases} 1 & \text{if } \exists\ m \in [\![1, C]\!]\ s.t.\ k, l \in \mathcal{C}_m \\ 0 & \text{elsewhere} \end{cases}$$

This matrix is symmetric by definition and connected. Indeed, all the component in $\mathcal{C}_m$ are considered as adjacent.

4.5.3 Group Creation

In this work, we analyze two methods to create groups based on the adjacency matrix $\boldsymbol{A}$.

Uniform Method (UM). A first strategy to create the groups from the adjacency matrix is to group all the components that are adjacent together, without considering that some components are more important than the other in the group. A component $Y^{(k)}$ is added in a group if it is adjacent to one component of the group

$$Y^{(k)} \in I_m \Leftrightarrow \max_{l \in I_m} A_{k,l} = 1 \Leftrightarrow \exists l \in I_m\ s.t.\ A_{k,l} = 1$$

This method is the one commonly used by the strategies that have been proposed in the literature (Abalov & Gubarev, 2014; Álvarez-Meza et al., 2013).

Hierarchical Method (HM). In this work, we propose a novel group creation method which takes into account the hierarchy between components. As stated earlier, the components are ranked based on their importance, in term of the explained variance. The eigenvalue λ_k associated to $Y^{(k)}$ is the part of the variance of x explained by this component. We propose to take into account the relative importance of the components. Our strategy add a component in the group when this component is adjacent to the dominant component in this group, in the sense of the explained variance. For each group I_m, the dominant component is $Y^{(k_m)}$ such that $k_m = \min I_m$ and for $l \in I_m$, $\lambda_l \leq \lambda_{k_m}$. The groups are then created such that

$$Y^{(k)} \in I_m \Leftrightarrow A_{k,k_m} = 1$$

If the component cannot be added into one of the existing group, a new group containing only this component is created. This strategy gives more importance to the dominant component in the group and therefore, produces groups with less noise and with a stronger coherence.

4.5.4 Evaluation

Signal Generation. The grouping strategies have been evaluated on randomly sampled artificial signals. The goal of the grouping is to identify the trend, the periodical components and the noise. The test signals are sampled at 100Hz following

$$X[t] = \underbrace{b_1 t^{\beta}}_{C^{(1)}} + \sum_{n=2}^{N} \underbrace{b_n e^{-\alpha_n t} \sin\left(2\pi f_n t + \phi_n\right)}_{C^{(n)}} + \epsilon_t \tag{4.6}$$

with ϵ_t a gaussian white noise with variance $\sigma = s\sigma_X$. The parameter are chosen uniformly such that $\beta \in \big[0, 5\big]$, $b_n \in \big[0, 1\big]$, $\phi_n \in \left[\frac{-\pi}{2}, \frac{\pi}{2}\right]$, $\alpha_n \in \left[0, \frac{1}{2}\right]$, $f_n \in \big[0, 50\big]$Hz and $s \in \big[0, 40\big]$dB. We define 3 classes of signals. For the first class, we fix $b_1 = \alpha_n = 0$, (harmonic + noise), then the second class is such that $\alpha_n = 0$, (trend + harmonic + noise) and the third class corresponds to the general model defined in (4.6), with all parameters sampled uniformly (trend + modulated harmonic + noise).

Evaluation Metrics. In previous works, the metric used to quantify the quality of the grouping is the determination coefficients r^2 (Abalov & Gubarev, 2014). This coefficient measures the variance of the error between 2 components y and $\bar{y}$ is computed with

$$r^2\left(Y, Y'\right) = 1 - \frac{\|Y - Y'\|^2}{\|Y - \mathbb{E}Y\|^2} \tag{4.7}$$

This correspond to the ratio between the estimation error and the signal variance. The recall and the precision of the grouping is then computed as

$$R = \frac{1}{N}\sum_{n=1}^{N} \min_{1\leq m\leq M} r^2\left(C^{(n)}, X^{(m)}\right) \qquad P = \frac{1}{M}\sum_{m=1}^{M} \min_{1\leq n\leq N} r^2\left(C^{(n)}, X^{(m)}\right) \tag{4.8}$$

where $\left\{X^{(m)}\right\}_{1\leq m\leq M}$ are the components obtained after the grouping and $\left\{C^{(n)}\right\}_{n=1}^{N}$ are the components defined in (4.6). To merge the two concepts, we propose to consider the score obtained for an optimal allocation between the ground truth components and

the components computed by the grouping procedure, *i.e.*

$$S = \frac{1}{N} \min_{\sigma \in \mathfrak{S}(M)} \sum_{n=1}^{N} r^2 \left(C^{(n)}, X^{\left(\sigma(n)\right)} \right) \tag{4.9}$$

with $\mathfrak{S}(M)$ denoting the permutation group of $\{1, \dots, m\}$. This metrics is more informative as it penalizes grouping strategies which do not aggregate correctly the components. If, several groups of components are close to the same original component $C^{(n)}$, the precision does not penalize the number of components. For the recall, the same group component can be used to approximate two different components. The metric S is more robust as it considers that one group component should be associated to one and only one original component.

The grouping quality is highly dependent of the quality of the initial component computed with the SSA. Thus, it is important to compare the score with the score which would have been obtained from the raw components from the SSA. To measure this relative improvement, the scores are computed for the evaluated components obtained with the grouping strategy as R, P, S and for the components obtained without grouping R_0, P_0 and S_0 . The relative metric is then computed as the rate of increase between the 2 scores, *i.e.*

$$R_r = \frac{R - R_0}{1 - R_0} \ , \quad P_r = \frac{P - P_0}{1 - P_0} \ \text{ and } \quad S_r = \frac{S - S_0}{1 - S_0} \ . \tag{4.10}$$

These metrics are less sensible to starting components, produced by bad SSA parameters.

Numerical Results. For all the simulation, parameter W has been fixed to half the length of the signal $T/2$. The parameters from existing algorithm where chosen based on the proposed value in the original work (Abalov & Gubarev, 2014; Álvarez-Meza et al., 2013), *i.e.* $\rho_0 = 0.8$, $\rho_1 = 0.8$, $\rho_2 = 0.05$, $\rho_c = 0.8$, $\rho_p = 0.8$ and $\rho_r = 0.4$. For the two proposed methods **HSG** and **wCG**, parameter have been fixed to $\rho_f = 0.001$ and $\rho_s = 0.6$ from a few empirical observation, outside our test samples.

Figure 4.4 illustrates the decomposition of a third class signal, composed by a trend, 4 exponentially modulated harmonic and a Gaussian white noise (SNR 9.7dB) using **HG** similarity metric and **HM** group formation strategy. This grouping retrieves four out of five components with very little distortion. The mean coefficient of determination r^2 is 0.96 for these four components, instead of 0.21 without grouping. As the amplitude of the fifth components is small compared to the other components, it is included in the noise term by the grouping procedure. This is expected as it is very hard to find components with low SNR.

The different procedures have been tested on 1000 signals from each of the three classes and Table 4.1 reports the mean of the resulting scores for these 3000 signals. For the relative recall R_r, method (**HG**)-(**HM**) is statically better than other methods, in the sense of Friedman (1937) test, with a mean increase rate of 48.4% . For other metrics P_r, S_r, methods using (**KM**) give statically better results than other procedures, with mean increase rates of 80.4% for P_r and 63.9% for S_r. This boost in precision for (**KM**) is explained by the good estimation of the number of components used in this method. This also explain the improvement in S_r .

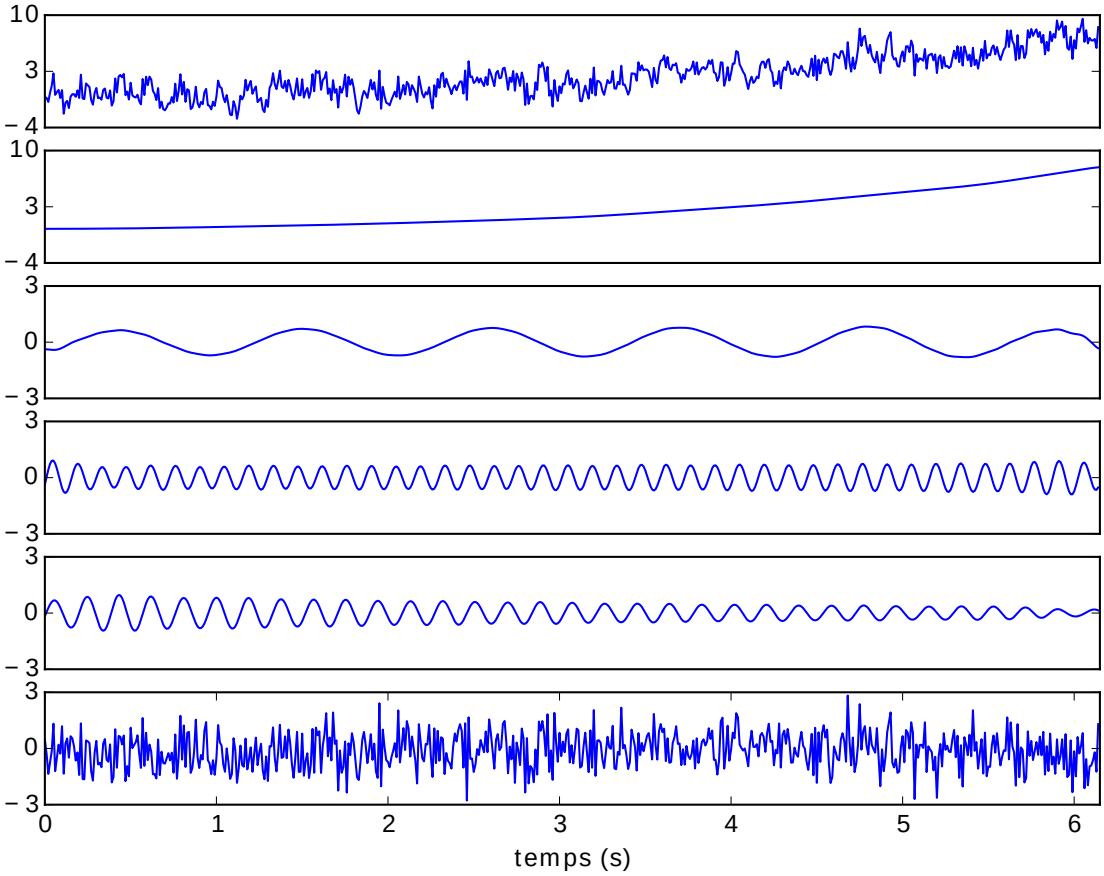


Figure 4.4: Signal decomposition using SSA with (**HG**)-(**HM**) grouping procedure. (*top*) Original signal. (*middle 4*) Four components resulting from the grouping procedure. (*bottom*) Residual signal, containing the fifth original component and the noise.

Méthodes	**AG1**	**AG2**	**wCG**	**HG**	**KM**	**HSG**
R_r - **UM**	0.426	0.33	0.436	0.402	0.395	0.475
R_r - **HM**	0.427	0.367	0.437	**0.484**	0.396	0.459
P_r - **UM**	0.338	0.329	0.351	0.693	0.804	0.409
P_r - **HM**	0.339	0.347	0.351	0.705	**0.804**	0.439
S_r - **UM**	0.361	0.373	0.373	0.567	0.639	0.422
S_r - **HM**	0.361	0.389	0.373	0.592	**0.639**	0.437

Table 4.1: Evaluation of the automatic grouping procedure on the signals from the test base. The results are presented for both grouping strategy (**UM**) and (**HM**).

For all similarity, the use of strategy (**HM**) introduced in this chapter improves the results both for precision and recall. This effect is small for correlation based measures and for the (**KM**) procedure. For the other metrics based on the periodogram, the improvement is statistically noticeable, except for the recall with (**HSG**) which is statistically equivalent for (**UM**) and (**HM**). The (**HM**) group formation strategy can be safely used with any methods as it provides more robust results, without degrading the performance compared to (**UM**).

For signal in the first class, composed only of harmonic components, the similarity (**HG**) and (**HSG**) give equivalent results for all three metrics and are statistically better than other methods, even (**KM**), for these metrics. These two metrics are well suited for the reconstruction of pure harmonic components, and thus for the class one signals. For the components with very concentrated spectrum, the two similarity give similar results, explaining this equivalence between them. For class 2 and 3, using (**KM**) gives statistically better results for P_r and S_r. For the recall, it is also statistically better than all the methods, except (**HSG**) used with (**HM**) which is equivalent.

This analysis is supported by the component-wise analysis of the results. The mean determination coefficient r^2 obtained for the $C^{(1)}$ trend component in class 2 and 3 is greater in probability with (**KM**) than with other methods. This method uses the whole frequency spectrum to compare the components. This is advantageous to retrieve the trend component which can have a wide periodogram. For the harmonic components $(C^{(n)})_{n>1}$ in class 1 and 2, not modulated, using similarity measure (**HSG**) or (**HG**) with the group formation (**HM**) gives a statistically better r^2 score. The metrics based on the periodogram properties are particularly adapted for these components as their spectrum displays clear spikes. For exponentially modulated components, all the methods are statistically equivalent. These components are harder to estimate than the harmonic components and all the described methods fail to retrieve them precisely.

The general framework for automatic grouping strategy introduced in this chapter shows that the choice of the grouping strategy is dependent on the relative importance given to the extraction of each kind of components. The (**KM**) method seems to be the most robust to estimate different kind of components. It also gives a precise estimation of the number of components, which could be use in combination with other grouping methods. Finally, the hierarchical group formation strategy (**HM**) proposed in this chapter slightly improves the performances of the automatic grouping methods.

4.6 Conclusion and Perspectives

In this chapter, we show that the Singular Spectrum Analysis solves a non-convex optimization problem, which is related to the one for convolutional representations, up to border effects. The decomposition computed with SSA is thus a good starting point for convolutional sparse coding with regularization parameter λ close to 0. We also introduce in Section 4.5 a general framework for automatic grouping of the components extracted with SSA. This step – necessary to ensure the interpretability of the decomposition – is usually done manually. New similarity measures are introduced, along with the description of the literature's methods. We also propose a novel group formation strategy, which takes the importance of the features into account to improve the grouping quality. The different grouping strategies are then compared on artificial signals, highlighting their strengths and weaknesses.

With the SSA, we have ssen that the application of a dimension reduction technique on the sub-series, in addition to the duality between trajectory matrices and signals, captures the local variations efficiently. The same idea of using a global technique on all sub-signals has been used for various methods. The STFT is one example, with a fixed analysis. Another example is the classical dictionary learning applied for all patches in an image. In this case, the original signal is also reconstructed by stitching the patches together. The study of extensions of the SSA to other matrix factorization techniques could allow computing convolutional representation of the signal with differ-

ent constraints on the dictionary. For instance, using Independent Component Analysis (ICA) could retrieve components generated with independent patterns for blind source separation problems. Another interesting idea is to use a pattern selection technique to retrieve the patterns which permit to best encode the other one as good starting point for convolutional dictionary learning.

5

Distributed Convolutional Sparse Coding

Contents

This chapter proposes a novel algorithm to solve the convolutional sparse coding problem. This algorithm was designed to run in a distributed setting, with local message passing, making it communication efficient. It relies on locally greedy updates which are shown to accelerate the resolution, compared to the greedy coordinate descent. We prove the convergence of this algorithm and highlight its computational speed-up, which is super-linear for the distributed algorithm, compared to the classical greedy coordinate descent, but sub-linear compared to our new locally greedy version. These properties are backed with numerical experiments.

5.1 Convolutional Representation for Long Signals

Sparse coding aims at building sparse linear representations of a data set based on a family of basic elements called atoms. It has proven to be useful in many applications, ranging from EEG analysis to images and audio processing (Adler et al., 2013; Kavukcuoglu et al., 2010; Mairal et al., 2010; Grosse et al., 2007). Convolutional sparse coding is a specialization of this approach, focused on building sparse, shift-invariant representations of signals. Such representations present a major interest for applications like segmentation or classification as they separate the shape and the localization of patterns in a signal. Convolutional sparse coding can also be used to estimate the similarity of two signals which share similar local patterns and to find the correspondences between different temporal events. Depending on the context, the dictionary can either be fixed analytically (*e.g.* wavelets, see Mallat 2008), or learned from the data (Bristow et al., 2013; Mairal et al., 2010).

Several algorithms have been proposed to solve the convolutional sparse coding problem. In Kavukcuoglu et al. (2010), the authors extend to convolutional sparse coding the coordinate descent (CD) methods introduced by Friedman et al. (2007). This method greedily optimizes one coordinate at each iteration using fast local updates. The Fast Iterative Soft-Thresholding Algorithm (FISTA) was adapted for convolutional problems in Chalasani et al. (2013) and uses proximal gradient descent to compute the representation. The Feature Sign Search (FSS), introduced in Grosse et al. (2007), solves at each step a quadratic sub-problem for an active set of the estimated nonzero coefficients and the Fast Convolutional Sparse Coding (FCSC) of Bristow et al. (2013) is based on Alternating Direction Method of Multipliers (ADMM). We refer the reader to Section 3.3 for a more detailed presentation of these algorithms.

To our knowledge, there is no scalable version of these algorithms for long signals. This is a typical situation, for instance, in physiological signal processing where sensor information can be collected for a few hours with sampling frequencies ranging from 100 to 1000Hz. For ℓ_1-regularized optimization, some existing algorithms are already optimal in terms of the number of iterations they use. To accelerate them on large scale problems, it is the computational complexity of their iterations that should be considered. A first line of work to improve the complexity of these algorithm considers the estimation of the non-zero coefficients of the optimal solution to reduce the dimension of the optimization space, using screening (El Ghaoui et al., 2012; Fercoq et al., 2015) or active-set algorithms (Johnson & Guestrin, 2015). Another line of work focuses on developing parallel algorithms which compute multiple updates simultaneously. Recent studies have considered distributing coordinate descent algorithms for general ℓ_1-regularized minimization (Scherrer et al., 2012a,b; Bradley et al., 2011; Yu et al., 2012). These papers derive general purpose synchronous algorithms using either locks or synchronizing steps to ensure convergence in general cases. In You et al. (2016), the authors derive an asynchronous distributed algorithm for the projected coordinate descent. However, this work relies on centralized communication and finely tuned step size to ensure the convergence of the method. In this chapter, we develop a distributed algorithm to accelerate the convolutional sparse coding. As this research direction is orthogonal to the support estimation, it is possible to use them jointly with our algorithm. The evaluation of the performances of our algorithm with active-set or screening strategies is left for future work.

By exploiting the structure of the convolutional problem, we design a novel distributed algorithm based on coordinate descent, named Distributed Convolution Coordinate Descent (DICOD). DICOD is asynchronous and each process can run independently without locks or synchronization steps. This algorithm uses a local communication scheme to reduce the number of inter-process messages needed and does not rely on external learning rates. We prove in this chapter that this algorithm scales super-linearly with the number of cores compared to the sequential CD, up to certain limitations.

In Section 5.3, we introduce the DICOD algorithm for the resolution of convolutional sparse coding. Then, we prove in Section 5.4 that DICOD converges to the optimal solution for a wide range of settings and we analyze its complexity. Finally, Section 5.5 presents numerical experiments that illustrate the benefits of the DICOD algorithm with respect to other state-of-the-art algorithms and validate our theoretical analysis.

5.2 Convolutional Coordinate Descent

In this section, we recall the convolutional sparse coding problem and the greedy coordinate descent algorithm to solve it. For more details, we refer the readers to Chapter 3.

5.2.1 Convolutional Sparse Coding

Consider the multivariate signal $X \in \mathcal{X}_T^P$. Let $\boldsymbol{D} = \{D_k\}_{k=1}^K \subset \mathcal{X}_W^P$ be a set of K patterns with $W \ll T$ and $Z \in \mathcal{X}_L^K$ be K activation signals with $L{=}T{-}W{+}1$. The convolutional sparse representation models a multivariate signal X as the sum of K convolutions between a local pattern D_k and an activation signal Z_k such that:

$$X[t] = \sum_{k=1}^{K} (Z_k * D_k)[t] + \mathcal{E}[t], \qquad \forall t \in [\![0, T-1]\!] \ . \tag{5.1}$$

with $\mathcal{E} \in \mathcal{X}_T^P$ representing an additive noise term. This model also assumes that the coding signals Z_k are sparse, in the sense that only a few entries are nonzero in each signal. The sparsity property forces the representation to display localized patterns in the signal. Note that this model can be extended to higher order signals such as images by using the proper convolution operator. In this study, we focus on 1D-convolution for the sake of simplicity.

Given a dictionary of patterns $\boldsymbol{D}$, convolutional sparse coding aims to retrieve the sparse decomposition Z^* associated to the signal X by solving an ℓ_1-regularized optimization problem:

$$Z^* = \operatorname*{argmin}_Z E(Z) \triangleq \frac{1}{2} \left\| X - \sum_{k=1}^{K} Z_k * D_k \right\|_2^2 + \lambda \left\| Z \right\|_1 , \tag{5.2}$$

for a given regularization parameter $\lambda > 0$. (5.2) can be interpreted as a special case of the LASSO problem with a band circulant matrix. Therefore, classical optimization techniques designed for LASSO can be applied to solve it with the same convergence guarantees.

5.2.2 Convolutional Coordinate Descent.

The coordinate descent is a method which updates one coordinate at each iteration. This type of optimization algorithms is efficient for sparse optimization problem as few coefficients need to be updated to find the optimal solution and the greedy selection of updated coordinates is a good strategy to get quick convergence to the optimal point. The localized updates make it natural to consider the parallelization of such algorithm.

The method proposed by Kavukcuoglu et al. (2010) iteratively updates at each iteration one coordinate (k_0, t_0) of the sparse code. Algorithm 3.7 gives the details of the algorithm. When the coefficient (k, t) of $Z^{(q)}$ is updated to a value $u \in \mathbb{R}$ for $Z^{(q+1)}$, a simple function of u gives the reduction of the cost obtained with this update and we denote its maxima

$$\Delta E_k[t] = \max_{u \in \mathbb{R}} e_{k,t}(u) = E(Z^{(q)}) - E(Z^{(q+1)}) \ . \tag{5.3}$$

The greedy coordinate descent updates the chosen coordinate (k_0, t_0) to the value $Z'_{k_0}[t_0] = \text{argmax}_{u \in \mathbb{R}} e_{k_0,t_0}(u)$, maximizing the cost reduction of the update. The coordinate is chosen as the one with the largest difference $\max_{(k,t)} \left|\Delta Z_k[t]\right|$ between its current value $Z_k[t]$ and the value $Z'_k[t]$ with

$$\Delta Z_k[t] = Z_k[t] - Z'_k[t] \tag{5.4}$$

The updates are run until the $\max_{k,t} |\Delta Z_k[t]|$ become smaller than a specified tolerance parameter ϵ We studied this update scheme as it aims to get the largest gain from each updates. Other coordinate update strategies were proposed such as cyclic updates (Friedman et al., 2007) or random updates (Shalev-Shwartz & Tewari, 2009) and our algorithm can easily be implemented with such update schemes. In this study, we focus on the greedy approach as it aims to get the largest gain from each update. Moreover, as the updates in the greedy scheme are more complex to compute, distributing them provides a larger speedup compare to other strategies. We refer the reader to the work by Nutini et al. (2015) which discussed extensively the difference between these schemes.

A closed form solution exists to compute the optimal value $Z'_{k_0}[t_0]$ of e_{k_0,t_0}

$$Z'_{k_0}[t_0] = \frac{1}{\|D_{k_0}\|_2^2} \text{Sh}(\beta_{k_0}[t_0], \lambda) = \underset{u \in \mathbb{R}}{\text{argmax}}\, e_{k_0,t_0}(u), \tag{5.5}$$

with $\beta_k[t] = \left(\widetilde{D}_k * \left(X - \sum_{\substack{k'=1 \\ k' \neq k}}^{K} Z_{k'} * D_{k'} - \Phi_t\left(Z_k\right) * D_k \right) \right)[t]$ and Sh the soft thresholding operator $\text{Sh}(u,\lambda)=\text{sign}(u)\max(|u|-\lambda,0)$. The success of this algorithm highly depends on the efficiency in computing the coordinate update. For problem (5.2), Kavukcuoglu et al. (2010) show that if at iteration q, the coefficient (k_0, t_0) of $Z^{(q)}$ is updated to the value $Z'_{k_0}[t_0]$, then it is possible to compute $\beta^{(q+1)}$ from $\beta^{(q)}$ using

$$\beta_k^{(q+1)}[t] = \beta_k^{(q)}[t] - \mathcal{S}_{k,k_0}[t - t_0]\Delta Z_{k_0}^{(i)}[t_0], \qquad \forall (k,t) \neq (k_0, t_0) \tag{5.6}$$

with $\mathcal{S}_{k,l}[t] = (\widetilde{D_k} * D_l)[t]$. For all $t \notin [\![-W+1, W-1]\!]$, $\mathcal{S}[t]$ is zero. Thus, only $\mathcal{O}(KW)$ operations are needed to maintain β up to date with the current estimate Z. Finally, the complexity of an iteration of CD is dominated by the $\mathcal{O}(KT)$ operations needed to find the maximum of $|\Delta Z_k[t]|$.

Algorithm 5.1 DICOD$_M$

Input: $\boldsymbol{D}, X$, parameter $\epsilon > 0$
In parallel for $m = 1 \cdots M$
For all (k, t) in $\mathcal{C}_m$, initialize $\beta_k[t]$ and $Z_k[t]$
repeat
  Receive messages and update β with (5.6)
  $\forall (k, t) \in \mathcal{C}_m$, compute $Z'_k[t]$ with (5.5)
  Choose $(k_0, t_0) = \underset{(k,t)\in\mathcal{C}_m}{\operatorname{argmax}} |\Delta Z_k[t]|$
  Update β with (5.6) and $Z_{k_0}[t_0] \leftarrow Z'_{k_0}[t_0]$
  if $t_0 - mL_M < W$ **then**
    Send $(k_0, t_0, \Delta Z_{k_0}[t_0])$ to core $m - 1$
  if $(m + 1)L_M - t_0 < W$ **then**
    Send $(k_0, t_0, \Delta Z_{k_0}[t_0])$ to core $m + 1$
until for all cores, $|\Delta Z_{k_0}[t_0]| < \epsilon$

5.3 Distributed Convolutional Coordinate Descent

This section introduces an asynchronous algorithm called DICOD, which exploits the local independence of the coordinate descent updates to derive a distributed algorithm solving (5.2).

5.3.1 Algorithm

Algorithm 5.1 describes the steps of DICOD with M workers. Each worker $m \in [\![1, M]\!]$ is in charge of updating the coefficients of a segment $\mathcal{C}_m$ of length $L_M = L/M$ defined by:

$$\mathcal{C}_m = \Big\{ (k, t) \ ; \ k \in [\![1, K]\!], \ t \in [\![(m-1)L_M, mL_M - 1]\!] \Big\} \ .$$

The local updates are performed in parallel for all the cores using the greedy coordinate descent introduced in Subsection 5.2.2. When a core m updates the coordinate (k_0, t_0) such that $t_0 \in [\![(m-1)L_M + W, mL_M - W]\!]$, the coefficients of β that are updated are all contained in $\mathcal{C}_m$ and there is no need to update β on all the other cores. In these cases, the update is equivalent to a sequential update. When $t_0 \in [\![mL_M - W, mL_M]\!]$ $\Big($resp. $t_0 \in [\![(m-1)L_M, (m-1)L_M + W]\!]\Big)$, some of the coefficients of β in core $m + 1$ (resp. $m - 1$) need to be updated and the update is not local anymore. This can be done by sending the position of updated coordinate (k_0, t_0), and the value of the update $\Delta Z_{k_0}[t_0]$ to the neighboring core. Figure 5.1 illustrates this communication process. Inter-processes communications are very limited in DICOD. One node only communicates with its neighbors when it updates coefficients close to the extremity of its segment. When the size of the segment is reasonably large compared to the size of the patterns, only a small part of the iterations needs to send messages. We cannot apply the stopping criterion of CD in each worker of DICOD, as this criterion might not be reached globally. The updates in the neighbor cores can break this criterion. To avoid this issue, the convergence is considered to be reached once all the cores achieve this criterion simultaneously. Workers that reach this state locally are paused, waiting for incoming communication or for the global convergence to be reached.

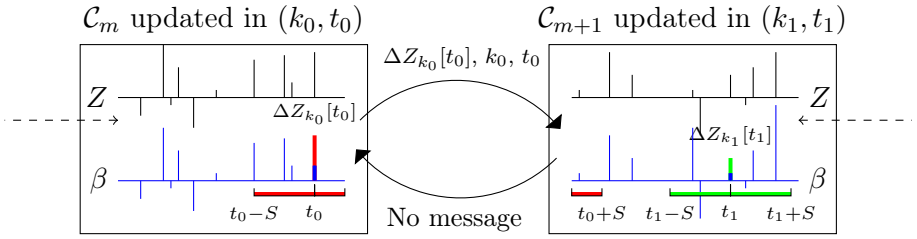


Figure 5.1: Core $m+1$ updates $(k_1, t_1) \in \mathcal{C}_{m+1}$ independently from the other cores as it is located out of the interference zones. Core m is updating a coordinate $(k_0, t_0) \in \mathcal{C}_m$ which is in the interference zone $[\![mL_M - S, mL_M]\!]$. Therefore, it needs to notify the process $m+1$ of the update by sending a message composed of the value of the update $\Delta Z_{k_0}[t_0]$ and its location (k_0, t_0). When core $m+1$ retrieves the pending message from m, it will update β to take into account the update of $Z_{k_0}[t_0]$.

The key point that allows distributing the convolutional coordinate descent algorithm is that the solutions on time segments that are not overlapping are only weakly dependent. Equation (5.6) shows that a local change has impact on a segment of length $2W - 1$ centered around the updated coordinate. Thus, if two far enough coordinates are updated simultaneously, the resulting point Z is the same as if these two coordinates had been updated sequentially. By splitting the signal into continuous segments over multiple cores, coordinates can be updated independently on each core up to certain limits.

5.3.2 Interferences

When two coefficients (k_0, t_0) and (k_1, t_1) are updated by two neighboring cores before receiving the communications of the other update, the updates might not be independent and cannot be considered sequential. The local version of β used for the second update does not account for the first update. We say that the updates are *interfering*. The cost reduction resulting from these two updates is denoted $\Delta E_{k_0,k_1}[t_0, t_1]$ and simple computations, detailed in Proposition 5.7.2, show that

$$\Delta E_{k_0,k_1}[t_0, t_1] = \overbrace{\Delta E_{k_0}[t_0] + \Delta E_{k_1}[t_1]}^{\text{iterative steps}} - \overbrace{\mathcal{S}_{k_0,k_1}[t_1 - t_0] \Delta Z_{k_0}[t_0] \Delta Z_{k_1}[t_1]}^{\text{interference}}, \tag{5.7}$$

If $|t_1 - t_0| \geq W$, then $\mathcal{S}_{k_0,k_1}[t_1 - t_0] = 0$ and the updates can be considered as sequential as the interference term is zero. When $|t_1 - t_0| < W$, the interference term does not vanish but Section 5.4 shows that under mild assumption, this term is controlled and does not break the convergence of DICOD.

5.3.3 Randomized Locally Greedy Coordinate Descent (SeqDICOD)

The theoretical analysis in Theorem 5.3 shows that DICOD provides a super-linear acceleration compared to the greedy coordinate descent. This result is backed with the numerical experiment presented in Figure 5.6. The super-linear speed up results from a double acceleration, provided by the parallelization of the updates – we update M coefficients at each iteration – and also by the reduction of the iteration complexity. Indeed, each core computes greedy updates with linear in complexity on $1/M$-th of the signal. Because the updates are only weakly dependent, choosing the coordinate to

Algorithm 5.2 Locally greedy coordinate descent SeqDICOD$_M$

Input: $\boldsymbol{D}, X$, parameter $\epsilon > 0$, number of segments M
Initialize $\beta_k[t]$ and $Z_k[t]$ for all (k, t) in $\mathcal{C}$
Initialize $dZ_m = +\infty$ for $m \in [\![1, M]\!]$
repeat
  Randomly select $m \in [\![1, M]\!]$
  $\forall (k, t) \in \mathcal{C}_m$, compute $Z'_k[t]$ with (5.5)
  Choose $(k_0, t_0) = \underset{(k,t)\in\mathcal{C}_m}{\operatorname{argmax}} |\Delta Z_k[t]|$
  Update β with (5.6)
  Update the current point estimate $Z_{k_0}[t_0]^{(q+1)} \leftarrow Z'_{k_0}[t_0]$
  Update max updates vector $dZ_m = \left| Z_{k_0}[t_0]^{(q+1)} - Z'_{k_0}[t_0] \right|$
until for all cores, $\|dZ\|_\infty < \epsilon$ and $\|\Delta Z_k[t]\|_\infty < \epsilon$

update using the locally greedy scheme instead of the fully greedy one does not fundamentally change the convergence rate. This super-linear speed-up means that running DICOD sequentially will still provide a speed-up compared to the greedy coordinate descent algorithm.

Algorithm 5.2 presents SeqDICOD. This algorithm is a sequential version of DICOD. At each step, one segment $\mathcal{C}_m$ is selected uniformly at random between the M segments. The greedy coordinate descent algorithm is applied locally on this segment. This update is only locally greedy and maximizes

$$(k_0, t_0) = \underset{(k,t)\in\mathcal{C}_m}{\operatorname{argmax}} |\Delta Z_k[t]|$$

This coordinate is then updated to its optimal value $Z'_{k_0}[t_0]$. In this case, there is no interference as the segments are not updated simultaneously.

Note that if $M = T$, this algorithm become very close to the randomized coordinate descent. The coordinate is selected greedily only between the K different channels of the signal Z at the selected time. So the selection of M depends on a tradeoff between the randomized coordinate descent and the greedy coordinate descent.

Also, the stopping criterion has to be modified to match the one from the coordinate descent. Indeed, it is necessary to wait until all the segments have their next update amplitude below the threshold to declare that SeqDICOD has converged. We define the vector $dZ \in \mathbb{R}^M$ with dZ_m the magnitude of the previous update on segment $m \in [\![1, M]\!]$. When $\|dZ_m\|_\infty = \max_m \left|dZ_m\right|$ decrease below ϵ, there is a chance that the algorithm converged and we can check the second condition. This method avoid the costly check of the second condition at each iteration, with a computational cost of $\mathcal{O}\left(KT\right)$, using the first condition which only have complexity $\mathcal{O}\left(M\right)$. The first condition alone does not guarantee convergence as an update in one segment can increase a coefficient on another segment. The second condition ensures that the convergence is reached, on all segments.

5.3.4 Existing Distributed Coordinate Descent Algorithms

This algorithm differs from the existing paradigm to distribute CD as it does not rely on centralized communication. Indeed, other distributed coordinate descent algorithms rely on a parameter server (PS), which is an extra worker that holds the current value

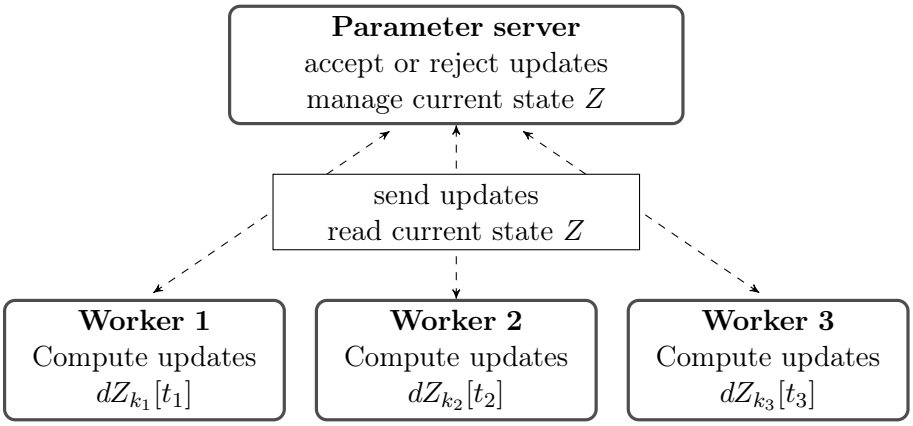


Figure 5.2: Parameter Server with centralized connection. The workers compute independently updates, by fetching the necessary information in the parameter server and then sending back the update to perform. The parameter server can accept or refuse the update, depending on its validity.

of Z. Figure 5.2 illustrates this concept. The PS can be in charge of avoiding the interferences. If an update sent by a worker m is in the interfering zone of a "recent" update by worker $m + 1$, which has not been transmitted to worker m, the PS can reject the update. But this parallelization scheme relies on centralized communication. All workers need to communicate with the PS to send the newly computed updates and to retrieve the current state. This is difficult to scale, when the number of workers grows and the communication cost goes up. The PS can be implemented using multiple processes to increase the communication efficiency but it means more resources are allocated to handling the updates and it makes the system more complex. The natural workload split proposed with DICOD allows for more efficient interactions between the workers and reduces the need for inter-node communications as only updates are sent, avoiding the communication of the current state. Also, communications are reduced to direct messages passing between neighboring workers and the necessary bandwidth is reduced.

Moreover, other distributed settings are designed without taking into account the specificity of the convolutional operator. Indeed, in most studies, the algorithm need to account for functions for which changing a coordinate value impacts the update value that would be used for all the other coordinates. To prevent this inter-dependence from breaking the convergence of the algorithm, methods designed by Bradley et al. (2011), Scherrer et al. (2012a) and Yu et al. (2012) use synchronous updates. Each process computes the update of one coordinate independently and then, all these updates are applied simultaneously to the current solution estimate, either by using a parameter server or by broadcasting them. Using this technique, each worker always has an up to date version of the solution estimate and there can be no interference. Scherrer et al. (2012b) proposed to improve the performances of these methods by clustering the coordinates into groups that are highly correlated. Then, the updates are computed such that at most one coordinate in each group is updated, reducing the rejection rate of the updates by the PS. The necessary synchronization steps of these four algorithms reduce the speed of the updates as any delay in one node computations impacts all the other updates. Also, the synchronization mechanism can be hard to scale to multiple machines over a network. The synchronization step is necessary for vectorial LASSO as

the change of one coordinate influences all the others and thus, there can be multiple updates interfering together. But in the convolutional setting, the weak dependency between time segments can be used to design a fully asynchronous algorithm as the interference can only happen between two neighbors.

Another approach, taken by Liu et al. (2015) for the randomized coordinate selection, is to control the size of the steps taken by the coordinate descent to avoid interferences. In their work, they do not use updates that replace the coordinate by its optimal value for the single coordinate problem but use a coordinate-wise gradient step described by (3.27). This update size is controlled by a learning rate parameter α and Liu et al. (2015) show that setting this parameter low enough guarantees the convergence of parallel and asynchronous coordinate descent steps to the solution of the LASSO. The actual upper bound on the learning rate to show convergence of this asynchronous algorithm depends on the communication delay between the nodes. You et al. (2016) show similar results with greedy coordinate selection. Once again, the smaller step size is designed to control interference between more than two workers. This cannot happen in the convolutional setting when the workers handle disjoint time segments. Thus, it is possible to use greedy updates that accelerates the algorithm convergence.

5.4 Properties of DICOD

5.4.1 Convergence of DICOD.

The interference magnitude is related to the value of the cross-correlation between dictionary elements, as shown in Proposition 5.1. Thus, when the interferences have low probability and small magnitude, the distributed algorithm behaves as if the updates were applied sequentially, resulting in a large acceleration compared to the sequential CD algorithm.

Proposition 5.1. *For concurrent updates for coefficients (k_0, t_0) and (k_1, t_1) of a sparse code Z, the cost update $\Delta E_{k_0 k_1}[t_0, t_1]$ is lower bounded by*

$$\Delta E_{k_0 k_1}[t_0, t_1] \geq \Delta E_{k_0}[t_0] + \Delta E_{k_1}[t_1] - 2\frac{\mathcal{S}_{k_0,k_1}[t_0 - t_1]}{\|D_{k_0}\|_2 \|D_{k_1}\|_2}\sqrt{\Delta E_{k_0}[t_0]\Delta E_{k_1}[t_1]}. \quad (5.8)$$

The proof of this proposition is given in supplementary materials. It relies on the $\|D_k\|_2^2$-strong convexity of (5.5), which gives $|\Delta Z_k[t]| \leq \frac{\sqrt{2\Delta E_k[t](Z)}}{\|D_k\|_2}$ for all Z. Using this inequality with (5.7) yields the expected result.

This proposition controls the interference magnitude using the cost reduction associated to a single update. When the correlations between the different elements of the dictionary are small enough, the interfering update does not increase the cost function. The updates are less efficient but do not degrade the current estimate. Using this control on the interferences, we can prove the convergence of DICOD.

Theorem 5.2. *If the following hypotheses hold*

H1. *For all (k_0,t_0),(k_1,t_1) such that $t_0 \neq t_1$,* $\left|\frac{\mathcal{S}_{k_0,k_1}[t_0-t_1]}{\|D_{k_0}\|_2\|D_{k_1}\|_2}\right| < 1$.

H2. *There exists $A \in \mathbb{N}^*$ such that all cores $m \in [\![1, M]\!]$ are updated at least once between iteration i and $i + A$ if the solution is not locally optimal.*

H3. *The delay in communication between the processes is inferior to the update time.*

Then, the DICOD algorithm converges to the optimal solution Z^* of (5.2)

Assumption (**H1**) is satisfied as long as the dictionary elements are not replicated in shifted positions in the dictionary. It ensures that, at each step, the cost is updated in the right direction. This assumption can be linked to the shifted mutual coherence introduced in Papyan et al. (2016).

Hypothesis (**H2**) ensures that all coefficients are updated regularly if they are not already optimal. This analysis is not valid when one of the cores fails. As only one core is responsible for the update of a local segment, if a worker fails, this segment cannot be updated anymore and thus the algorithm will not converge to the optimal solution.

Finally, under (**H3**), an interference only results from one update on each core. Multiple interferences occur when a core updates multiple coefficients in the border of its segment before receiving the communication from other processes border updates. When $T \gg W$, the probability of multiple interference is low and this hypothesis can be relaxed if the updates are not concentrated on the borders.

Proof sketch for Theorem 5.2. The full proof can be found in Subsection 5.7.3. The key point in proving the convergence is to show that most of the updates can be considered sequentially and that the remaining updates do not increase the cost of the current point. By (**H3**), for a given iteration, a core can interfere with at most one other core. Thus, without loss of generality, we can consider that at each step q, the variation of the cost E is either $\Delta E_{k_0}[t_0](Z^{(q)})$ or $\Delta E_{k_0k_1}[t_0, t_1](Z^{(q)})$, for some $(k_0, t_0), (k_1, t_1) \in [\![1, K]\!] \times [\![0, T-1]\!]$. Proposition 5.1 and (**H1**) proves that $\Delta E_{k_0k_1}[t_0, t_1](Z^{(q)}) \geq 0$. For a single update $\Delta E_{k_0}[t_0](Z^{(q)})$, the update is equivalent to a sequential update in CD, with the coordinate chosen randomly between the best in each segments. Thus, $\Delta E_{k_0}[t_0](Z^{(q)}) > 0$ and the convergence is eventually proved using results from Osher & Li (2009). □

5.4.2 Speedup of DICOD.

We denote $S_{cd}(M)$ the speedup of DICOD compared to the sequential greedy CD. This quantify the number of iteration that can be run by DICOD during one iteration of CD.

Theorem 5.3. *Let $\alpha = \frac{W}{T}$ and $M \in \mathbb{N}^*$. If $\alpha M < \frac{1}{4}$ and if the non-zero coefficients of the sparse code are distributed uniformly in time, the expected speedup $\mathbb{E}[S_{cd}(M)]$ is lower bounded by*

$$\mathbb{E}[S_{cd}(M)] \geq M^2(1 - 2\alpha^2M^2\left(1 + 2\alpha^2M^2\right)^{\frac{M}{2}-1}) \ .$$

This result can be simplified when the interference probability $(\alpha M)^2$ is small.

Corollary 5.4. *The expected speedup $\mathbb{E}[S_{cd}(M)]$ when $(M\alpha)^2 \to 0$ is such that*

$$\mathbb{E}[S_{cd}(M)] \underset{\alpha \to 0}{\gtrsim} M^2(1 - 2\alpha^2M^2 + \mathcal{O}(\alpha^4M^4)) \ .$$

Proof sketch for Theorem 5.3. The full proof can be found in Subsection 5.7.4. There are two aspects involved in DICOD speedup: the computational complexity and the acceleration due to the parallel updates. As stated in Subsection 5.2.2, the

complexity of each iteration for CD is linear with the length of the input signal T. In DICOD, each core runs on a segment of size $\frac{T}{M}$. This accelerates the execution of individual updates by a factor M. Moreover, all the cores compute their update simultaneously. The updates without interference are equivalent to sequential updates. Interfering updates happen with probability $\left(M\alpha\right)^2$ and do not degrade the cost. Thus, one iteration of DICOD with N_i interferences provides a cost variation equivalent to $M-2N_i$ iterations using sequential CD and, in expectation, it is equivalent to $M-2\mathbb{E}[N_i]$ iterations of DICOD. The probability of interference depends on the ratio between the length of the segments used for each core and the size of the dictionary. If all the updates are spread uniformly on each segment, the probability of interference between 2 neighboring cores is $\left(\frac{MW}{T}\right)^2$. $\mathbb{E}[N_i]$ can be upper bounded using this probability and this yields the desired result. □

The overall speedup of DICOD is super-linear compared to sequential greedy CD for the regime where $(\alpha M)^2 \ll 1$. It is almost quadratic for small M but as M grows, there is a sharp transition that significantly deteriorates the acceleration provided by DICOD. Section 5.5 empirically highlights this behavior. For a given α, it is possible to approximate the optimal number of cores M to solve convolutional sparse coding problems.

Note that this super-linear speed up is due to the fact that CD is inefficient for long signals, as its iterations are computationally too expensive to be competitive with the other methods. The fact that we have a super-linear speed-up means that running DICOD sequentially will provide an acceleration compared to CD. This is what we did with SeqDICOD. For SeqDICOD, we have a linear speed-up in comparison to CD, when M is small enough. Indeed, the iteration cost is divided by M as we only need to find the maximal update on a local segment of size $\frac{T}{M}$. When increasing M over $\frac{T}{W}$, the iteration cost does not decrease anymore as updating β costs $\mathcal{O}\left(KW\right)$ and finding the best coordinate has the same complexity.

Using the same arguments, we can show that the expected speed-up of DICOD_M compared to SeqDICOD_M, denoted $\mathcal{S}_{dicod}$ is sub-linear.

Theorem 5.5. *Let $\alpha = \frac{W}{T}$ and $M \in \mathbb{N}^*$. If $\alpha M < \frac{1}{4}$ and if the non-zero coefficients of the sparse code are distributed uniformly in time, the expected speedup $\mathbb{E}[S_{SeqDICOD}(M)]$ of DICOD_M compared to SeqDICOD_M is lower bounded by*

$$\mathbb{E}[S_{dicod}(M)] \geq M(1 - 2\alpha^2M^2\left(1 + 2\alpha^2M^2\right)^{\frac{M}{2}-1}) \ .$$

This theorem has the same proof as Theorem 5.3, but the iteration cost is not decreased by M. As for $\mathbb{E}[\mathcal{S}_{cd}]$, it is almost linear for small M but as M grows, there is a sharp transition that significantly deteriorates the acceleration provided by DICOD, when the interferences begin to be too frequent.

5.5 Numerical Results

All the numerical experiments are run on five Linux machines with 16 to 24 Intel Xeon 2.70 GHz processors and at least 64 GB of RAM on local network. We use a combination of Python, C++ and the OpenMPI 1.6 for the algorithm implementation. The code

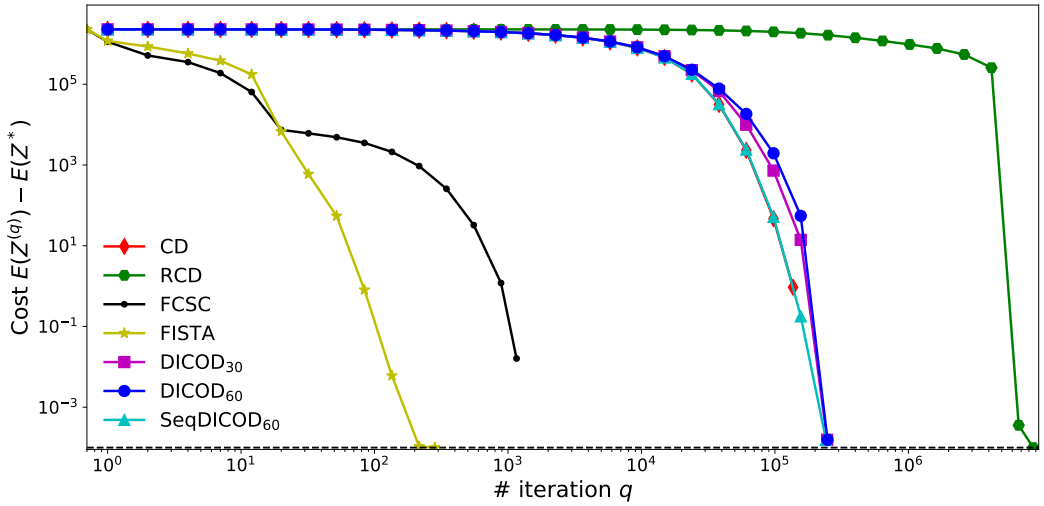


Figure 5.3: Evolution of the loss function for DICOD, SeqDICOD, CD, FCSC and Fista while solving sparse coding for a signal generated with default parameters, relatively to time. This highlights the speed of the algorithm on the given problem.

to reproduce the figures is available online [1]. The run time denotes the time for the system to run the full algorithm pipeline, from cold start and includes for instance the time to start the sub-processes.

5.5.1 Long convolutional Sparse Coding Signals

To further validate our algorithm, we generate signals and test the performances of DICOD compared to state-of-the-art methods proposed to solve convolutional sparse coding. We generate a signal X of length T in $\mathbb{R}^P$ following the model described in (5.1). The K dictionary atoms D_k of length W are drawn as a generic dictionary. First, each entry is sampled from a Gaussian distribution. Then, the pattern is normalized such that $\|D_k\|_2 = 1$. The sparse code entries are drawn from a Bernoulli-Gaussian distribution with Bernoulli parameter $\rho = 0.007$, mean 0 and standard variation $\sigma = 10$. The noise term $\mathcal{E}$ is chosen as a Gaussian white noise with variance 1. The default values for the dimensions are set to $W = 200$, $K = 25$, $P = 7$, $T = 600 \times W$ and we used $\lambda = 1$.

5.5.2 Performances on Artificial Signals

DICOD is compared to the main state-of-the-art optimization algorithms for convolutional sparse coding: Fast Convolutional Sparse Coding (FCSC) from Bristow et al. (2013), Fast Iterative Soft Thresholding Algorithm (FISTA) using Fourier domain computation as described in Wohlberg (2016), the greedy convolutional coordinate descent (CD, Kavukcuoglu et al. 2010) and the randomized coordinate descent (RCD, Nesterov 2012). All these algorithms are described in Section 3.3 of this manuscript and the specific parameters for these algorithms are fixed based on the authors' recommendations. DICOD$_M$ denotes the DICOD algorithm run using M cores. We also include SeqDICOD$_M$, for $M \in \{60, 600\}$, the sequential run of the DICOD algorithm using M segments, as described in Algorithm 5.2. Figure 5.3 presents the evolution of the cost

[1] The code is made available at https://github.com/tommoral/Dicod.

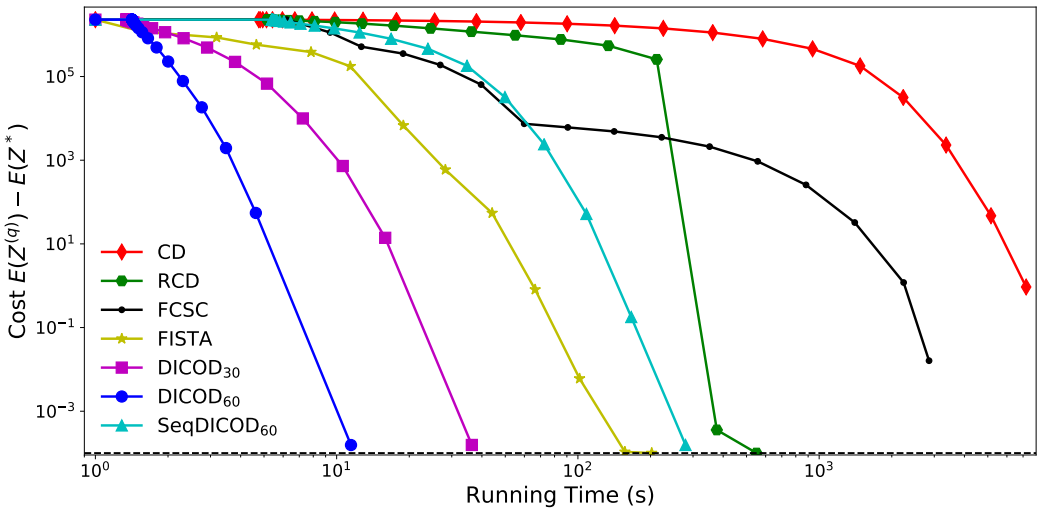


Figure 5.4: Evolution of the loss function for DICOD, SeqDICOD, CD, FCSC and Fista while solving sparse coding for a signal generated with default parameters relatively to the number of iterations.

function value relatively to the number of iterations of the algorithms and Figure 5.4 relatively to the running time. To ensure reasonable computation, a timeout is set to 2hours and it was only reached by the CD algorithm. As the stopping criterion of the different algorithms are not the same, there is a small discrepancy between their resolution but the curves show the general behavior.

Figure 5.3 shows that the evolution of the performances of SeqDICOD relatively to the iterations are very close to the performances of CD. The difference between these two algorithms is that the updates are only locally greedy in SeqDICOD. As there is little difference visible between the two curves, this means that in this case, the computed updates are essentially the same. The differences are larger for SeqDICOD$_{600}$, as the choice of coordinates are more local in this case. The performance of DICOD$_{60}$ and DICOD$_{30}$ are also close to the iteration-wise performances of CD and SeqDICOD. The small differences between DICOD and SeqDICOD result from the iterations where there are interferences. Indeed, if two iterations interfere, the cost does not go down as much as if the iteration where done sequentially. Thus, it requires more steps to reach the same accuracy with DICOD$_{60}$ than with SeqDICOD and with DICOD$_{30}$, as there are more interferences when the number of cores M increases. This explain the discrepancy in the decrease of the cost around the iteration 10^5. However, the number of extra steps required is quite low compared to the total number of steps and the performances are not highly degraded by the interferences. The performances of RCD in terms of iteration are much slower than the greedy methods. Indeed, as only a few coefficients are useful, it takes many iterations to draw them randomly. In comparison, the greedy methods are focused on the coefficients which largely divert from their optimal value, and are thus most likely to be important. Another observation is that the iteration wise performances of the global methods FCSC and FISTA are much better than the methods based on local updates iteration wise. As each iteration can update all the coefficients for FISTA, the number of iterations needed to reach the optimal solution is indeed smaller than for CD, where only one coordinate is updated at a time.

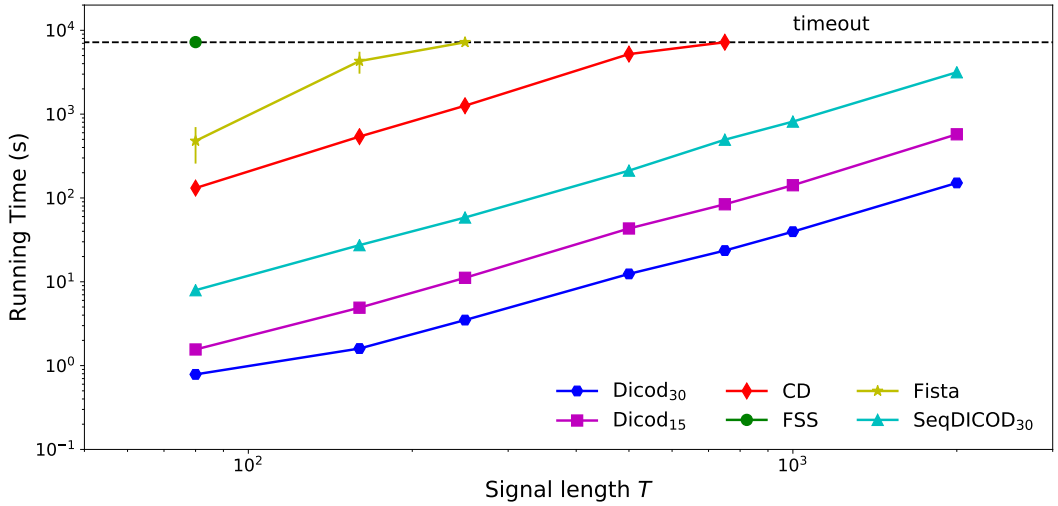


Figure 5.5: Runtime in seconds of the different algorithms averaged over 10 repetitions for different problem sizes, ranging from $80W$ to $2000W$. A timeout was set to 2h.

In Figure 5.4, the speed of theses algorithms can be observed. Even though it needs much more iteration to converge, the randomized coordinate descent is faster than the greedy coordinate descent. Indeed, for very long signals, the iteration complexity of greedy CD is prohibitive. However, using the locally greedy updates, with SeqDICOD$_{60}$ and SeqDICOD$_{600}$, the greedy algorithm can be made more efficient. SeqDICOD$_{600}$ is also faster than the other state-of-the-art algorithms FISTA and FCSC. The choice of $M = 600$ is a good tradeoff for SeqDICOD as it means that the segments are of the size of the dictionary W. With this choice for $M = \frac{T}{W}$, the computational complexity of choosing a coordinate is $\mathcal{O}\left(KW\right)$ and the complexity of maintaining β is also $\mathcal{O}\left(KW\right)$. Thus, the iterations of this algorithm have the same complexity as RCD but are more efficient.

The distributed algorithm DICOD is faster compared to all the other sequential algorithms and the speed up increases with the number of cores. Also, DICOD has a shorter initialization time compared to the other algorithms. The first point in each curve indicates the time taken by the initialization. For all the other methods, the computations for constants – necessary to accelerate the iterations – have a computational cost equivalent to the on of the gradient evaluation. As the segments of signal in DICOD are smaller, the initialization time is also reduced. This shows that the overhead of starting the cores is balanced by the reduction of the initial computation for long signals. For shorter signals, we have observed that the initialization time is of the same order as the other methods. The spawning overhead is indeed constant whereas the constants are cheaper to compute for small signals.

5.5.3 Numerical Complexity

Figure 5.5 displays the running time of each algorithm for different problem sizes, averaged over 10 repetitions. All methods were considered to have converged when the ℓ_∞-norm of the updates reached a certain threshold $\epsilon = 5e^{-2}$. This figure highlights the speedup obtained with the parallelization. The speed up ratio between DICOD$_{15}$ and DICOD$_{30}$ is on average 4.99, and the ratio between DICOD$_{15}$ and CD is on average

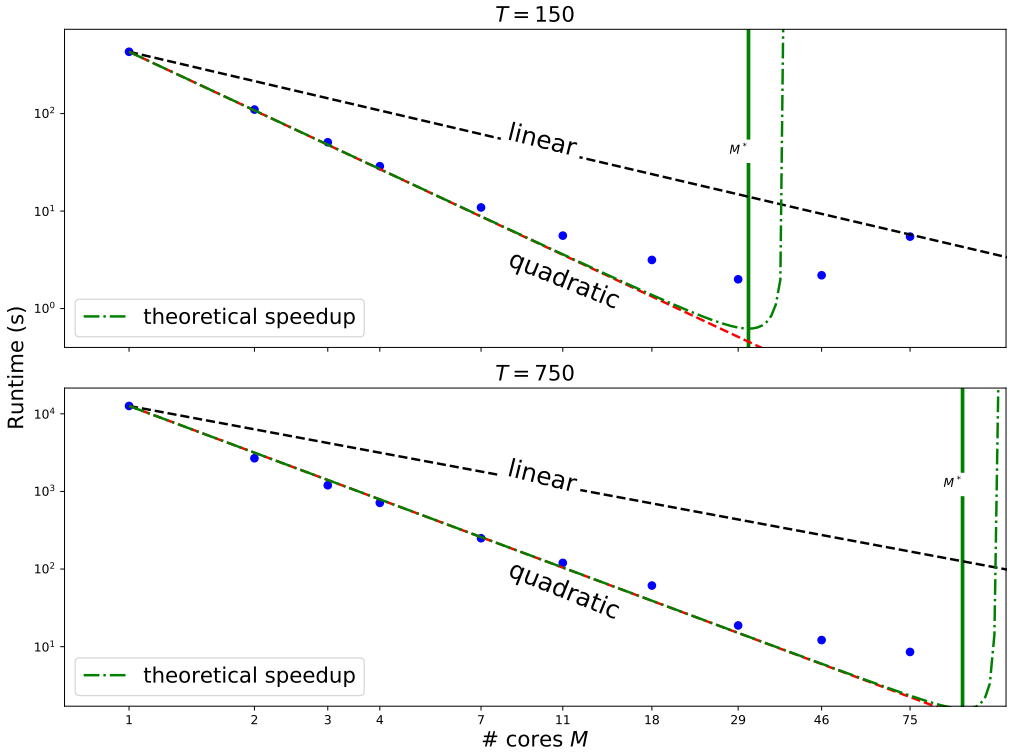


Figure 5.6: Speedup of DICOD as a function of the number of processes used, averaged over 10 runs on different generated signals. This highlights a sharp transition between a regime of quadratic speedups and the regime where the interferences are drastically slowing down the convergence.

91.59. This corroborates the complexity analysis provided in Corollary 5.4 which states that the speedup obtained between DICOD and CD is quadratic in M^2. This plot also shows that DICOD is capable of solving problems with large size ($T = 400000$ samples) in around 3 minutes, whereas the running time of the other algorithms is above two hours.

Figure 5.6 displays the speedup of DICOD as a function of the number of cores. We used 10 generated problems for 2 signal lengths $T = 150 \cdot W$ and $T = 750 \cdot W$ and we solved them using DICOD$_M$ with a number of cores M ranging from 1 to 75. The blue dots display the average running time for a given number of workers. For both setups, the speedup is super-linear up to the point where $M\alpha = \frac{1}{2}$. For small M the speedup is very close to quadratic and a sharp transition occurs as the number of cores grows. The vertical solid green line indicates the approximate position of the maximal speedup given in Corollary 5.4 and the dashed lined is the expected theoretical run time derived from the same expression. The transition after the maximum is very sharp. This approximation of the speedup for small values of $M\alpha$ is close to the experimental speedup observed with DICOD. The computed optimal value of M^* is close to the optimal number of cores in these two examples.

5.6 Discussion

In this chapter, we introduced an asynchronous distributed algorithm, DICOD, which is able to accelerate the solution of the Convolutional Sparse Coding problem for long signals. This algorithm is guaranteed to converge to the optimal solution of (5.2) and scales super-linearly with the number of cores used to distribute it, compared to the greedy coordinate descent. These claims are supported by numerical experiments highlighting the performances of DICOD compared to other state-of-the-art methods. This super-linear speedup results from the inefficiency of the greedy CD for long signals, and running DICOD iteratively also accelerates the solution of problem (5.2). This sequential algorithm, called SeqDICOD, make use of locally greedy updates and is effective in practice.

For the distributed algorithm convergence, our proof relies extensively on the use of one dimensional convolutions. In this setting, a process m only has two neighbors $m-1$ and $m+1$. This ensures that there are no high order interferences between the updates. Our analysis does not apply to distributed computation using square patches of images as the interferences are more complicated. A way to apply our algorithm with its guarantees to images is to only split the signals along one direction, to avoid higher order interferences. The development of a distributed algorithm, using locally greedy updates, for higher order convolution operators is considered for future work. A way to do that would be to compute in each worker an estimate of the β vector in its neighbors and to only perform an update in the interfering zone if it is larger than the estimated updates in the interfering zone of the neighbor. This idea would implement a sort of "soft" lock for the updates, to reduce the probability of interference.

5.7 Proofs

5.7.1 Computation for the Cost Updates

When a coefficient $Z_k[t]$ is updated to $u \in \mathbb{R}$, the cost update is a simple function of $Z_k[t]$ and u.

Proposition 5.7.1. *The update of the weight in (k_0, t_0) from $Z_{k_0}[t_0]$ to $u \in \mathbb{R}$ gives a cost variation:*

$$\begin{aligned}\Delta E_{k_0}[t_0] &= e_{k_0,t_0}(Z_{k_0}[t_0]) - e_{k_0,t_0}(u) \\ &= \frac{\|D_{k_0}\|_2^2}{2}(Z_{k_0}[t_0]^2 - u^2) - \beta_{k_0}[t_0](Z_{k_0}[t_0] - u) + \lambda(|Z_{k_0}[t_0]| - |u|).\end{aligned}$$

with $e_{k,t}(u) = \frac{\|D_k\|_2^2}{2}u^2 - \beta_k[t]u + \lambda|u|$.

Proof. Let $\alpha_{k_0}[t] = \left(X - \sum_{k=1}^K Z_k * D_k\right)[t] + D_{k_0}[t - t_0]Z_{k_0}[t_0]$ for all $t \in [\![0, T-1]\!]$ and

$$Z_k^{(1)}[t] = \begin{cases} u, & \text{if } (k,t) = (k_0, t_0) \\ Z_k[t], & \text{elsewhere} \end{cases}.$$

$$\begin{aligned}e_{k_0,t_0}(u) =& \frac{1}{2}\sum_{t=0}^{T-1}\left(X - \sum_{k=1}^K Z_k * D_k\right)^2[t] + \lambda\sum_{k=1}^K \|Z_k\|_1 \\ &- \frac{1}{2}\sum_{t=0}^{T-1}\left(X - \sum_{k=1}^K Z_k^{(1)} * D_k\right)^2[t] - \lambda\sum_{k=1}^K \|Z_k^{(1)}\|_1 \\ =& \frac{1}{2}\sum_{t=0}^{T-1}\left(\alpha_{k_0}[t] - D_{k_0}[t-t_0]Z_{k_0}[t_0]\right)^2 - \frac{1}{2}\sum_{t=0}^{T-1}\left(\alpha_{k_0}[t] - D_{k_0}[t-t_0]u\right)^2 \\ &+ \lambda(|Z_{k_0}[t_0]| - |u|) \\ =& \frac{1}{2}\sum_{t=0}^{T-1} D_{k_0}[t-t_0]^2(Z_{k_0}[t_0]^2 - u^2) - \sum_{t=0}^{T-1}\alpha_{k_0}[t]D_{k_0}[t-t_0](Z_{k_0}[t_0] - u) \\ &+ \lambda(|Z_{k_0}[t_0]| - |u|) \\ =& \frac{\|D_{k_0}\|_2^2}{2}(Z_{k_0}[t_0]^2 - u^2) - \underbrace{(\widetilde{D_{k_0}} * \alpha_{k_0})[t]}_{\beta_{k_0}[t_0]}(Z_{k_0}[t_0] - u) + \lambda(|Z_{k_0}[t_0]| - |u|)\end{aligned}$$

This concludes our proof. □

Using this result, we can derive the optimal value $Z'_{k_0}[t_0]$ to update the coefficient (k_0, t_0) as the solution of the following optimization problem:

$$Z'_{k_0}[t_0] = \operatorname*{argmax}_{y\in\mathbb{R}} e_{k_0,t_0}(Z_{k_0}[t_0]) - e_{k_0,t_0}(u) \sim \operatorname*{argmin}_{u\in\mathbb{R}} \frac{\|D_{k_0}\|_2^2}{2}\left(u - \frac{\beta_{k_0}[t_0]}{\|D_{k_0}\|_2^2}\right)^2 + \lambda|u| \ . \tag{5.9}$$

In the case where two coefficients $(k_0, t_0), (k_1, t_1)$ are updated in the same iteration to values u and $Z'_{k_1}[t_1]$, we obtain the following cost variation.

Proposition 5.7.2. *The update of the weight $Z_{k_0}[t_0]$ and $Z_{k_1}[t_1]$ to values $Z'_{k_0}[t_0]$ and $Z'_{k_1}[t_1]$ with $\Delta Z_k[t] = Z_k[t] - Z'_k[t]$ gives an update of the cost:*

$$\Delta E_{k_0k_1}[t_0, t_1] = \Delta E_{k_0}[t_0] + \Delta E_{k_1}[t_1] - \mathcal{S}_{k_0,k_1}[t_0 - t_1]\Delta Z_{k_0}[t_0]\Delta Z_{k_1}[t_1]$$

Proof. We define $Z_k^{(1)}[t] = \begin{cases} Z_{k_0}[t_0], & \text{if } (k,t) = (k_0, t_0) \\ Z_{k_1}[t_1], & \text{if } (k,t) = (k_1, t_1) \\ Z_k[t], & \text{otherwise} \end{cases}$

Let $\alpha[t] = \left(X - \sum_{k=1}^K Z_k D_k\right)[t] + D_{k_0}[t - t_0]Z_{k_0}[t_0] + D_{k_1}[t - t_1]Z_{k_1}[t_1]$.

We have $\alpha[t] = \alpha_{k_0}[t] + D_{k_1}[t - t_1]Z_{k_1}[t_1] = \alpha_{k_1}[t] + D_{k_0}[t - t_0]Z_{k_0}[t_0]$.

$$\begin{aligned}
\Delta E_{k_0k_1}[t_0, t_1] =& \frac{1}{2}\sum_{t=0}^{T-1}\left(X - \sum_{k=1}^K Z_k * D_k\right)[t]^2 + \frac{1}{2}\sum_{k=1}^K \lambda\|Z_k\|_1 \\
&- \sum_{t=0}^{T-1}\left(X - \sum_{k=1}^K Z_k^{(1)} * D_k\right)^2[t] - \lambda\sum_{k=1}^K \|Z_k^{(1)}\|_1 \\
=& \frac{1}{2}\sum_{t=0}^{T-1}\Big(\alpha[t] - D_{k_0}[t - t_0]Z_{k_0}[t_0] - D_{k_1}[t - t_1]Z_{k_1}[t_1]\Big)^2 \\
&- \frac{1}{2}\sum_{t=0}^{T-1}\Big(\alpha[t] - D_{k_0}[t - t_0]Z'_{k_0}[t_0] - D_{k_1}[t - t_1]Z'_{k_1}[t_1]\Big)^2 \\
&+ \lambda(|Z_{k_0}[t_0]| - |Z'_{k_0}[t_0]|) + \lambda(|Z_{k_1}[t_1]| - |Z'_{k_1}[t_1]|) \\
=& \frac{1}{2}\sum_{t=0}^{T-1}\Big[D_{k_0}[t - t_0]^2(Z_{k_0}[t_0]^2 - Z'_{k_0}[t_0]^2) + D_{k_1}[t - t_1]^2(Z_{k_1}[t_1]^2 - Z'_{k_1}[t_1]^2)\Big] \\
&- \sum_{t=0}^{T-1}\Big[\alpha_{k_0}[t]D_{k_0}[t - t_0]\Delta Z_{k_0}[t_0] + \alpha_{k_1}[t_1]D_{k_1}[t - t]\Delta Z_{k_1}[t_1] \\
&\qquad + D_{k_0}[t - t_0]D_{k_1}[t - t_1](\Delta Z_{k_0}[t_0]Z'_{k_1}[t_1] + \Delta Z_{k_1}[t_1]Z'_{k_0}[t_0]) \\
&\qquad - D_{k_0}[t - t_0]D_{k_1}[t - t_1](Z_{k_0}[t_0]Z_{k_1}[t_1] - Z'_{k_0}[t_0]Z'_{k_1}[t_1])\Big] \\
&+ \lambda(|Z_{k_0}[t_0]| - |Z'_{k_0}[t_0]| + |Z_{k_1}[t_1]| - |Z'_{k_1}[t_1]|) \\
=& \Delta E_{k_0}[t_0] + \Delta E_{k_1}[t_1] \\
&- \sum_{t=0}^{T-1} D_{k_0}[t - t_0]D_{k_1}[t - t_1]\Big[Z_{k_0}[t_0]Z_{k_1}[t_1] - Z'_{k_0}[t_0]Z_{k_1}[t_1] \\
&\qquad - Z_{k_0}[t_0]Z'_{k_1}[t_1] + Z'_{k_1}[t_1]Z'_{k_0}[t_0]\Big] \\
=& \Delta E_{k_0}[t_0] + \Delta E_{k_1}[t_1] \\
&- \sum_{t=0}^{T-1} D_{k_0}[t]D_{k_1}[t + t_0 - t_1](Z_{k_0}[t_0] - Z'_{k_0}[t_0])(Z_{k_1}[t_1] - Z'_{k_1}[t_1]) \\
=& \Delta E_{k_0}[t_0] + \Delta E_{k_1}[t_1] - \widetilde{D}_{k_0} * D_{k_1}[t_0 - t_1]\Delta Z_{k_0}[t_0]\Delta Z_{k_1}[t_1]
\end{aligned}$$

By definition of $\mathcal{S}_{k_0,k_1}[t] = \widetilde{D}_{k_0} * D_{k_1}[t]$. This concludes our proof. □

5.7.2 Intermediate Results

Consider solving a convex problem of the form:

$$\min E(Z) = F(Z) + \sum_{t=0}^{L-1} \sum_{k=1}^{K} g_i(Z_k[t]) \tag{5.10}$$

where F is differentiable and convex, and g_i is convex. Let us first recall a theorem stated and proven in Osher & Li (2009).

Theorem 5.7.3. *Suppose $F(u)$ is smooth and convex, with $\left|\frac{\partial^2 F}{\partial u_i \partial u_j}\right|_\infty \leq M$, and E is strictly convex with respect to any one variable u_i, then the statement that $u = (u_1, u_2, \dots u_n)$ is an optimal solution of (5.10) is equivalent to the statement that every component u_i is an optimal solution of E with respect to the variable u_i for any i.*

In the convolutional sparse coding problem, the function

$$F(Z) = \frac{1}{2}\|X - \sum_{k=1}^{K} Z_k * D_k\|$$

is smooth and convex and its Hessian is constant. The following Lemme 5.7.4, can be used to show that the function E restricted to one of its variables is strictly convex and thus satisfies the condition of 5.7.3.

Lemme 5.7.4. *The function $f : \mathbb{R} \to \mathbb{R}$ defined for $\alpha, \lambda > 0$ and $b \in \mathbb{R}$ by $f(x) = \frac{\alpha}{2}(x-b)^2 + \lambda|x|$ is α-strongly convex.*

Proof. The property of monotone subdifferential states that a function f is α-strongly convex if and only if

$$\forall (x, x'), \quad \langle f(x) - f(x'), x - x' \rangle \geq \alpha \|x - x'\|_2^2$$

Let us define the subdifferential of f:

$$\partial f = \begin{cases} \alpha(x-b) + \lambda \text{sign}(x) & \text{if } x \neq 0 \\ -\alpha b + \lambda t, \text{ for } t \in [-1, 1] & \text{if } x = 0 \end{cases}$$

The inequality is an equality for $x = x'$.
If $x' = 0$, we get for $|t| \leq 1$:

$$\langle \alpha(x-b) + \lambda \text{sign}(x) + \alpha b - \lambda t), x \rangle = \alpha x^2 + \lambda \underbrace{(|x| - tx)}_{\geq 0} \geq \alpha x^2 = \alpha(x - x')^2$$

If $x' \neq 0$, we get:

$$\begin{aligned} \langle \alpha(x - x') + \lambda(\text{sign}(x) - \text{sign}(x'), x - x' \rangle &= \alpha(x-x')^2 + \lambda(|x| + |x'| - \text{sign}(x)x' - \text{sign}(x')x) \\ &= \alpha(x-x')^2 + \lambda \underbrace{(1 - \text{sign}(x)\text{sign}(x'))}_{\geq 0}(|x| + |x'|) \\ &\geq \alpha(x-x')^2 \end{aligned}$$

Thus f is α-strongly convex. □

This can be applied to the function e_k, t defined in (5.9), showing that the problem in one coordinate (k, t) is $\|D_k\|_2^2$-strongly convex.

5.7.3 Proof of Convergence for DICOD (Theorem 5.2)

We define

$$C_{k_0,k_1}[t] = \frac{\mathcal{S}_{k_0,k_1}[t]}{\|D_{k_0}\|_2 \|D_{k_1}\|_2} \tag{5.11}$$

Let us first show how C_{k_0,k_1} controls the interfering cost update.

Proposition 5.1. *For concurrent updates for coefficients (k_0, t_0) and (k_1, t_1) of a sparse code Z, the cost update $\Delta E_{k_0k_1}[t_0, t_1]$ is lower bounded by*

$$\Delta E_{k_0k_1}[t_0, t_1] \geq \Delta E_{k_0}[t_0] + \Delta E_{k_1}[t_1] - 2\frac{\mathcal{S}_{k_0,k_1}[t_0 - t_1]}{\|D_{k_0}\|_2 \|D_{k_1}\|_2} \sqrt{\Delta E_{k_0}[t_0] \Delta E_{k_1}[t_1]}. \tag{5.8}$$

Proof. The problem in one coordinate (k, t) given that all the others are fixed can be reduced to (5.9). Simple computations show that:

$$\Delta E_k[t] = e_{k,t}(Z_k[t]) - e_{k,t}(Z'_k[t]). \tag{5.12}$$

We have shown in Lemme 5.7.4 that $e_{k,t}$ is $\|D_k\|_2^2$-Strong convex. Thus by definition of the strong convexity, and using the fact that $Z'_k[t]$ is optimal for $e_{k,t}$

$$|e_{k,t}(Z_k[t]) - e_{k,t}(Z'_k[t])| \geq \frac{\|D_k\|_2^2}{2}(Z_k[t] - Z'_k[t])^2 \tag{5.13}$$

i.e., $|\Delta Z_k[t]| \leq \frac{\sqrt{2\Delta E_k[t]}}{\|D_k\|_2}$, and the result is obtained using this inequality with Proposition 5.7.2. □

Theorem 5.2. *If the following hypotheses hold*

H1. *For all (k_0,t_0),(k_1,t_1) such that $t_0 \neq t_1$,* $\left| \frac{\mathcal{S}_{k_0,k_1}[t_0 - t_1]}{\|D_{k_0}\|_2 \|D_{k_1}\|_2} \right| < 1$.

H2. *There exists $A \in \mathbb{N}^*$ such that all cores $m \in [\![1, M]\!]$ are updated at least once between iteration i and $i + A$ if the solution is not locally optimal.*

H3. *The delay in communication between the processes is inferior to the update time.*

Then, the DICOD algorithm converges to the optimal solution Z^* of (5.2)

Proof. If several updates $(k_0, t_0), (k_1, t_1), \dots (k_m, t_m)$ are updated in parallel without interference, then the update is equivalent to the sequential updates of each (k_q, t_q). We thus consider that for each step q, without loss of generality that

$$\Delta E^{(q)} = \begin{cases} \Delta E^{(q)}_{k_0}[t_0], & \text{if there is no interference} \\ \Delta E^{(q)}_{k_0k_1}[t_0, t_1], & \text{otherwise} \end{cases}$$

If $\forall (k, t), \Delta Z^{(q)}_k[t] = 0$, then $Z^{(q)}$ is coordinate wise optimal. Using the result from 5.7.3, $Z^{(q)}$ is optimal. Thus if $Z^{(q)}$ is not optimal, $\Delta E^{(q)}_{k_0}[t_0] > 0$.

Using Proposition 5.1 and (**H1**)

$$\Delta E^{(q)}_{k_0k_1}[t_0, t_1] > \left(\sqrt{\Delta E^{(q)}_{k_0}[t_0]} - \sqrt{\Delta E^{(q)}_{k_1}[t_1]} \right)^2 \geq 0 \ ,$$

so the update $\Delta E^{(q)}$ is positive.

The sequence $(E(Z^{(q)}))_n$ is decreasing and bounded by 0. It converges to E^* and $\Delta E^{(q)} \xrightarrow[q\to\infty]{} 0$. As $\lim_{\|Z\|_\infty \to \infty} E(Z) = +\infty$, there exist $M \geq 0, q_0 \geq 0$ such that $\|Z^{(q)}\|_\infty \leq M$ for all $q > q_0$. Thus, there exists a subsequence $(Z^{q_n})_n$ such that $Z^{q_n} \xrightarrow[n\to\infty]{} \bar{Z}$. By continuity of E, $E^* = E(\bar{Z})$

Then, we show that $Z^{(q)}$ converges to a point $\bar{Z}$ such that each coordinate is optimal for the one coordinate problem. By Proposition 5.1, the sequence $(Z^{(q)})_q$ is ℓ_∞-bounded. It admits at least a limit point $Z^{(q_n)} \xrightarrow[n\to\infty]{} \bar{Z}$. Moreover, the sequence $Z^{(q)}$ is a Cauchy sequence for the norm ℓ_∞ as for $n, p > 0$

$$\begin{aligned}\|Z^{(p)} - Z^{(n)}\|_\infty^2 &\leq \frac{2}{\|D\|_{\infty,2}^2} \sum_{l>n} \Delta E^{(l)} \\ &= \frac{2}{\|D\|_{\infty,2}^2} \left(E(Z^{(n)}) - E^*\right) \underset{n\to\infty}{\to} 0\end{aligned}$$

Thus $Z^{(q)}$ converges to $\bar{Z}$.

Let m denote one of the M cores and (k, t) be coordinates in $\mathcal{C}_m$. We consider the function $h_{k,t} : \mathbb{R}^{K\times L} \to \mathbb{R}$ such that

$$h(Z) = Z'_k[t] = \frac{1}{\|D_k\|_2^2} \mathrm{Sh}(\beta_k[t], \lambda) \ .$$

We recall that

$$\beta_k[t](Z) = \left(\widetilde{D}_k * \left(X - \sum_{\substack{k'=1 \\ k'\neq k}}^{K} Z_{k'} * D_{k'} - \Phi_t\left(Z_k\right) * D_k \right) \right)[t]$$

The function $\phi : Z \to \beta_k[t](Z)$ is linear. As Sh is continuous in its first coordinate and $h(Z) = \mathrm{Sh}(\phi(Z), \lambda)$, the function $h_{k,t}$ is continuous. For $(k, t) \in \mathcal{C}_m$, the gap between $\bar{Z}_k[t]$ and $\bar{Z}'_k[t]$ is such that

$$\begin{aligned}|\bar{Z}_k[t] - \bar{Z}'_k[t]| &= |\bar{Z}_k[t] - h_{k,t}(\bar{Z}_k[t])| \\ &= \lim_{q\to\infty} |Z_k^{(q)}[t] - h(Z_k^{(q)}[t])| \\ &= \lim_{q\to\infty} |Z_k^{(q)}[t] - Y_k^{(q)}[t]| \end{aligned} \tag{5.14}$$

Using **(H2)**, for all $q \in \mathbb{N}$, if $Z_k^{(q)}[t]$ is not optimal, there exists $i_q \in [q, q + A]$ such that the updated coefficient at iteration i_q is $(k_{i_q}, t_{i_q}) \in \mathcal{C}_m$. As no updates are done on $\mathcal{C}_m$ coefficients between the updates q and i_q, $Z_k^{(q)}[t] = Z_k^{(i_q)}[t]$. By definition of the update,

$$\begin{aligned}\left|Z_k^{(q)}[t] - Y_k^{(q)}[t]\right| &= \left|Z_k^{(i_q)}[t] - Y_k^{(i_q)}[t]\right| \\ &\leq \left|Z_{k_{i_q}}^{(i_q)}[t_{i_q}] - Y_{k_{i_q}}^{(i_q)}[t_{i_q}]\right| && \text{(greedy updates)} \\ &\leq \frac{\sqrt{2\Delta E^{(i_q)}}}{\|D_{k_{i_q}}\|} \underset{q\to\infty}{\to} 0 && \text{(Proposition 5.1)}\end{aligned}$$

Using this results with (5.14), $\left|\bar{Z}_k[t] - \bar{Y}_k[t]\right| = 0$. This proves that $\bar{Z}$ is optimal in each coordinate. By 5.7.3, the limit point $\bar{Z}$ is optimal for the problem (2).

□

5.7.4 Proof of DICOD Speedup (Theorem 5.3)

Theorem 5.3. *Let $\alpha = \frac{W}{T}$ and $M \in \mathbb{N}^*$. If $\alpha M < \frac{1}{4}$ and if the non-zero coefficients of the sparse code are distributed uniformly in time, the expected speedup $\mathbb{E}[S_{cd}(M)]$ is lower bounded by*

$$\mathbb{E}[S_{cd}(M)] \geq M^2(1 - 2\alpha^2 M^2 \left(1 + 2\alpha^2 M^2\right)^{\frac{M}{2}-1}) \ .$$

This result can be simplified when the interference probability $(\alpha M)^2$ is small.

Corollary 5.4. *The expected speedup $\mathbb{E}[S_{cd}(M)]$ when $(M\alpha)^2 \to 0$ is such that*

$$\mathbb{E}[S_{cd}(M)] \underset{\alpha \to 0}{\gtrsim} M^2(1 - 2\alpha^2 M^2 + \mathcal{O}(\alpha^4 M^4)) \ .$$

Proof. There are two aspects involved in DICOD speedup: the computational complexity and the acceleration due to the parallel updates.

As stated in Section 5.4, the complexity of each iteration for CD is linear with the length of the input signal T. The dominant operation is the one that finds the maximal coordinate. In DICOD, each core runs the same iterations on a segment of size $\frac{T}{M}$. The hypothesis $\alpha M < \frac{1}{4}$ ensures that finding the maxima is the dominant operation. Thus, when CD runs one iteration, one core of DICOD can run M local iterations as the complexity of each iteration is divided by M.

The other aspect of the acceleration is the parallel update of Z. All the cores perform their update simultaneously and each update happening without interference can be considered as a sequential update. Interfering updates do not degrade the cost. Thus, one iteration of DICOD with N_i interference is equivalent to $M - 2 * N_{interf}$ iterations using CD and thus,

$$\mathbb{E}[N_{dicod}] = M - 2 * \mathbb{E}[N_{interf}] \tag{5.15}$$

The probability of interference depends on the ratio between the length of the segments used for each core and the size of the dictionary. If all the updates are spread uniformly on each segment, the probability of interference between 2 neighboring cores is $\left(\frac{MW}{T}\right)^2 = (M\alpha)^2$.

A process can only create one interference with one of its neighbors. Thus, an upper bound on the probability to get exactly $j \in [0, \frac{M}{2}]$ interferences is

$$\mathbb{P}(N_i = j) \leq \binom{\frac{M}{2}}{j} (2\alpha^2 M^2)^j$$

Using this result, we can upper bound the expected number of interferences for the algorithm

$$\begin{aligned} \mathbb{E}[N_{interf}] = \sum_{j=1}^{\frac{M}{2}} j\mathbb{P}(N_{interf} = j) \ , \quad &\leq \quad \sum_{j=1}^{\frac{M}{2}} j \binom{\frac{M}{2}}{j} (2\alpha^2 M^2)^j, \\ &\leq \quad \alpha^2 M^3 \left(1 + 2\alpha^2 M^2\right)^{\frac{M}{2}-1} \ . \end{aligned}$$

Plugging this result in (5.15) gives us:

$$\begin{aligned}\mathbb{E}[N_{dicod}] &\geq M(1-2\alpha^2M^2\left(1+2\alpha^2M^2\right)^{\frac{M}{2}-1}) \ , \\ &\underset{\alpha\to 0}{\gtrsim} M(1-2\alpha^2M^2+\mathcal{O}(\alpha^4M^4)) \ .\end{aligned} \tag{5.16}$$

Finally, by combining the two source of speedup, we obtain the desired result.

$$\mathbb{E}[S_{cd}(M)] \geq M^2(1-2\alpha^2M^2\left(1+2\alpha^2M^2\right)^{\frac{M}{2}-1}) \ .$$

□

Part II

Representation in Deep Networks

In this part, we analyze the link between deep learning methods and sparse representations. We start by recalling the general framework of deep learning in Chapter 6. Then, Chapter 7 introduces the post-training step for deep neural networks. This extra learning step can be used after normal training of a network and provides a boost in performance for neural networks by optimizing the last layer of the network. The main idea behind this step comes from analysis which splits deep models between the first layers, learning general representations of the data and the last layers, which solve the specific task. The post-training ensures that the learned representation is optimally used for this task. In Chapter 8, we analyze the reason why the Learned ISTA models (LISTA) are able to accelerate the resolution of the LASSO problem. LISTA is a model designed to mimic the behavior of ISTA by replacing gradient computations with general linear operations. It is able to solve the sparse coding problem efficiently when the gram matrix of the problem admits a certain sparse factorization. Understanding the theoretical properties is a step to explicit the link between neural networks and dictionary learning models.

6

Interpretability in Deep Learning Models

"To judge is obviously not to understand, because if we understood, we could not judge anymore."

— *André Malraux*

Contents

This chapter starts by presenting the general framework for feedforward neural networks in Section 6.1. These models - very efficient in practice - are often seen as black-boxes with no guarantees on their performance. We describe in Section 6.2 different theoretical results which shed light on the directions pursued to understand deep learning models. Then, in Section 6.3, we review recent research directions aiming to design interpretable neural networks in order to make their decision process less opaque.

6.1 Feedforward Neural Networks

The foundations for artificial neural network models were introduced in the 40s with the work of McCulloch & Pitts (1943). Their article proposes a framework based on computation networks without circles, with non-linear activation of the graph units, which can be considered as the ancestor of the feedforward networks. Following several developments during the 70s, Werbos (1982) described the first efficient learning rules for these models, with backpropagation. With this technique, it became possible to

efficiently compute the cost derivative for the parameters, in a graph with many layers. These models with multiple layers are called deep networks. They were appealing due to their adaptability but because of a lack of large data sets and computational power, they were judged impractical. With the development of parallel computing, the popularization of GPUs and the increase in the available data, neural networks have become more popular in the past decade. Starting with work by Krizhevsky et al. (2012), many tasks involving signals have seen quick progress thanks to the use of deep learning methods.

These techniques rely on a parametrization of the function space which allows for efficient search and is expressive enough to handle a lot of different data. The parametrization is based on the composition of simple parametric functions, arranged in successive *layers*. The model is said to be deep when multiple layers are stacked. The number of layers and their size control the expressiveness of the model. The parameters are then learned during a training phase with an end-to-end algorithm such as stochastic gradient descent (SGD). This class of models is efficient due to the possibility of computing gradient with respect to the parameters using the chain rule applied to simple differentiable functions. The chain rule in this context is named backpropagation and is described in Subsection 6.1.3.

6.1.1 General Framework

The architecture of a network is defined by specifying the number of layers $L \in \mathbb{N}$ and the input space of each of them $\mathcal{X}_l$. Each layer is then specified as an application $\phi_l : \mathcal{X}_l \mapsto \mathcal{X}_{l+1}$, with $\mathcal{X}_{L+1} = \mathcal{Y}$. We denote $\boldsymbol{\mathcal{W}}$ the parameter of the network and $\Phi_{\boldsymbol{\mathcal{W}}}$ the mapping from $\mathcal{X}$ to $\mathcal{Y}$ computed by the network with parameters $\boldsymbol{\mathcal{W}}$. The mapping is defined by composing the ϕ_l functions, *i.e.*

$$\Phi_{\boldsymbol{\mathcal{W}}} = \phi_L \circ \phi_{L-1} \circ \cdots \circ \phi_1 \tag{6.1}$$

This framework is very general and the functions ϕ_l can be chosen in various classes. The design of the network architecture is mostly empirical, based on the performance of these layers for this type of data with past experiments or a validation set. Below, we present two of the most common layer architectures.

Linear Layer with Point-wise Activation. Linear layers are the most basic layers in the deep learning literature. The output of a layer results from the composition of a linear operation and an activation function. Let $\mathcal{X}_l$ be $\mathbb{R}^{d_l}$ for a specified $d_l \in \mathbb{N}$. The layer is parametrized with a matrix $\boldsymbol{W}_l \in \mathbb{R}^{d_l \times d_{l+1}}$ and the activation function ψ_l. We refer the reader to Subsection 6.1.2 for more details about the choice of the activation function. For $x \in \mathbb{R}^{d_l}$, the mapping ϕ_l defined by the l-th layer is computed using

$$\phi_l(x) = \psi_l(\boldsymbol{W}_l x) \ .$$

The parameter of the full network is $\boldsymbol{\mathcal{W}} = \left(\boldsymbol{W}_1, \ldots, \boldsymbol{W}_L\right)$ and the parameter space is $\Pi_{l=1}^{L} \mathbb{R}^{d_l \times d_{l+1}}$. The full network mapping is thus

$$\Phi_{\boldsymbol{\mathcal{W}}}(x) = \psi_L \left(\boldsymbol{W}_L \cdot \psi_{L-1} \left(\boldsymbol{W}_{L-1} \cdot \ldots \psi_1 \left(\boldsymbol{W}_1 x \right) \right) \right) \ . \tag{6.2}$$

When all the layers in a network are linear layers with non-linear activation, the network can also be called a Multi-Layer Perceptron (MLP).

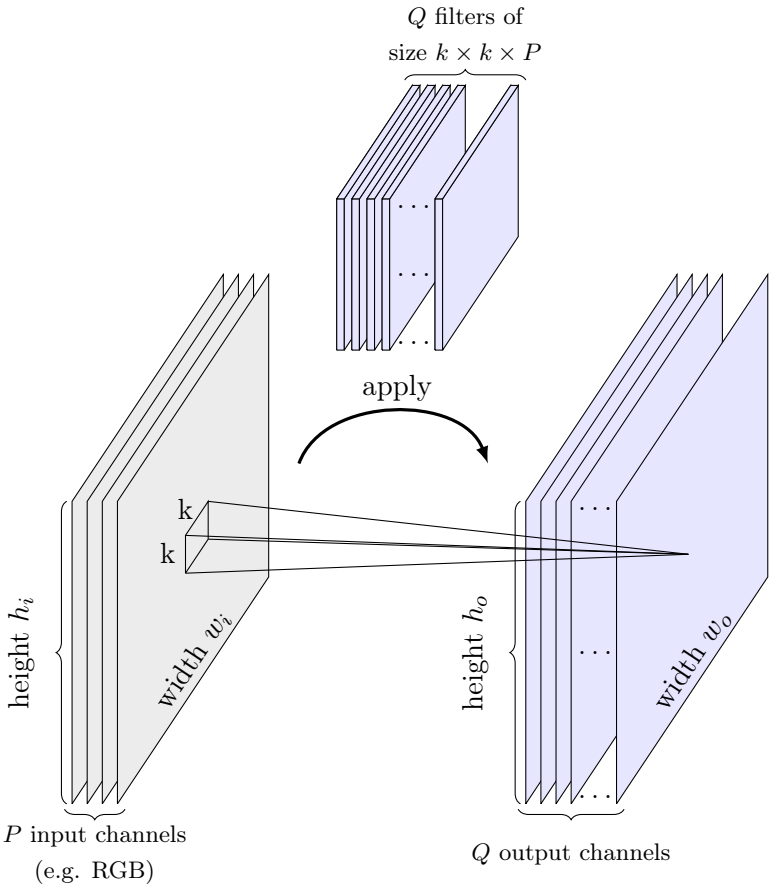


Figure 6.1: A convolutional layer with 2D convolution. The output of the layer is computed with the product of convolution of a linear filter $\mathcal{W}_q$ of size $k \times k$ in dimension P with the input signal in dimension P. Each of the Q linear filters produces one of the output channels.

Convolutional Layer. Convolutional layers are particular case of linear layers where the product operation between matrices is replaced by a convolution product. The most common one is the 2D convolutional layer, used in convolutional neural networks for image recognition. This layer takes as an input a signal X_l in dimension d_l, of size $w_l \times h_l$. It then computes the convolution product of this input with a set of d_{l+1} linear filters $\boldsymbol{\mathcal{W}}_l = \{\mathcal{W}_q\}_{q=1}^{d_{l+1}}$ of size $k_l \times k_l$ in dimension d_l. Each of the d_{l+1} linear filters produces one channel of the output. Figure 6.1 presents this process. The output is then computed using an activation function ψ_l on the result from the convolution product, as in the linear case. The parameter of this layer is a fourth order tensor. For $x \in \mathbb{R}^{w_l \times h_l \times d_l}$, the mapping defined by the l-th layer is

$$\phi_l(x) = \psi_l(\boldsymbol{\mathcal{W}}_l * x) \in \mathbb{R}^{w_{l+1} \times h_{l+1} \times d_{l+1}}$$

The size of the internal representation in the $(l+1)$-th layer is computed using the size of the previous layer and the size of the filters, such that $w_{l+1} = w_l - k_l + 1$ and $h_{l+1} = h_l - k_l + 1$. To keep simple notations, we used squared filters with size $k_l \times k_l$ although these sizes can be chosen independently.

Figure 6.2: Three common activation functions for neural networks.

Figure 6.3: The sigmoïd and the step activation function. The sigmoid can be seen as a smooth version of the hinge activation.

Figure 6.4: RELU activation function and its smooth approximation, the softplus.

6.1.2 Activation Function

Activation functions play an important role in artificial neural networks. Originally, the idea behind the concept of activation was to mimic the behavior of neurons, which have binary responses, depending on their stimulation level. This behavior can be modeled with the step function, which outputs 0 for negative inputs and 1 otherwise. The use of an activation function is critical for the performances of neural networks. When the activation is linear, a neural network also becomes linear and loses its expressiveness.

The important characteristics of the activation function are its non-linearity, efficient gradient or sub-gradient computations and non-null gradients. As stated above, the non-linearity is necessary for the expressiveness of the model. Then, for computational reasons, efficient computation of the gradient or sub-gradient ensures that the activation function can be used in practice on large data sets. Finally, we say that a function has null gradients when its gradient is 0 almost everywhere. Non-null gradients are required to be able to train the model. Figure 6.2 displays three of the most commonly used activation functions: RELU, sigmoïd and hyperbolic tangent (tanh).

Sigmoïd. The step function was the first activation function proposed. It is non-linear but its sub-gradient is 0 almost everywhere. Thus, the step-function is not a practical activation function. To avoid this issue, this activation can be approximated with a smooth and differentiable function called the sigmoïd. The sigmoïd function is the original function which was used as an activation function for MLP. This function is defined for $x \in \mathbb{R}$ as

$$\sigma : x \mapsto \frac{1}{1+e^{-x}}$$

It projects $\mathbb{R}$ on the open segment $]0,1[$. This function is $\mathcal{C}^\infty$ and it is easy to compute its derivative in $x \in \mathbb{R}$ from its value $\sigma(x)$. Indeed,

$$\sigma'(x) = \sigma(x)\Big(1-\sigma(x)\Big) \ .$$

Using this formula with the backpropagation described in Subsection 6.1.3, computing the derivatives of the loss is very efficient.

Rectified Linear Unit (RELU). RELU activation is a piece-wise linear function defined for $x \in \mathbb{R}$ by

$$\text{RELU} : x \mapsto \max(0, x) \ .$$

It was introduced by Hahnloser et al. (2000) based on some symmetry properties of the neural coactivation (Hahnloser et al., 2003). This function is non-linear and its sub-gradient can be computed using

$$\text{RELU}'(x) = \begin{cases} 1, & \text{if x >= 0 ,} \\ 0, & \text{elsewhere .} \end{cases}$$

This function can thus be used directly as an activation function and there is no need to rely on an approximation, unlike for the step-function. The advantage of the RELU activation function over the sigmoïd is that it does not have a vanishing gradient. Indeed, when the neuron activation is far from 0, the sigmoïd gradient approaches 0 exponentially fast. Using the RELU activation, the gradient for positive activation is constant and it is easier to learn the parameters of the model. Note that in cases where the sub-gradient cannot be used, it is possible to approximate the RELU function with a $\mathcal{C}^\infty$ function *softplus*

$$\phi(x) = \log\left(1 + e^x\right) .$$

Softmax. Another important activation function is the soft-max function. When training a classifier, we would like the output of the network to be a discrete probability vector with value 1 for the coordinate encoding the right class and 0 for the other. This behavior can be matched with a max activation function, which sets the maximal output to 1 and the other to 0. But this function is impractical as it has null gradient. An approximated version of the maximum was proposed with *softmax*. This activation function is not point-wise but takes a vector $x \in \mathbb{R}^d$ and output $y \in \mathbb{R}^d$ such that for $i \in [\![1, d]\!]$,

$$y_i = \frac{e^{x_i}}{\sum_{j=1}^d e^{x_j}} .$$

The output y can be seen as a discrete probability vector, with each coordinate i representing the probability that the input is taken in class i. Indeed, its coordinates sum up to 1. Moreover, the soft-max function tends to make the maximal coordinate in x even bigger in y, which is the desired behavior. This activation function is often used for the last layer of a classifier network, in combination with the cross entropy function.

6.1.3 Back-propagation

The efficiency of these methods is dependent on the computational complexity of the gradient computation. One key element has been the formalization of computation rules tagged as *backpropagation* which are easy to implement and require few operations. Backpropagation relies on the chain rule to compute the gradient in the network. Indeed, if the derivative of the training cost $\mathcal{E}_n$ relative to the output h_{l+1} of a given layer $l+1$ is known, it is easy to compute the derivative compared to the previous layer output h_l using the chain rule,

$$\frac{\partial \mathcal{E}_n}{\partial h_l} = \frac{\partial \mathcal{E}_n}{\partial h_{l+1}} \frac{\partial h_{l+1}}{\partial h_l} .$$

The computation of $\frac{\partial h_{l+1}}{\partial h_l}$ depends only on the function ϕ_l and the parameter of layer l. The name backpropagation comes from this backward recursion, which allows computing the gradient of the layer l using the gradient of the layer $l+1$.

Algorithm 6.1 Stochastic Gradient Descent (SGD)

1: **Input:** $\mathcal{D}$, initial parameter $\boldsymbol{\mathcal{W}}^{(0)}$, learning rate $\alpha > 0$
2: **repeat**
3: Randomly select a pair (x_n, y_n) in $\mathcal{D}$
4: Compute $\nabla \mathcal{E}_n(W^{(q)})$
5: Update the current parameter: $\boldsymbol{\mathcal{W}}^{(q+1)} = \boldsymbol{\mathcal{W}}^{(q)} - \alpha \nabla E_n \left(\boldsymbol{\mathcal{W}}^{(q)}\right)$
6: **until** convergence

6.1.4 Stochastic Gradient Descent

Supervised training of neural networks is a very complicated task, due to the high dimension of the parameter space and the fact that the minimization problem involved is non-convex. When the loss is bounded below, local minimums exist in the loss surface. Gradient descent can be used in this case to find a local minimum. In the following, $\mathcal{P} = (X, Y)$ denotes the input distribution of our model in $(\mathcal{X}, \mathcal{Y})$, with $\mathcal{X}$ the input space of the network and $\mathcal{Y}$ the labels of the supervised task. $\mathcal{D} = (x_i, y_i)_{i=1}^N$ is a training set drawn from this distribution. The given task is to minimize the following loss

$$\mathcal{E}_o(\boldsymbol{\mathcal{W}}) = \mathbb{E}_{(x,y)\sim\mathcal{P}} \left[\ell(\Phi_{\boldsymbol{\mathcal{W}}}(x), y)\right] \ , \tag{6.3}$$

for a given function $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}^+$, convex and continuous, and for $\boldsymbol{\mathcal{W}}$ in a given parameter space Θ, determined by the architecture of the network. In the training phase, we do not have access to the input distribution $\mathcal{P}$. The stochastic gradient descent (SGD) estimates the gradient of $\mathcal{E}_o$ by randomly selecting a sample (x_n, y_n) of the data set $\mathcal{D}$ and computing the gradient of

$$\mathcal{E}_n\left(\boldsymbol{\mathcal{W}}\right) = \ell\left(\Phi_{\boldsymbol{\mathcal{W}}}(x_n), y_n\right) \ .$$

The minimization of (6.3) is then performed using Algorithm 6.1. This algorithm estimates the gradient of the true oracle function $\mathcal{E}_o$ by sampling in the distribution $\mathcal{P}$ using $\mathcal{D}$ as a proxy.

Minibatch SGD. The variance of the estimation of $\nabla \mathcal{E}_o$ with only one point from $\mathcal{P}$ can be high and lead to slow convergence. A way to reduce the variance is to use more samples to estimate the gradient. The mini-batch SGD selects $\bar{N}$ samples $\{x_n\}_{n=1..\bar{N}}$ from $\mathcal{D}$ instead of only one and estimates the gradient of $\mathcal{E}_o$ by computing the gradient of

$$\frac{1}{\bar{N}} \sum_{i=1}^{\bar{N}} \mathcal{E}_n\left(\boldsymbol{\mathcal{W}}\right) \ .$$

The gradient computed with this technique is the average of the gradients computed for each element selected. This effectively reduces the variance of the estimation of the oracle gradient. Moreover, as the computations for each element is independent, each gradient can be computed in parallel.

6.2 Theoretical properties of neural networks

In their paper, Bottou & Bousquet (2008) introduce a decomposition of the empirical risk error for a given task which splits the error between three independent sources.

We consider a space of input-output pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ with an input distribution $\mathcal{P}$. The considered task is given with a loss function ℓ which measures the discrepancy between the predicted output $\hat{y}$ and the ground-truth y. The goal of the model is to estimate the function f^* which minimizes the expected risk

$$f^* = \underset{f}{\operatorname{argmin}}\, E(f) \triangleq \mathbb{E}_{\mathcal{P}}\left[\ell(f(x), y)\right]$$

The model is usually chosen in a class of function $\mathcal{F}$ which might not contain f^* and the function which minimizes the expected risk in this class of function is denoted $f_{\mathcal{F}}$. Moreover, as the input distribution $\mathcal{P}$ is unknown, it is not possible to directly minimize E. As described in Subsection 6.1.4, training is performed using the empirical risk minimization for the function

$$E_N(f) = \frac{1}{N}\sum_{n=1}^{N} \ell\left(f(x^{(n)}), y^{(n)}\right)$$

for points $(x^{(n)}, y^{(n)}) \in \mathcal{D}$. We denote $f_N^* = \operatorname{argmin}_{f\in\mathcal{F}} E_N(f)$. For a given model $\bar{f}_N$, the error made is computed relatively to the best possible model f^*, and can be split in three parts

$$E(\bar{f}_N) - E(f^*) = \underbrace{E(\bar{f}_N) - E(f_N^*)}_{\mathcal{E}_{opt}} + \underbrace{E(f_N^*) - E(f_{\mathcal{F}})}_{\mathcal{E}_{est}} + \underbrace{E(f_{\mathcal{F}}) - E(f^*)}_{\mathcal{E}_{app}},$$

These three sources of error are

- **Approximation error $\mathcal{E}_{app}$:** This error measures the discrepancy between the class of model $\mathcal{F}$ considered and the optimal function f^*. If f^* is in $\mathcal{F}$, this error can be zero. Most of the time, due to computational restrictions, the chosen function class $\mathcal{F}$ does not contain the optimal solution and this error measures how far our model class is from the data generation process. To reduce this error term, it is necessary to use a larger functional space $\mathcal{F}$, but it will also increase the computational cost of learning the model. The model class $\mathcal{F}$ must be chosen with the tradeoff between computations and approximation in mind.

- **Estimation error $\mathcal{E}_{est}$:** This error measures the effect of the approximation of E by E_N during the training. This error term is directly tied to the size of the training set N and the class of functions $\mathcal{F}$. For classes of functions linearly parametrized with parameters in $\mathbb{R}^d$, the error between E and E_N scales with $\mathcal{O}\left(\sqrt{\frac{d}{N}}\right)$. Using this result, it is possible to show the same rate for $\mathcal{E}_{est}$ and thus the estimation error is reduced when N increases. The growth of N also increases the computational cost for fitting the model as E_N becomes more complex. Also, if the class of functions $\mathcal{F}$ becomes more complex, for instance via an increase in d, this error term can grow. The tradeoff involved in choosing $\mathcal{F}$ must include the estimation error with the computational complexity.

- **Optimization error $\mathcal{E}_{opt}$:** The error of optimization results from the inexact minimization performed by numerical optimization algorithms. When using a numerical optimization method with finite computation time, the minimizer of $\bar{f}_N$ is computed up to a precision $\rho \geq 0$ such that

$$E_N(\bar{f}_N) < E_N(f_N^*) + \rho$$

		$\mathcal{F}\nearrow$	$N\nearrow$	$\rho\nearrow$
$\mathcal{E}_{app}$	(approximation error)	$\searrow$		
$\mathcal{E}_{est}$	(estimation error)	$\nearrow$	$\searrow$	
$\mathcal{E}_{opt}$	(optimization error)	...	...	$\nearrow$
Q	(computational complexity)	$\nearrow$	$\nearrow$	$\searrow$

Table 6.1: Typical variations when $\mathcal{F}$, N and ρ increases. (adapted from Bottou & Bousquet 2008)

This parameter ρ controls the scale of the optimization error and the computational cost of the learning process, since the smaller it is, the more computationally heavy the optimization method is.

Finally, Table 6.1 summarizes the typical variations when the class of function $\mathcal{F}$, the number of training samples N and the optimization resolution ρ increases. The choice of these parameters is done by matching the scales of the three error sources, while maintaining a reasonable computational cost $Q < Q_{\max}$. Recent theoretical results on deep learning can be classified according to the error sources they analyze.

6.2.1 Approximation Error

The first theoretical properties of deep learning models have been proposed in the late 80s (Cybenko, 1989; Hornik et al., 1989; Hornik, 1991; Barron, 1993). These results prove the high expressiveness of deep learning with universal approximation theorems. Hornik (1991) showed that multilayered feedforward networks are universal approximators provided that sufficiently many hidden units are available and under very general conditions on the hidden unit activation function. These results are not constructive, as they do not quantify the number of hidden units needed to approximate a given function, and they do not show if the parameter of such a network can be estimated with end-to-end procedures such as SGD. In recent literature, Telgarsky (2016) showed the benefit of depth for neural networks. His results show that an exponential number of hidden units is necessary for a low depth fully connected network to approximate a deeper fully connected network with $\mathcal{O}(1)$ hidden units. The expressiveness advantage of deep networks compared to shallower networks are also demonstrated in Montúfar et al. (2014). Cohen et al. (2017) also show similar results for convolutional neural networks, and show that these networks are efficient to approximate a large class of functions. Their results on the effect of depth, widths, pooling, convolution geometry and interconnectivity provide guidelines for network design. Finally, the study of mathematical properties of deep networks as feature extractors, pioneered by Mallat (2012), showed the invariance properties of the functions learned with deep architectures (Bruna & Mallat, 2013; Mallat, 2016; Wiatowski & Bölcskei, 2018).

6.2.2 Estimation Error

The estimation error can be controlled with the Vapnik-Chervonenkis (VC) theory (Vapnik & Chervonenkis, 1971) and Bartlett & Maass (2003) showed that the VC dimension of a neural network grows polynomially with the number of parameter in the network. This result fails to explain the fact that neural networks with far more parameter than training samples generalize well (Caruana et al., 2001; He et al., 2016). The

estimation error has received a lot of attention recently, to explain the generalization properties of the neural networks. The effect of the regularization have been studied for dropout strategies (Wan et al., 2012; Srivastava et al., 2014). Neyshabur (2017) and Zhang et al. (2017) show the role of explicit and implicit regularization of the parameters in controlling the estimation error. Another line of work study the invariance properties of the neural networks to explain their generalization (Giryes et al., 2016b; Sokolic et al., 2017; Liang et al., 2017). The impact of the optimization strategy on the generalization properties of the resulting network has also been studied. In their paper, Keskar et al. (2017) explore the sharpness of the local minimizer found using SGD and empirically demonstrate that using larger batches in the SGD leads to worse performance on the test sets compared to the same algorithm, with the same initialization, but smaller batch size. The results in this work are not so clear, as pointed out by Dinh et al. (2017) which showed that it was possible to arbitrarily change the sharpness of the minimum using re-parametrization of the network. Using the *Information Plane* visualization of the training (Tishby & Zaslavsky, 2015), Schwartz-ziv & Tishby (2017) highlight the dynamic of the estimation error during the training procedure. Neyshabur et al. (2015) propose a modified SGD algorithm to improve the generalization properties of the network.

6.2.3 Optimization Error

Another challenge of the deep learning methods is how to efficiently solve a highly complex and non-convex optimization problem. Learning the network parameters involves a non-convex optimization problem in very high dimension and Dauphin et al. (2014) show that the number of critical points grows exponentially with the dimension. Previous works showed examples of success and failures depending on the characteristics of the trained network, input distributions or class of estimated functions (Baldi & Hornik, 1989; Brady et al., 1989; Gori & Tesi, 1991; Frasconi et al., 1997; Andoni et al., 2014). Thus, the choice of an appropriate training strategy is of paramount importance, as small mistakes can drive the algorithm into a sub-optimal local minimum, resulting in poor performance (Bengio & LeCun, 2007). Most of the training algorithms are derived from the SGD and it was recently shown that these algorithms are guaranteed to converge to local minimum of the loss (Lee et al., 2016). Novel studies have highlighted new characterization of the loss local minima (Choromanska et al., 2015; Chaudhari & Soatto, 2015; Freeman & Bruna, 2017). In particular Choromanska et al. (2015) show that the ratio of "bad" local minima decreases with the size of the network in certain configurations, and Freeman & Bruna (2017) study the connectedness of the loss level sets for half rectified networks. New training algorithms have been proposed in order to reduce the optimization error. Chaudhari et al. (2017a,b) modify the training loss with the local entropy to get an effective training procedure, with connection to partial differential equations, and Janzamin et al. (2015) which derive an effective training procedure based on the power method for tensor decomposition and give theoretical guarantees with number of samples polynomial in the input dimension and the number of neurons. Haeffele & Vidal (2015) also propose a guaranteed learning procedure and a certificate, to find the global optimum if the size of the network is allowed to vary.

6.2.4 Variance Reduction Strategies

Several variance reduction techniques have been proposed to improve the estimation of the gradient and hence reduce the optimization error. But while these techniques stabilized the convergence of the learning algorithm, their impact on the generalization property of the network is less clear. It has been shown empirically, due to the finite data set, the usage of variance reduction can lead to quicker over-fitting and might increase the estimation error. In practice, these techniques often improve the learning procedure without large impact on the generalization error. We list here three of the most commonly used variance reduction strategies.

Adagrad. The adaptive gradient method (Adagrad) is a technique introduced by Duchi et al. (2011) to reduce the variance of the gradient estimation in SGD. It fixes the learning rate in SGD independently for each parameter, by considering the history of previous gradient updates. For parameter coefficient $\boldsymbol{\mathcal{W}}_i^{(q+1)}$, we keep an auxiliary variable

$$\eta_i^{(q+1)} = \eta_i^{(q)} + \left[\nabla \mathcal{E}_n \left(\boldsymbol{\mathcal{W}}^{(q)}\right)_i\right]^2$$

and the updates are computed using

$$\boldsymbol{\mathcal{W}}_i^{(q+1)} = \boldsymbol{\mathcal{W}}_i^{(q)} - \frac{\alpha}{\sqrt{\eta_i^{(q)}}} \nabla \mathcal{E}_n \left(\boldsymbol{\mathcal{W}}^{(q)}\right)_i .$$

The intuition behind this method is that if some parameters are updated very often with large gradients, the training step is too large.

RMSProp. Hinton et al. (2012) formalized in their online course the Root Mean Square Propagation (RMSProp). This method is a generalization of Adagrad which weights the history of the previous gradients differently. The auxiliary variable for parameter coefficient $\boldsymbol{\mathcal{W}}_i^{(q+1)}$ is defined as

$$\eta_i^{(q+1)} = \gamma_1 \eta_i^{(q)} + (1-\gamma_1) \left[\nabla \mathcal{E}_n \left(\boldsymbol{\mathcal{W}}^{(q)}\right)_i\right]^2 ,$$

with a forgetting parameter $0 < \gamma_1 < 1$, fixed as an input of the algorithm. Then, the updates are computed using the same formula as Adagrad, *i.e.*

$$\boldsymbol{\mathcal{W}}_i^{(q+1)} = \boldsymbol{\mathcal{W}}_i^{(q)} - \frac{\alpha}{\sqrt{\eta_i^{(q)}}} \nabla \mathcal{E}_n \left(\boldsymbol{\mathcal{W}}^{(q)}\right)_i$$

The main difference with Adagrad is the computation of η. In the case of RMSprop, all the gradient history is not given the same weight, as the forgetting factor give more importance to recent updates. With $\gamma_1 = 1$, this method is equivalent to Adagrad.

Adam. Both RMSprop and Adagrad use a biased estimate of the first and second moment of the gradient. In their work, Kingma & Ba (2015) proposed a novel strategy to correct the bias and improve the variance reduction. This strategy is called Adam and uses the following auxiliary variables

$$\nu_i^{(q+1)} = \gamma_2 \nu_i^{(q)} + (1-\gamma_2) \nabla \mathcal{E}_n \left(\boldsymbol{\mathcal{W}}^{(q)}\right)_i ,$$

$$\eta_i^{(q+1)} = \gamma_1 \eta_i^{(q)} + (1-\gamma_1) \left[\nabla \mathcal{E}_n \left(\boldsymbol{\mathcal{W}}^{(q)}\right)_i\right]^2 ,$$

for forgetting factors $0 < \gamma_1, \gamma_2 < 1$. These two quantities are biased estimates of the first and second moment of the gradient. The updates for Adam are computed such that

$$\boldsymbol{W}_i^{(q+1)} = \boldsymbol{W}_i^{(q)} - \alpha \frac{\sqrt{1-\gamma_1^q}\ \ \nu_i^{(q)}}{\left(1-\gamma_2^q\right)\sqrt{\eta_i^{(q)}}} \ .$$

6.3 Interpretability of Deep Learning

6.3.1 Internal Representations in Neural Networks

One of the key aspects for the success of deep learning methods is that the information extraction process is included in the model and trained using an end-to-end procedure. This ensures that the information relevant to the task is selected. These models are composed by chaining different operations. The intermediate computation results, called *hidden units*, constitute internal representations of the data points in successive spaces. The internal representation of the data in the model can be linked to some kind of pattern extraction. For instance, Le et al. (2013) showed that their convolutional neural networks trained on images had representations in their upper layers which detected cats in the raw images. The properties of the learned patterns highly depends on the considered task and the model architecture. With convolutional layers, the neural networks are able to learn and extract local patterns relevant for the application context, from the previous layer. The internal patterns are often hard to retrieve as they are extracted from a chain of intermediate representation. Other architectures use attention mechanisms, introduced by Cho et al. (2015), which learn to focus on specific parts of the signal for specific tasks. The model is trained to distinguish and localize the relevant structures from the raw signal in order to solve a given task. These two type of layers are designed to highlight the local structure in the signals. However, it is often not possible to visualize the intermediate representations of the signals learned by a deep learning model back in the original signal space. These models are often considered as black-box techniques to solve classification or regression problems, without really understanding the decision process. The comprehension and interpretation of the inner representation in neural networks could yield very interesting insights to analyze signal data.

Different works design networks aiming to get interpretable extracted patterns. The first attempt to produce interpretable representation using deep learning has been the development of Deep Belief Networks (DBN), introduced by Hinton et al. (2006). The deep belief networks are composed by stacking Restricted Bolzman Machine (RBM) layers to compute internal representations. Each of these layers consists of an encoding and a decoding part that are trained in an unsupervised manner to try to best reconstruct the input from the encoded representation. By decoding sequentially with the previous layers decoding function a hidden unit representation of an input, the patterns captured by the internal representation can be interpreted in the original input space. Lee et al. (2009) extended the DBN for the convolutional setting, making the learned pattern more local and shift invariant. Another work to capture the patterns learned by neural networks is the Deconvolution Network, designed by Zeiler et al. (2010). This network architecture aims to produce a hierarchy of convolutional sparse representations, like we described in Chapter 3. Each layer encodes the coding signal computed by the previous layer using model (3.1) with a sparsity constraint. The original paper uses this unsupervised mode to perform image denoising and then in a supervised set-

ting to classify images, tweaking the dictionary elements to capture patterns relevant for the task. Also, interpretable networks have been designed using analytic signal processing tools. In their work, Bruna & Mallat (2013) introduce the scattering convolution network. This network uses the wavelet transform to construct successive hierarchical representations. This network's internal representations can be studied as we know their analytical forms and the authors show that this network captures invariant information which improves image classification. This network achieves state-of-the-art performance on hand-written digit recognition without training the representation and for various classification tasks (Sifre, 2014; Andén & Mallat, 2014). The design of new mother-wavelets allows adapting the invariant properties captured by such networks.

6.3.2 Role of the Layers

Another approach to understand the internal representations in deep models is to analyze the role of the different layers (Erhan et al., 2010; Montavon et al., 2011). It has been shown in various cases that the first layers of a deep neural network tend to learn general feature extractors which can be reused in other architectures with the same type of data, independently of the solved task. These layers are qualified as *general*. For example, in image processing, the first layers of convolutional network tend to exhibit features similar to Gabor filters and color blob (Krizhevsky et al., 2012). Similarly, it has been shown that the last layers of the network greatly depend on the chosen dataset and task, and are referred to as *specific* layers. The idea of general and specific layers has been successfully applied to transfer learning. In Yosinski et al. (2014), the authors study the properties of transferred layers by *freezing* some layers, *i.e.* treating their weights as constants. They show that general layer parameters can be reused as initialization for the first layers of a different architecture, providing a head start in the network training. Inversely, reusing specific layers does not help the performance of the network.

6.3.3 Interpreting the Model

Neural networks can also be considered in the optimization context, as it is shown in Rozell et al. (2008). In this paper, the authors design a network architecture composed of linear layers and soft-thresholding activation functions to approximately solve the sparse coding problem (3.8). The architecture of this network is designed to match the computation steps of the ISTA algorithm and the weights can be learned to get good approximation of the sparse code (Gregor & Lecun, 2010). A similar analysis has been conducted for the block coordinate descent in Sprechmann et al. (2012) and for ISTA and FISTA applied to convolutional sparse coding (Goroshin, 2015, Chapter 4). While the design of these architectures are specific to the considered optimization algorithms, the kind of layers used are relatively common. Indeed, convolutional and linear layers have been the basic layers in neural networks and the non-linearity involved in these networks are close to RELU for the soft-thresholding and to the max-pooling for the greedy coordinate descent. This approach of the neural networks, and its link to sparse coding optimization algorithms, has been used in Eigen et al. (2014) to study the influence of the network architecture design.

This link between optimization algorithms and neural networks can also be used in order to link neural networks with sparse representations. Indeed, the LISTA network is close to the generic architecture of feedforward networks. Thus, the internal representation of the networks can be interpreted as successive estimates of the solution of a sparse

coding problem, for a given dictionary. If it is possible to compute such a dictionary, based on the properties of LISTA, the internal representation can then be interpreted in the input space. As LISTA architecture can be modified to solve the convolutional sparse coding by replacing the linear layers by convolutional layers, it is also possible to make the same link between convolutional neural networks and convolutional sparse representations.

7

Post-training for Deep Learning Models

"Ce qui est difficile, c'est la partie pédalo, pas la partie canard. Le canard, c'est un bec, un col vert... Bon, n'en parlons plus!"

– OSS 117

Contents

In this chapter, we propose an extra training step, called post-training, which only optimizes the last layer of the network. The goal of this step is to make sure that the embedding, or representation, of the data is used as well as possible by the model to solve the considered task. This procedure can be analyzed in the kernel method framework, with the first layers computing an embedding of the data used by the last layer to solve the task with a simple statistical model. This idea is then tested on multiple architectures with various data sets and provides a small boost in performance.

7.1 Training Neural Networks

One of the main challenges of the deep learning methods is to efficiently solve the highly complex and non-convex optimization problem involved in the training step. Many parameters influence the performances of trained networks, and small mistakes can drive the algorithm into a sub-optimal local minimum, resulting into poor performances (Bengio & LeCun, 2007). Consequently, the choice of an appropriate training strategy is critical to the usage of deep learning models.

The most common approach to train deep networks is to use the stochastic gradient descent (SGD) algorithm described in Subsection 6.1.4. This method selects a few points in the training set, called a batch, and computes the gradient of a cost function with respect to all the parameters of all layers and uses this gradient to update the weights of all layers. Empirically, this method often converges to a local minimum of the cost function that has good generalization properties. Stochastic updates are used to estimate the gradient of the error on the input distribution, and variance reduction techniques such as Adagrap (Duchi et al., 2011), RMSprop (Hinton et al., 2012) or Adam (Kingma & Ba, 2015) have been proposed to achieve faster convergence (see Subsection 6.2.4).

While these algorithms converge to a local minima, this minima is often influenced by the properties of the initialization used for the network weights. A frequently used approach to find a good starting point is to use pre-training (Larochelle et al., 2007; Hinton et al., 2006; Hinton & Salakhutdinov, 2006). This method iteratively constructs each layer by training them as auto-encoders, using continuous extensions of the Restricted Bolzman Machine (RBM). This unsupervised learning ensures that hidden units capture the information from the data. The network is then fine-tuned using SGD to solve the task at hand. Pre-training strategies have been applied successfully to many applications, such as classification tasks (Bengio & LeCun, 2007; Poultney et al., 2006), regression (Hinton & Salakhutdinov, 2008), robotics (Hadsell et al., 2008) or information retrieval (Salakhutdinov & Hinton, 2009). The influence of different pre-training strategies over the different layers has been thoroughly studied in Larochelle et al. (2009). In addition to improving the training strategies, these works also shed light onto the role of the different layers (Erhan et al., 2010; Montavon et al., 2011). The first layers of a deep neural network, qualified as *general*, tend to learn feature extractors which can be reused in other architectures, independently of the solved task. Meanwhile, the last layers of the network are much more dependent of the task and data set, and are said to be *specific* (see Subsection 6.3.2).

The idea of general and specific layers has been successfully applied to transfer learning (Yosinski et al., 2014; Oquab et al., 2014; Razavian et al., 2014). Yosinski et al. (2014) study the properties of transferred layers by *freezing* some layers, *i.e.* treating their weights as constants. They show that general layer parameters can be reused as initialization for the first layers of a different architecture, providing a head start in the network training. Inversely, reusing specific layers do not help the performance of the network. Razavian et al. (2014) show that reusing the features learned using the varied ImageNet data set (Jia Deng et al., 2009) and the `Overfeat` neural network (Sermanet et al., 2014) with a Support Vector Machine (SVM) provide a competitive base line for various classification tasks. These techniques used in transfer learning are the same as the one used in our study as the main idea is to re-use the representation computed by early layers to solve a given task. The main difference is the aim of the method. While transfer learning focuses on the good initialization of networks for novel tasks, we propose to use the technique to improve the performance of the original network.

Deep architectures generally achieve better results than shallow structures, but the later are generally easier to train as optimization algorithm are more stable. When the representation of the data is fixed, the training problem for convex model such as the logistic regression is also convex. When the representation is learned simultaneously, for instance with dictionary learning or with EM algorithms, the problem often become non-convex. The separation between the representation and the model learning is a

key ingredient to make the model easily trainable. But this coupling between the representation and the model is critical for end-to-end models. For instance, Hinton et al. (2006) showed that for networks trained using pre-training, the fine-tuning step – where all the layers are trained together – improves the performances of the network. This shows the importance of the adaptation of the representation to the task in end-to-end models.

Our contribution in this chapter is an additional training step which improves the use of the representation learned by the network to solve the considered task. This new step is called *post-training*. It is based on the idea of separating representation learning and statistical analysis and it should be used after the training of the network. In this step, only the specific layers are trained. Since the general layers – which encode the data representation – are fixed, this step focuses on finding the best usage of the learned representation to solve the desired task. In particular, we chose to study the case where only the last layer is trained during the post-training, as this layer is the most specific one (Yosinski et al., 2014). In this setting, learning the weights of the last layer corresponds to learning the weights for the kernel associated to the feature map given by the previous layers. The post-training scheme can thus be interpreted in light of kernel methods. To summarize our contributions:

- We introduce a post-training step, where all layers except the last one are frozen. This method can be applied after any traditional training scheme for deep networks. Note that this step does not replace the end-to-end training, which co-adapts the last layer representation with the solver weights, but it makes sure that this representation is efficiently used by the model to solve the given task.
- We show that this post-training step is easy to use, that it can be effortlessly added to most learning strategies, and that it is computationally inexpensive.
- We highlight the link existing between this method and the kernel techniques. We also show numerically that the previous layers can be used as a kernel map when the problem is small enough.
- We experimentally show that the post-training often produces a small improvement for various architectures and data sets.

The rest of the chapter is organized as follows: Section 7.2 introduces the post-training step and Section 7.3 discusses its relation with kernel methods. Section 7.4 presents our numerical experiments with multiple neural network architectures and data sets and Section 7.5 discusses these results.

7.2 Post-training

In this section, we consider a feedforward neural network with L layers, where $\mathcal{X}_1, \dots, \mathcal{X}_L$ denote the input space of the different layers, typically $\mathbb{R}^{d_l}$ with $d_l > 0$ and $\mathcal{Y} = \mathcal{X}_{L+1}$ the output space of our network. Let $\phi_l : \mathcal{X}_l \mapsto \mathcal{X}_{l+1}$ be the applications which respectively compute the output of the l-th layer of the network, for $1 \leq l \leq L$, using the output of the $l-1$-th layer and $\Phi_L = \phi_L \circ \cdots \circ \phi_1$ be the mapping of the full network from $\mathcal{X}_1$ to $\mathcal{Y}$. Also, for each layer l, we denote $\boldsymbol{W}_l$ its weights matrix and ψ_l its activation function.

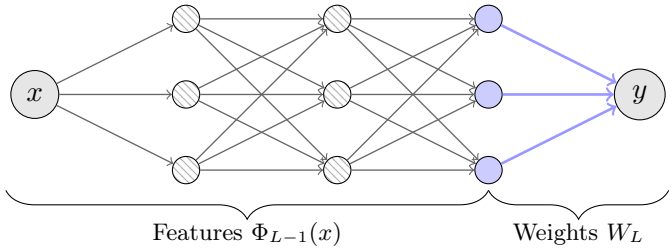


Figure 7.1: Illustration of post-training applied to a neural network. During the post-training, only the weights of the blue edges are updated. The blue nodes can be seen as the embedding of x in the feature space $\mathcal{X}_L$.

The training of our network is done using a convex and continuous loss function $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}^+$. The objective of the neural network training is to find weights parametrizing Φ_L that solves the following problem:

$$\min_{\Phi_L} \mathbb{E}_{(x,y)\sim\mathcal{P}} \left[\ell \left(\Phi_L(x), y \right) \right] + \Omega(\Phi_L) \ , \tag{7.1}$$

for a certain input distribution $\mathcal{P}$ in $(\mathcal{X}_1, \mathcal{Y})$ and a regularization function Ω. The training set is $\mathcal{D} = \left(x_i, y_i\right)_{i=1}^N$, drawn from this input distribution.

Using these notations, the training objective (7.1) can then be rewritten

$$\min_{\Phi_{L-1}, \boldsymbol{W}_L} \mathbb{E}_{(x,y)\sim\mathcal{P}} \left[\ell \left(\psi_L \left(\boldsymbol{W}_L \Phi_{L-1}(x) \right), \ y \right) \right] + \Omega(\Phi_{L-1}, \boldsymbol{W}_L) \ . \tag{7.2}$$

This reformulation highlights the special role of the last layer in our network compared to the others. When Φ_{L-1} is fixed, the problem of finding $\boldsymbol{W}_L$ is simple for several popular choices of activation function ψ_L and loss ℓ . For instance, when the activation function ψ_L is the `softmax` function and the loss ℓ is the `cross entropy`, (7.2) is a multinomial logistic regression. In this case, training the last layer is equivalent to a regression of the labels y using the embedding of the data x in $\mathcal{X}_L$ by the mapping Φ_{L-1} . Since the problem is convex in $\boldsymbol{W}_L$ (see Section 7.6), classical optimization techniques can efficiently produce an accurate approximation of the optimal weights $\boldsymbol{W}_L$ – and this optimization given the mapping Φ_{L-1} is the idea behind post-training.

Indeed, during the regular training, the network tries to simultaneously learn a suitable representation for the data in the space $\mathcal{X}_L$ through its $L-1$ first layer and the best use of this representation with $\boldsymbol{W}_L$. This joint minimization is a non-convex problem, therefore resulting in a potentially sub-optimal usage of the learned data representation.

The post-training is an additional step of learning which takes place after the regular training and proceeds as follows:

1. **Regular training:** This step aims to obtain interesting features to solve the initial problem, as in any usual deep learning training. Any training strategy can be applied to the network, optimizing the empirical loss

$$\operatorname*{argmin}_{\Phi_L} \frac{1}{N} \sum_{i=1}^{N} \ell \left(\Phi_L(x_i), y_i \right) + \Omega \left(\Phi_L \right) \ , \tag{7.3}$$

where $\Omega(\Phi_L)$ is a regularization term for the network's parameters. The stochastic gradient descent explores the parameter space and provides a solution for Φ_{L-1} and $\boldsymbol{W}_L$. This step is non restrictive: any type of training strategy can be used here, including gradient bias reduction techniques, such as Adagrad (Duchi et al., 2011), or regularization strategies Ω, for instance using Dropout (Dahl et al., 2013) or weight norm regularization. Similarly, any type of stopping criterion can be used here. The training might last for a fixed number of epochs, or can stop after using early stopping (Morgan & Bourlard, 1990). Different combinations of training strategies and stopping criterion are tested in Section 7.4.

2. **Post-training:** During this step, the first $L-1$ layers are fixed and only the last layer of the network, ϕ_L, is trained by minimizing over $\boldsymbol{W}_L$ the following problem

$$\underset{\boldsymbol{W}_L}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \tilde{\ell}\left(\boldsymbol{W}_L \Phi_{L-1}(x_i), y_i\right) + \lambda \|\boldsymbol{W}_L\|_2^2 \,, \tag{7.4}$$

where $\tilde{\ell}(x, y) := \ell(\psi_L(x), y)$. This extra learning step uses the mapping Φ_{L-1} as an embedding of the data in $\mathcal{X}_L$ and learn the best linear predictor in this space. This optimization problem takes place in a significantly lower dimensional space and since there is no need for back propagation, this step is computationally faster. To reduce the risk of overfitting with this step, a ℓ_2-regularization is added. Figure 7.1 illustrates the post-training step.

We would like to emphasize the importance of the ℓ_2-regularization used during the post-training (7.4). This regularization is added regardless of the one used in the regular training, and for all the network architectures. The extra term improves the strong convexity of the minimization problem, making post-training more efficient, and promotes the generalization of the model. The choice of the ℓ_2-regularization is motivated from the comparison with the kernel framework discussed in Section 7.3 and from our experimental results.

Remark 7.1 (Dropout)**.** *It is important to note that Dropout should not be applied on the previous layers of the network during the post-training, as it would lead to changes in the feature function* Φ_{L-1}.

7.3 Link with Kernels

In this section, we show that for the case where $\mathcal{X}_L = \mathbb{R}^{d_L}$ for some $d_L > 0$ and $\mathcal{X}_{L+1} = \mathbb{R}$, $\boldsymbol{W}_L^*$ can be approximated using kernel methods. We define the kernel k as follows,

$$\begin{aligned} k : \mathcal{X}_1 \times \mathcal{X}_1 &\mapsto \mathbb{R} \\ (x_1, x_2) &\to \left\langle \Phi_{L-1}(x_1), \Phi_{L-1}(x_2) \right\rangle \,. \end{aligned} \tag{7.5}$$

Then k is the kernel associated with the feature function Φ_{L-1}. It is easy to see that this kernel is continuous positive definite and that for $\boldsymbol{W} \in \mathbb{R}^{d_L}$, the function

$$\begin{aligned} g_{\boldsymbol{W}} : \mathcal{X}_1 &\mapsto \mathcal{X}_{L+1} \\ x &\to \left\langle \boldsymbol{W}, \Phi_{L-1}(x) \right\rangle \end{aligned} \tag{7.6}$$

belongs by construction to the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_k$ generated by k. The post-training problem (7.4) is therefore related to the problem posed in the RKHS space $\mathcal{H}_k$, defined by

$$g^* = \underset{g \in \mathcal{H}_k}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \tilde{\ell}\left(g(x_i), y_i\right) + \lambda \|g\|_{\mathcal{H}_k}^2 \ , \tag{7.7}$$

This problem is classic for the kernel methods. With mild hypothesis on $\tilde{\ell}$, the generalized representer theorem can be applied (Schölkopf et al., 2001). As a consequence, there exists $\alpha^* \in \mathbb{R}^N$ such that

$$\begin{aligned} g^* &:= \underset{g \in \mathcal{H}_k}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \tilde{\ell}\left(g(x_i), y_i\right) + \lambda \|g\|_{\mathcal{H}_k}^2 \ , \\ &= \sum_{i=1}^{N} \alpha_i^* k(X_i, \cdot) = \sum_{i=1}^{N} \left\langle \alpha_i^* \Phi_{L-1}\left(x_i\right), \Phi_{L-1}(\cdot) \right\rangle \ . \end{aligned} \tag{7.8}$$

Rewriting (7.8) with g^* of the form (7.6), we have that $g^* = g_{\boldsymbol{W}^*}$, with

$$\boldsymbol{W}^* = \sum_{i=1}^{N} \alpha_i^* \Phi_{L-1}\left(x_i\right) . \tag{7.9}$$

We emphasize that $\boldsymbol{W}^*$ is the optimal solution for the problem (7.8) and should not be confused with $\boldsymbol{W}_L^*$, the optimum of (7.4). However, the two problems differ only in their regularization, which are closely related (see the next paragraph). Thus $\boldsymbol{W}^*$ can thus be seen as an approximation of the optimal value $\boldsymbol{W}_L^*$. It is worth noting that in our experiments, $\boldsymbol{W}^*$ appears to be a good estimator of $\boldsymbol{W}_L^*$ (see Subsection 7.4.3).

Relation between $\|\cdot\|_{\mathcal{H}}$ and $\|\cdot\|_2$. The problems (7.8) and (7.4) only differ in the choice of the regularization norm. By definition of the RKHS norm, we have

$$\|g_W\|_{\mathcal{H}} = \inf \left\{ \|v\|_2 \Big/ \quad \forall x \in \mathcal{X}_1, \quad \langle v,\ \Phi_{L-1}(x) \rangle = g_W(x) \right\} \ . \tag{7.10}$$

Consequently, we have that $\|g_W\|_{\mathcal{H}} \leq \|W\|_2$, with equality when $\operatorname{Vect}(\Phi_{L-1}(\mathcal{X}_1))$ spans the entire space $\mathcal{X}_L$. In this case, the norm induced by the RKHS is equal to the ℓ_2-norm. This is generally the case, as the input space is usually in a far higher dimensional space than the embedding space, and since the neural network structure generally enforces the independence of the features. Therefore, while both norms can be used in (7.4), we chose to use the ℓ_2-norm for all our experiments as it is easier to compute than the RKHS norm.

Closed-form Solution. In the particular case where $\ell(y_1, y_2) = \|y_1 - y_2\|^2$ and $f(x) = x$, (7.8) can be reduced to a classical Kernel Ridge Regression problem. In this setting, W^* can be computed by combining (7.9) and

$$\alpha^* = \left(\Phi_{L-1}(\mathcal{D})^{\mathsf{T}} \Phi_{L-1}(\mathcal{D}) + \lambda \boldsymbol{I}_N \right)^{-1} Y \ , \tag{7.11}$$

where $\Phi_{L-1}(\mathcal{D}) = \left[\Phi_{L-1}(x_1), \dots \Phi_{L-1}(x_N) \right]$ represents the matrix of the input data $\{x_1, \dots x_N\}$ embedded in $\mathcal{X}_L$, Y is the matrix of the output data $\{y_1, \dots, y_N\}$ and $\boldsymbol{I}_N$ is the identity matrix in $\mathbb{R}^N$. This result is experimentally illustrated in Subsection 7.4.3.

Although data sets are generally too large for (7.11) to be computed in practice, it is worth noting that some kernel methods, such as Random Features (Rahimi & Recht, 2007), can be applied to compute approximations of the optimal weights during the post-training.

Multidimensional Output. Most of the previously discussed results related to kernel theory hold for multidimensional output spaces, *i.e.* $\dim(\mathcal{X}_{L+1}) = d > 1$, using multitask or operator valued kernels (Kadri et al., 2015). Hence the previous remarks can be easily extended to multidimensional outputs, encouraging the use of post-training in most settings.

7.4 Experimental Results

This section provides numerical arguments to study post-training and its influence on performances, over different data sets and network architectures. All the experiments were run using `python` and `Tensorflow`. The code to reproduce the figures is available online[1]. The results of all the experiments are discussed in depth in Section 7.5.

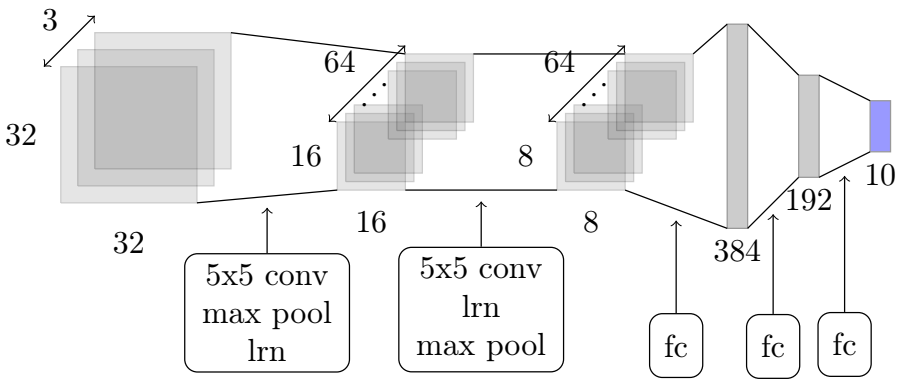


Figure 7.2: Illustration of the neural network structure used for CIFAR-10. The last layer, represented in blue, is the one trained during the post-training. The layers are composed with classical layers: 5x5 convolutional layers (5x5 conv), max pooling activation (max pool), local response normalization (lrn) and fully connected linear layers (fc).

7.4.1 Convolutional Neural Networks

The post-training method can be applied easily to feedforward convolutional neural network, used to solve a wide class of real world problems. To assert its performance, we apply it to three classic benchmark datsets: CIFAR10 (Krizhevsky, 2009), MNIST and FACES (Hinton & Salakhutdinov, 2006).

CIFAR10. This data set is composed of $60,000$ images 32×32, representing objects from 10 classes. We use the default architecture proposed by `Tensorflow` for CIFAR10 in our experiments, based on the original architecture proposed by Krizhevsky (2009). It is composed of 5 layers described in Figure 7.2. The first layers use various common tools such as local response normalization (lrn), max pooling and RELU activation. The

[1] https://github.com/tomMoral/post_training

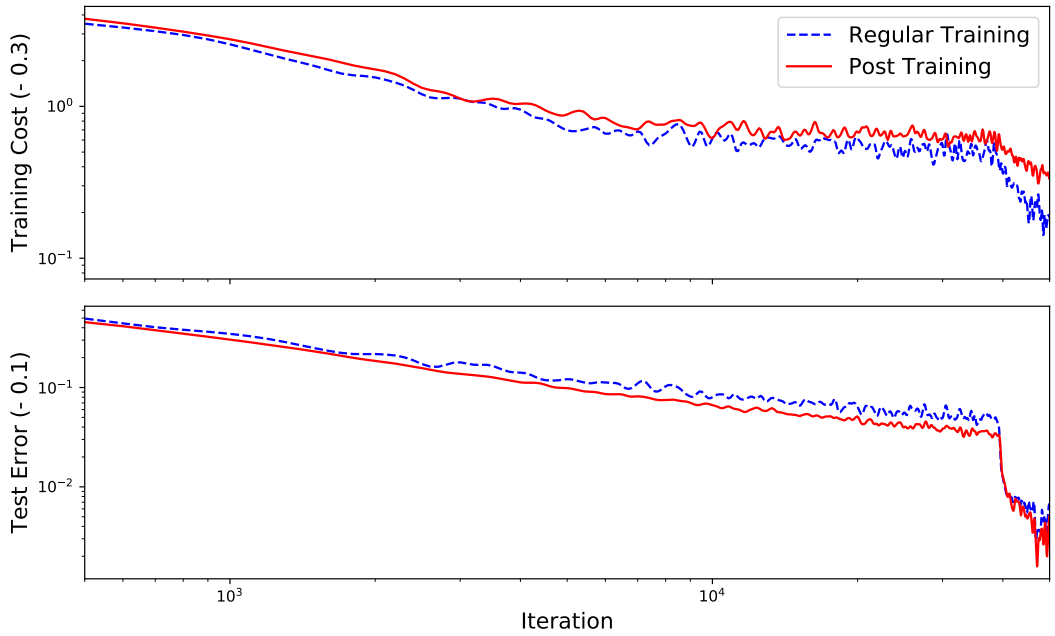


Figure 7.3: Evolution of the performances of the neural network on the CIFAR10 data set, (*dashed*) with the usual training and (*solid*) with the post-training phase. For the post-training, the value of the curve at iteration q is the error for a network trained for $q - 100$ iterations with the regular training strategy and then trained for 100 iterations with post-training. The *top* figure presents the classification error on the training set and the *bottom* figure displays the loss cost on the test set. The curves have been smoothed to increase readability.

last layer have a *softmax* activation function and the chosen training loss was the cross entropy function. The network is trained for $90k$ iterations, with batches of size 128, using stochastic gradient descent (SGD), dropout and an exponential weight decay for the learning rate. Figure 7.3 presents the performance of the network on the training and test sets for 2 different training strategies. The dashed line corresponds to classic training with SGD, with performance evaluated every 100 iterations and the solid line corresponds to the performance of the same network where the last 100 iterations are done using post-training instead of regular training. To be clearer, the value of this curve at iteration q is the error of the network, trained for $q - 100$ iterations with the regular training strategy, and then trained for 100 iterations with post-training. For the dashed line, the value of this curve at iteration q is the error of the network, trained for q iterations with the regular training strategy. The regularization parameter λ for post-training is set to 1×10^{-3}.

The results show that while the training cost of the network mildly increases due to the use of post-training, this extra step improves the generalization of the solution. The gain is smaller at the end of the training as the network converges to a local minimum, but it is consistent. Also, it is interesting to note that the post-training iterations are $4\times$ faster than the classic iterations, due to their inexpensiveness.

Table 7.1: Comparison of the performances (classification error) of different networks on different data sets, at different epochs, with or without post-training.

Data set	Network	Iterations	Mean (Std) Error in %	Mean (Std) Error with post-training in %
FACES	Small	5000	21,5 (10)	19,1 (12)
		10000	20 (4)	19 (3,5)
		20000	18 (0,9)	16,5 (0,8)
	Large	5000	25 (15)	24 (15)
		10000	15 (5)	12 (5)
		20000	11 (0,5)	10 (0,5)
MNIST	Small	1000	10.7 (1)	9.2 (1,1)
		2000	7,5 (0,7)	6,7 (0,6)
		5000	4,1 (0,2)	3,9 (0,2)
	Large	1000	9,1 (1,3)	8,5 (1,4)
		2000	4,1 (0,2)	3,5 (0,2)
		5000	1,1 (0,01)	0,9 (0,01)

Additional Data Sets. We also evaluate post-training on the MNIST data set (65000 images 27×27, with 55000 for train and 10000 for test; 10 classes) and the pre-processed FACES data set (400 images 64×64, from which 102400 sub-images, 32×32, are extracted, with 92160 for training and 10240 for testing; 40 classes). For each data set, we train three different convolutional neural networks – to assert the influence of the complexity of the network over post-training:

- a small network, with one convolutional layer (5×5 patches, 32 channels), one 2×2 max pooling layer, and one fully connected hidden layer with 512 neurons,
- a large network, with one convolutional layer (5×5 patches, 32 channels), one 2×2 max pooling layer, one convolutional layer (5×5 patches, 64 channels), one 2×2 max pooling layer and one fully connected hidden layer with 1024 neurons.

We use dropout for the regularization of the large networks, with dropout rate of 0.5, and we use $\lambda = 1 \times 10^{-2}$ for the post-training regularization. We compare the performance gain resulting of the application of post-training (100 iterations) at different epochs of each of these networks. The results are reported in Table 7.1.

As seen in Table 7.1, post-training improves the test performance of the networks with as little as 100 iterations – which is negligible compared to the time required to train the network. While the improvement varies depending on the complexity of the network, of the data set, and of the time spent training the network, it is important to remark that it always provides an improvement.

7.4.2 Recurrent Neural Network

While the kernel framework developed in Section 7.2 does not apply directly to Recurrent Neural Network, the idea of post-training can still be applied. In this experiment, we test the performances of post-training on Long Short-Term Memory-based networks (LSTM), using `PTB` data set (Marcus et al., 1993).

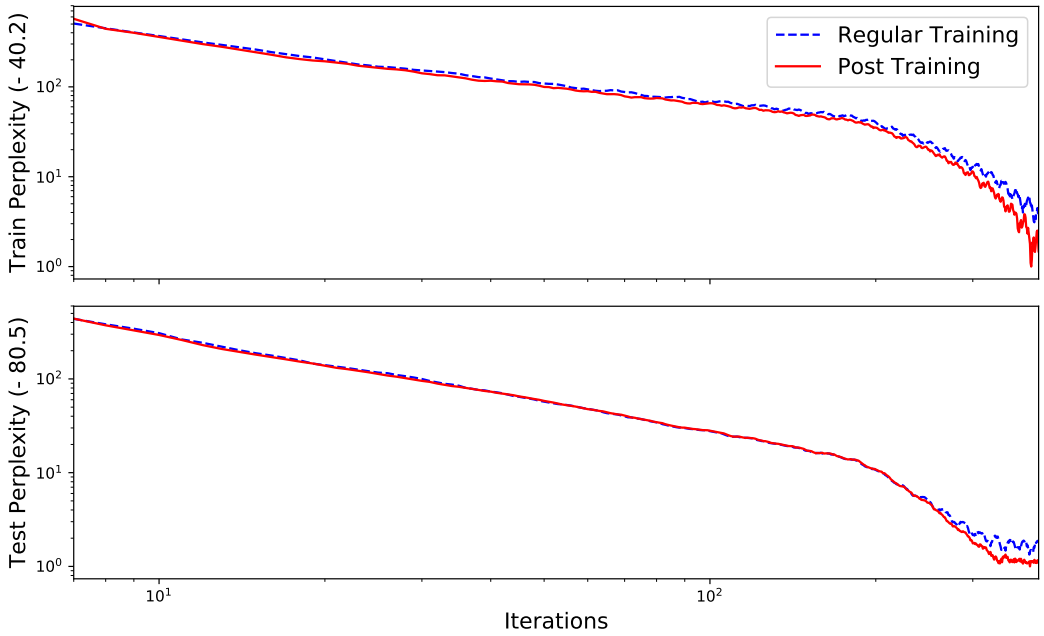


Figure 7.4: Evolution of the performances of the Recurrent network on the PTB data set. The *top* figure presents the train perplexity and the *bottom* figure displays the test perplexity. For the post-training, the value of the curve at iteration q is the error for a network trained for $q - 100$ iterations with the regular training strategy and then trained for 100 iterations with post-training.

Penn Tree Bank (PTB). This data set is composed of $929k$ training words and $82k$ test word, with a 10000 words vocabulary. We train a recurrent neural network to predict the next word given the word history. We use the architecture proposed by Zaremba et al. (2014), composed of 2 layers of 1500 LSTM units with `tanh` activation, followed by a fully connected `softmax` layer. The network is trained to minimize the average per-word perplexity for 100 epochs, with batches of size 20, using gradient descent, an exponential weight decay for the learning rate, and dropout for regularization. The performances of the network after each epoch are compared to the results obtained if the 100 last steps (i.e. 100 batches) are done using post-training. The regularization parameter for post-training, λ, is set to 1×10^{-3}. The results are reported in Figure 7.4, which presents the evolution of the training and testing perplexity.

Similarly to the previous experiments, post-training improves the test performance of the networks, even after the network has converged.

7.4.3 Optimal Last Layer for Deep Ridge Regression

In this subsection we aim to empirically evaluate the closed-form solution discussed in Section 7.2 for regression tasks. We set the activation function of the last layer to be the identity $f_L(x) = x$, and consider the loss function to be the least-squared error $\ell(x, y) = \|x - y\|_2^2$ in (7.1). In in each experiment, (7.11) and (7.9) are used to compute W^* for the kernel learned after the regular training of the neural network, which learn the embedding Φ_{L-1} and an estimate W_L . In order to illustrate this result, and to compare the performances of the weights W^* with respect to the weights W_L, learned either with usual learning strategies or with post-training, we train a neural network

Data set	Iterations	Error with classic training	Error with post-training	Error with optimal last layer
Parkinson	300	1.6	1.35	1.30
Parkinson	600	1.32	1.27	1.27
Parkinson	1000	1.27	1.27	1.27
Simulated	50	0.4	0.2	0.15
Simulated	100	0.2	0.15	0.15
Simulated	200	0.15	0.15	0.15

Table 7.2: Comparison of the performances (RMSE) of fully connected networks on different data sets, at different epochs, with or without post-training.

on two regression problems using a real and a synthetic data set. 70% of the data are used for training, and 30% for testing.

Real Data Set Regression. For this experiment, we use the Parkinson Telemonitoring data set (Tsanas et al., 2010). The input consists of $5,875$ instances of 24 dimensional data, and the output is a one dimensional real number. For this data set, a neural network made of two fully connected hidden layers of size 24 and 10, is trained for 300, 600 and $1,000$ iterations, with batches of size 50, using a ℓ_2-regularization. Then, for each network, 50 iterations of post-training are used and the performances are compared to the closed-form solutions computed using (7.11) for each saved network. The results are presented in Table 7.2.

Simulated Data Set Regression. For this experiment, we use a synthetic data set. The inputs were generated using a uniform distribution on $\big[0,1\big]^{10}$. The outputs are computed as follows:

$$Y = \tanh(XW_1)W_2$$

where $W_1 \in \big[-1,1\big]^{10\times 5}$ and $W_2 \in \big[-1,1\big]^5$ are randomly generated using a uniform law. In total, the data set is composed of $10,000$ pairs (x_i, y_j). For this data set, the same neural network as with the Parkinson Telemonitoring data set is used. Due to the simpler nature of the data, the model is trained for $50, 100$ and 200 iterations, with batches of size 50. We use the same protocol with 50 extra post-training iterations. The results are presented in Table 7.2.

For these two experiments, the post-training quickly converges to a near optimal solution, for several choices of stopping times. It is worth noting that the performances of these nearly optimal solutions are very similar to the ones obtained using the closed form solution presented in Section 7.3.

7.5 Discussion

The experiments presented in Section 7.4 show that post-training improves the performances of all the networks considered – including recurrent, convolutional and fully connected networks. There is a small gain, regardless of the time at which the regular training is stopped and the post-training is done. In both the CIFAR10 and the PTB experiment, the gap between the losses with and without post-training is more pronounced

if the training is stopped early, and tends to be smaller as the network converges to a better solution (see Figure 7.4 and Figure 7.3). The reduction of the gap between the test performances with and without post-training is made clear in Table 7.1. For the FACES data set, with a large-size convolutional neural network, while the error rate drops by nearly 4% when post-training is applied after 5000 iterations, this same error rate only drops by 0.4% when it is applied after 20000 iterations. This same observation can be done for the other results reported in Table 7.1. However, while the improvement is larger when the network did not fully converge prior to the post-training, it still seems to improve the performance when the network has reached its minimum: for example in PTB the final test perplexity is 81.7 with post-training and 82.4 without; in CIFAR10 the errors are respectively 0.147 and 0.154. If the networks are allowed to reach a local minima, for instance by training them with a regular algorithm for a very large number of iterations, the advantage provided by post-training vanishes. For example in PTB, the test perplexity after 2000 iterations (instead of 400) is 83.2 regardless of post-training. This is coherent with the intuition behind the post-training: once the training reaches a local minima, the last layer weights are already optimal for the learned representation and additional optimization steps will not improve the performance. But the local minima are usually avoided as they tend to result in networks which overfit and the features learned become less appropriate to the general problem. This is for instance the case when early stopping is used as a stopping criterion for the training.

It is important to note that the post-training computational cost is very low compared to the full training computations. For instance, in the CIFAR10 experiment, each iteration for post-training is $4\times$ faster on the same GPU than an iteration using the full gradient. Also, in the different experiments, post-training produces a performance gap after using as little as 100 batches. There are multiple reasons behind this efficiency: first, the system reaches a local minimum relatively rapidly for post-training as the problem (7.4) has a small number of parameters compared to the dimensionality of the original optimization problem. Second, the iterations used for the resolution of (7.4) are computationally cheaper, as there is no need to chain high dimensional linear operations, contrarily to regular backpropagation used during the training phase. Finally, since the post-training optimization problem is generally convex, the optimization is guaranteed to converge rapidly to the optimal weights for the last layer.

Another interesting point is that there is no evidence that the post-training step leads to overfitting. In CIFAR10, the test error is improved by the use of post-training, although the training loss is similar. The other experiments do not show signs of overfitting either as the test error is mostly improved by our additional step. This stems from the fact that the post-training optimization is much simpler than the original problem as it lies in a small-dimensional space – which, combined with the added ℓ_2-regularization, efficiently prevents overfitting. The regularization parameter λ plays an important role in post-training. Setting λ to be very large reduces the explanatory capacity of the classifiers whereas if λ is too small, the capacity can become too large and lead to overfitting. Overall, our experiments highlighted that the post-training produces significant results for any choice of λ reasonably small (i.e $10^{-5} \leq \lambda \leq 10^{-2}$). This parameter is linked to the regularization parameter of the kernel methods, as stated in Section 7.3.

Overall, these results show that the post-training step can be applied to most trained networks, without prerequisites about how optimized they are since post-training does not degrade their performances, providing a consistent gain in performances for a very

low additional computational cost.

In Subsection 7.4.3, numerical experiments highlight the link between post-training and kernel methods. As illustrated in Table 7.2, using the optimal weights derived from kernel theory immediately gives minimal errors for the considered network once the first layers weights are sufficiently trained. After 50 iterations for the simulated data and 600 for the Parkinson data set, the optimal weights for the kernel embedding reach the minimal test error we were able to achieve with the full training. However, computing the optimal weights for the last layer is only achievable for small data set due to the required matrix inversion. Moreover, the closed form solution is known only for specific problems, such as kernelized least square regression. But post-training approaches the same performance in these cases solving (7.4) with gradient-based methods.

While these preliminary results show that post-training might be of practical use, there is no clear evidence of its performances. We were not able to show the significance of the performance boost it provides as the performances of our networks on classical datasets do not reach state of the art results and experiments on modern network architectures would also be required to confirm its usefulness. Finally, it remains to be proven that post-training reaches a local minima with improved performances compare to regular training which might overfit. The link to kernel methods proposed in Section 7.3 only suggests this procedure provides good parameter for the last layer but it is not sufficient to guarantee that the last layer weight computed with post-training will be optimal.

7.6 Proofs

We show here, for the sake of completeness, that the post-training problem is convex for the softmax activation in the last layer and the cross entropy loss. This result is shown using classic arguments to show the convexity of a function.

Proposition 7.2 (convexity)**.** *$\forall N, M \in \mathbb{N}$, $\forall X \in \mathbb{R}^N$, $\forall j \in \big[1, M\big]$, the following function F is convex:*

$$\begin{aligned} F :& \mathbb{R}^{N\times M} \mapsto \mathbb{R} \\ & W \to \log\left(\sum_{i=1}^{M} \exp(XW_i)\right) - \sum_{i=1}^{M} \delta_{ij} \log\Big(\exp(XW_i)\Big). \end{aligned}$$

where δ is the Dirac function, and W_i denotes the i-th row of a W.

Proof 1. Let

$$P_i(W) = \frac{\exp(XW_i)}{\sum_{j=1}^{M} \exp(XW_j)}.$$

then

$$\frac{\partial P_i}{\partial W_{m,n}} = \begin{cases} -x_n P_i(W) P_m(W) & \text{if } i \neq m \\ -x_n P_m^2(W) + x_n P_m(W) & \text{otherwise} \end{cases}$$

Noting that

$$F(W) = -\sum_{i=1}^{M} \delta_{ij} \log\Big(P_i(W)\Big),$$

we have

$$\begin{aligned} \frac{\partial F(W)}{\partial W_{m,n}} &= -\sum_{i=1}^{M} \delta_{ij} \frac{1}{P_i(W)} \frac{\partial P_i}{\partial W_{m,n}} \\ &= x_n \left(\sum_{i=1}^{M} \delta_{ij} P_m(W) - \delta_{mj}\right) \\ &= x_n \Big(P_m(W) - \delta_{mj}\Big), \end{aligned}$$

hence

$$\begin{aligned} \frac{\partial^2 F(W)}{\partial W_{m,n} \partial W_{p,q}} &= x_n \left(\frac{\partial P_m}{\partial W_{p,q}}\right), \\ &= x_n x_q P_m(W) \Big(\delta_{m,p} - P_p(W)\Big). \end{aligned}$$

Hence the following identity

$$H(F) = \mathbf{P}(W) \otimes (XX^{\mathsf{T}})$$

where $\otimes$ is the Kronecker product, and the matrix $\mathbf{P}(W)$ is defined by $\mathbf{P}_{m,p} = P_m(W)\Big(\delta_{m,p} - P_p(W)\Big)$. Now since $\forall 1 \leq m \leq M$,

$$\begin{aligned}\sum_{p=1,p\neq m}^{M}\left|\mathbf{P}_{m,p}\right| &= P_m(W)\sum_{p=1,p\neq m}^{M}P_p(W)\\ &= P_m(W)\Big(1-P_m(W)\Big)\\ &= \mathbf{P}_{m,m}\end{aligned}$$

$\mathbf{P}(W)$ is thus a diagonally dominant matrix. Its diagonal elements are positive

$$\mathbf{P}_{m,m} = P_m(W)\Big(1-P_m(W)\Big) \geq 0, \qquad \text{as } P_m(W)\in[0,1]$$

and thus $\mathbf{P}(W)$ is positive semidefinite. Since XX^{T} is positive semidefinite too, their Kronecker product is also positive semidefinite, hence the conclusion.

□

8

Understanding Trainable Sparse Coding with Matrix Factorization

"If knowledge can create problems, it is not through ignorance that we can solve them..."

– Isaac Asimov

Contents

Recent work has revisited traditional optimization algorithms for sparse coding in light of the recent literature in deep learning. In particular, recent work shows that one can design trainable networks that provide accelerated solutions to the optimization problem. But the reasons for such acceleration remain unclear.

This chapter intends to provide elements that explain why the acceleration is possible in the case of ISTA. We show that ISTA can be solved faster when the design matrix admits a quasi-diagonal factorization with sparse eigenspaces. The resulting algorithm

has the same convergence rate with constant factors potentially improved. Then, we derive the conditions under which the constant factors of the adaptive algorithm are better than the ones of ISTA in expectation over generic Gaussian dictionary. Finally, we design neural networks which are able to compute this algorithm and show that they are a re-parametrization of LISTA, and thus performance of LISTA are at least as good as this algorithm. We conclude by designing adverse examples for our factorization based algorithm and show that LISTA also fails to accelerate on these cases.

As stated in Subsection 6.3.3, understanding the properties of optimization algorithms adapted through neural networks highlights the link between deep learning and sparse representations. Indeed, the architectures used in LISTA networks are very close to those used for feedforward dense neural networks. The properties of the LISTA network can thus be used to link the dense networks to sparse representations, in order to make their internal representations more interpretable. Moreover, the LISTA architecture can be modified to solve the convolutional sparse coding problem by replacing the fully connected layers by convolutional layers. Properties of these networks shed light on the link between convolutional neural networks and convolutional sparse representations. We focus on the non convolutional case in this chapter as it is easier to analyze. Moreover, the results can also be applied to the vector form of the convolutional LASSO (*cf* Subsection 3.1.3). Extending these results in the particular case of band circulant matrices is kept for future work.

8.1 Learning to Optimize

Feature selection is essential for high dimensional data analysis. Different techniques have been developed to tackle this problem efficiently, and among them sparsity has emerged as a leading paradigm. In statistics, the LASSO estimator (Tibshirani, 1996) provides a reliable way to select features and has been extensively studied in the last two decades (Hastie et al. 2015 and references therein). In machine learning and signal processing, sparse coding has made its way into several modern architectures, including large scale computer vision (Coates & Ng, 2011) and biologically inspired models (Cadieu & Olshausen, 2012). Also, dictionary learning is a generic unsupervised learning method to perform nonlinear dimensionality reduction with efficient computational complexity (Mairal et al., 2010). All these techniques heavily rely on the resolution of ℓ_1-regularized least squares.

The ℓ_1-sparse coding problem is defined as solving, for a given input $x \in \mathbb{R}^p$ and dictionary $D \in \mathbb{R}^{p \times K}$, the following problem:

$$z^*(x) = \underset{z \in \mathbb{R}^K}{\operatorname{argmin}} F_x(z) \triangleq \frac{1}{2}\|x - Dz\|^2 + \lambda \|z\|_1 \ . \tag{8.1}$$

This problem is very close from the convolutional sparse coding problem presented in Section 3.1. The main difference is that the sum of convolutions is here replaced by a matrix multiplication. As stated in Subsection 3.1.3, the convolutional problem can be rewritten in a vector form with a band circulant matrix D. Problem (8.1) is convex and can therefore be solved using convex optimization machinery. Proximal splitting methods (Beck & Teboulle, 2009) alternate between the minimization of the smooth and differentiable part using the gradient information and the minimization of the non-differentiable part using a proximal operator (Combettes & Bauschke, 2011). These methods can also be accelerated by considering a momentum term, as it is done

in FISTA (Beck & Teboulle, 2009; Nesterov, 2005). Coordinate descent (Friedman et al., 2007; Osher & Li, 2009) leverages the closed formula that can be derived for optimizing the problem (8.1) for one coordinate z_i given that all the others are fixed. At each step of the algorithm, one coordinate is updated to its optimal value, which yields an inexpensive scheme to perform each step. The choice of the coordinate to update at each step is critical for the performance of the optimization procedure. Least Angle Regression (LARS) (Hesterberg et al., 2008) is another method that computes the whole LASSO regularization path. These algorithms all provide an optimization procedure that leverages the local properties of the cost function iteratively. They can be shown to be optimal among the class of first-order methods for generic convex, non-smooth functions (Bubeck, 2015).

But all these results are given in the worst case and do not use the distribution of the considered problem. One can thus wonder whether a more efficient algorithm to solve (8.1) exists for a fixed dictionary D and generic input x drawn from a certain input data distribution. In Gregor & Lecun (2010), the authors introduced LISTA, a trained version of ISTA that adapts the parameters of the proximal splitting algorithm to approximate the solution of the LASSO using a finite number of steps. This method exploits the common structure of the problem to learn a better transform than the generic ISTA step. As ISTA is composed of a succession of linear operations and piece-wise non linearities, the authors use the neural network framework and the back-propagation to derive an efficient procedure solving the LASSO problem. In Sprechmann et al. (2012), the authors extended LISTA to more generic sparse coding scenarios and showed that adaptive acceleration is possible under general input distributions and sparsity conditions.

In this chapter, we are interested in the following question: Given a finite computational budget, what is the optimum estimator of the sparse coding? This question belongs to the general topic of computational tradeoffs in statistical inference. Randomized sketches (Alaoui & Mahoney, 2015; Yang et al., 2015) reduce the size of convex problems by projecting expensive kernel operators into random subspaces, and reveal a tradeoff between computational efficiency and statistical accuracy. Agarwal (2012) provides several theoretical results on performing inference under various computational constraints, and Chandrasekaran & Jordan (2013) considers a hierarchy of convex relaxations that provide practical tradeoffs between accuracy and computational cost. More recently, Oymak et al. (2015) provides sharp time-data tradeoffs in the context of linear inverse problems, showing the existence of a phase transition between the number of measurements and the convergence rate of the resulting recovery optimization algorithm. Giryes et al. (2016a) builds on this result to produce an analysis of LISTA that describes acceleration in conditions where the iterative procedure has linear convergence rate. Finally, Xin et al. (2016) also studies the capabilities of Deep Neural networks at approximating sparse inference. The authors show that unrolled iterations lead to better approximation if the weights are allowed to vary at each layer, contrary to standard splitting algorithms. Whereas their focus is on relaxing the convergence hypothesis of iterative thresholding algorithms, we study a complementary question, namely when is speedup possible, without assuming strongly convex optimization. Their results are consistent with ours, since our analysis also shows that learning shared layer weights is less effective.

Inspired by the LISTA architecture, our mathematical analysis reveals that adaptive acceleration is related to a specific matrix factorization of the Gram matrix of the dictionary $B = D^\mathsf{T} D$ as $B = A^\mathsf{T} S A - R$,where A is unitary, S is diagonal and the residual is positive semidefinite: $R \succeq 0$. Our factorization balances between near diagonalization by asking that $\|R\|$ be small and that small perturbation of the ℓ_1 norm, *i.e.* $\|Az\|_1 - \|z\|_1$ be small. When this factorization succeeds, we prove that the resulting splitting algorithm enjoys a convergence rate with improved constants with respect to the non-adaptive version. Moreover, our analysis also shows that acceleration is mostly possible at the beginning of the iterative process, when the current estimate is far from the optimal solution, which is consistent with numerical experiments. We also show that the existence of this factorization is not only sufficient for acceleration, but also necessary. This is shown by constructing dictionaries whose Gram matrix diagonalizes in a basis that is incoherent with the canonical basis, and verifying that LISTA fails to accelerate with respect to ISTA in that case.

In our numerical experiments, we design a specialized version of LISTA called FacNet, with more constrained parameters, which is then used as a tool to show that our theoretical analysis captures the acceleration mechanism of LISTA. Our theoretical results can be applied to FacNet and as LISTA is a generalization of this model, it always performs at least as well, showing that the existence of the factorization is a sufficient certificate for acceleration by LISTA. Reciprocally, we show that for cases where no acceleration is possible with FacNet, the LISTA model also fails to provide acceleration, linking the two speedup mechanisms. This numerical evidence suggests that the existence of our proposed factorization is sufficient and somewhat necessary for LISTA to show good results.

The rest of this chapter is structured as follows. Section 8.2 presents our mathematical analysis and proves the convergence of the adaptive algorithm as a function of the quality of the matrix factorization. In Section 8.3, we highlight under which the conditions on the problem design, it is possible to accelerate the resolution of the LASSO with our algorithm, in expectation over generic dictionaries, drawn uniformly on the ℓ_2 unit sphere. Finally, Section 8.4 describe the generic architectures that will enable the usage of such schemes and Section 8.5 present the numerical experiments, which validate our analysis over a range of different scenarios.

8.2 Accelerating Sparse Coding with Sparse Matrix Factorization

8.2.1 Unitary Proximal Splitting

In this section we describe our setup for accelerating sparse coding based on the Proximal Splitting method. Let $\Omega \subset \mathbb{R}^p$ be the set describing our input data, and $D \in \mathbb{R}^{p \times K}$ be a dictionary, with $K > p$. We wish to find fast and accurate approximations of the sparse coding $z^*(x)$ of any $x \in \Omega$, defined in (8.1). For simplicity, we denote $B = D^\mathsf{T} D$ and $y = D^\dagger x$ to rewrite (8.1) as

$$z^*(x) = \arg\min_z F_x(z) = \underbrace{\frac{1}{2}(y-z)^\mathsf{T} B(y-z)}_{E(z)} + \underbrace{\lambda \|z\|_1}_{G(z)} \ . \tag{8.2}$$

For clarity, we will refer to F_x as F and to $z^*(x)$ as z^* when there is no ambiguity. The classic proximal splitting technique finds z^* as the limit of sequence $(z^{(q)})_q$, obtained by successively constructing a surrogate loss $F_q(z)$ of the form

$$F_q(z) = E(z^{(q)}) + (z^{(q)} - y)^\mathsf{T} B(z - z^{(q)}) + L_q\|z - z^{(q)}\|_2^2 + \lambda\|z\|_1 \ , \tag{8.3}$$

satisfying $F_q(z) \geq F(z)$ for all $z \in \mathbb{R}^K$. Since F_q is separable in each coordinate of z, $z^{(q+1)} = \operatorname{argmin}_z F_q(z)$ can be computed efficiently. This scheme is based on a majorization of the quadratic form $(y-z)^\mathsf{T} B(y-z)$ with an isotropic quadratic form $L_q\|z^{(q)} - z\|_2^2$. The convergence rate of the splitting algorithm is optimized by choosing L_q as the smallest constant satisfying $F_q(z) \geq F(z)$, which corresponds to the largest singular value of B.

The computation of $z^{(q+1)}$ remains separable by replacing the quadratic form $L_q \boldsymbol{I}_K$ by any diagonal form. However, the Gram matrix $B = D^\mathsf{T} D$ might be poorly approximated via diagonal forms for general dictionaries. Our objective is to accelerate the convergence of this algorithm by finding appropriate factorizations of the matrix B such that

$$B \approx A^\mathsf{T} S A \ , \quad \text{and} \quad \|Az\|_1 \approx \|z\|_1 \ ,$$

where A is unitary and S is diagonal positive definite. Given a point $z^{(q)}$ at iteration q, we can rewrite $F(z)$ as

$$F(z) = E(z^{(q)}) + (z^{(q)} - y)^\mathsf{T} B(z - z^{(q)}) + Q_B(z, z^{(q)}) \ , \tag{8.4}$$

with $Q_B(v,w) := \dfrac{1}{2}(v-w)^\mathsf{T} B(v-w) + \lambda\|v\|_1$. For any diagonal positive definite matrix S and unitary matrix A, the surrogate loss

$$\widetilde{F}(z, z^{(q)}) := E(z^{(q)}) + (z^{(q)} - y)^\mathsf{T} B(z - z^{(q)}) + Q_S(Az, Az^{(q)})$$

can be explicitly minimized, since

$$\begin{aligned} \operatorname*{argmin}_z \widetilde{F}(z, z^{(q)}) &= A^\mathsf{T} \operatorname*{argmin}_u \Big((z^{(q)} - y)^\mathsf{T} B A^\mathsf{T} (u - Az^{(q)}) + Q_S(u, Az^{(q)}) \Big) \\ &= A^\mathsf{T} \operatorname*{argmin}_u Q_S \Big(u, Az^{(q)} - S^{-1} AB(z^{(q)} - y) \Big) \end{aligned} \tag{8.5}$$

where we use the variable change $u = Az$. As S is diagonal positive definite, (8.5) is separable and can be computed easily, using a linear operation followed by a point-wise non-linear soft-thresholding. Thus, any couple (A, S) ensures a computationally cheap scheme. The question is then how to factorize B using S and A in an optimal manner, that is, such that the resulting proximal splitting sequence converges as fast as possible to the sparse coding solution.

8.2.2 Non-asymptotic Analysis

We will now establish convergence results based on the previous factorization. These bounds will inform us on how to best choose the factors A_q and S_q in each iteration.

For that purpose, let us define

$$\delta_A(z) = \lambda \Big(\|Az\|_1 - \|z\|_1 \Big) \ , \text{ and } \ R = A^\mathsf{T} S A - B \ . \tag{8.6}$$

The quantity $\delta_A(z)$ thus measures how invariant the ℓ_1 norm is to the unitary operator A, whereas R corresponds to the residual of approximating the original Gram matrix B by our factorization $A^\mathsf{T}SA$. Given a current estimate $z^{(q)}$, we can rewrite

$$\widetilde{F}(z, z^{(q)}) = F(z) + \frac{1}{2}(z - z^{(q)})^\mathsf{T} R(z - z^{(q)}) + \delta_A(z) \ . \tag{8.7}$$

By imposing that R is a positive semidefinite residual, one immediately obtains the following bound.

Proposition 8.1. *Suppose that $R = A^\mathsf{T}SA - B$ is positive definite, and define*

$$z^{(q+1)} = \operatorname*{argmin}_z \widetilde{F}(z, z^{(q)}) \ . \tag{8.8}$$

$$\textit{Then} \quad F(z^{(q+1)}) - F(z^*) \leq \frac{1}{2}\|R\|\|z^{(q)} - z^*\|_2^2 + \delta_A\left(z^*\right) - \delta_A\left(z^{(q+1)}\right) \ . \tag{8.9}$$

Proof. By definition of $z^{(q+1)}$ and using the fact that $R \succ 0$ we have

$$\begin{aligned} F(z^{(q+1)}) - F(z^*) &\leq F(z^{(q+1)}) - \widetilde{F}(z^{(q+1)}, z^{(q)}) + \widetilde{F}(z^*, z^{(q)}) - F(z^*) \\ &= -\frac{1}{2}(z^{(q+1)} - z^{(q)})^\mathsf{T} R(z^{(q+1)} - z^{(q)}) - \delta_A(z^{(q+1)}) \\ &\qquad + \frac{1}{2}(z^* - z^{(q)})^\mathsf{T} R(z^* - z^{(q)}) + \delta_A(z^*) \\ &\leq \frac{1}{2}(z^* - z^{(q)})^\mathsf{T} R(z^* - z^{(q)}) + \left(\delta_A(z^*) - \delta_A(z^{(q+1)})\right) \ . \end{aligned}$$

where the first line results from the definition of $z^{(q+1)}$ and the third line makes use of R positiveness. □

This simple bound reveals that to obtain fast approximations to the sparse coding it is sufficient to find S and A such that $\|R\|$ is small and that the ℓ_1 commutation term δ_A is small. These two conditions will be often in tension: one can always obtain $R \equiv 0$ by using the Singular Value Decomposition of $B = A_0^\mathsf{T} S_0 A_0$ and setting $A = A_0$ and $S = S_0$. However, the resulting A_0 might introduce large commutation error δ_{A_0}. Similarly, as the absolute value is non-expansive, *i.e.* $\Big||a| - |b|\Big| \leq |a - b|$, we have that

$$\begin{aligned} |\delta_A(z)| = \lambda\Big|\|Az\|_1 - \|z\|_1\Big| &\leq \lambda\|(A - \boldsymbol{I}_K)z\|_1 \\ &\leq \lambda\sqrt{2\max(\|Az\|_0, \|z\|_0)} \ \cdot \|A - \boldsymbol{I}_K\| \ \cdot \|z\|_2 \ , \end{aligned} \tag{8.10}$$

where we have used the Cauchy-Schwartz inequality $\|x\|_1 \leq \sqrt{\|x\|_0}\|x\|_2$ in the last equation. In particular, (8.10) shows that unitary matrices in the neighborhood of $\boldsymbol{I}_K$ with $\|A - \boldsymbol{I}_K\|$ small have small ℓ_1 commutation error δ_A but can be inappropriate to approximate general B matrix.

The commutation error also depends upon the sparsity of z and Az . If both z and Az are sparse then the commutation error is reduced, which can be achieved if A is itself a sparse unitary matrix. Moreover, since

$$|\delta_A(z) - \delta_A(z')| \leq \lambda|\|z\|_1 - \|z'\|_1| + \lambda|\|Az\|_1 - \|Az'\|_1|$$

$$\text{and} \quad |\|z\|_1 - \|z'\|_1| \leq \|z - z'\|_1 \leq \sqrt{\|z - z'\|_0}\|z - z'\|_2 \ ,$$

it results that δ_A is Lipschitz with respect to the Euclidean norm; let us denote by $L_A(z)$ its local Lipschitz constant in z, which can be computed using the norm of the sub-gradient in z[1]. A uniform upper bound for this constant is $(1 + \|A\|_1)\lambda\sqrt{m}$, but it is typically much smaller when z and Az are both sparse.
Equation (8.8) defines an iterative procedure determined by the pairs $\{(A_q, S_q)\}_q$. The following theorem uses the previous results to compute an upper bound of the resulting sparse coding estimator.

Theorem 8.2. *Let A_q, S_q be the pair of unitary and diagonal matrices corresponding to iteration q, chosen such that $R_q = A_q^\mathsf{T} S_q A_q - B \succ 0$. It results that*

$$F(z^{(q)}) - F(z^*) \leq \frac{(z^* - z^{(0)})^\mathsf{T} R_0 (z^* - z^{(0)}) + 2L_{A_0}(z^{(1)})\|z^* - z_1\|_2}{2q} + \frac{\alpha - \beta}{2q} \;, \tag{8.11}$$

$$\begin{aligned} \textit{with} \quad \alpha &= \sum_{i=1}^{q-1} \left(2L_{A_i}(z^{(i+1)})\|z^* - z^{(i+1)}\|_2 + (z^* - z^{(i)})^\mathsf{T}(R_{i-1} - R_i)(z^* - z^{(i)}) \right) \;, \\ \beta &= \sum_{i=0}^{q-1} (i+1) \left((z^{(i+1)} - z^{(i)})^\mathsf{T} R_i (z^{(i+1)} - z^{(i)}) + 2\delta_{A_i}(z^{(i+1)}) - 2\delta_{A_i}(z^{(i)}) \right) \;, \end{aligned}$$

where $L_A(z)$ denotes the local lipschitz constant of δ_A at z.

Remark. If one sets $A_q = \boldsymbol{I}_K$ and $S_q = \|B\|\boldsymbol{I}_K$ for all $q \geq 0$, (8.11) corresponds to the bound of the ISTA algorithm (Beck & Teboulle, 2009).

The proof is deferred to Subsection 8.7.1. We can specialize the theorem in the case when A_0, S_0 are chosen to minimize the bound (8.9) and $A_q = \boldsymbol{I}_K$, $S_q = \|B\|\boldsymbol{I}_K$ for $q \geq 1$.

Corollary 8.3. *If $A_q = \boldsymbol{I}_K$, $S_q = \|B\|\boldsymbol{I}_K$ for $q \geq 1$ then*

$$\begin{aligned} F(z^{(q)}) - F(z^*) \leq & \frac{(z^* - z^{(0)})^\mathsf{T} R_0 (z^* - z^{(0)}) + (z^* - z^{(1)})^\mathsf{T} R_0 (z^* - z^{(1)})^\mathsf{T}}{2q} \\ & + \frac{L_{A_0}(z_1)(\|z^* - z^{(1)}\| + \|z^{(1)} - z^{(0)}\|)}{q} \;. \end{aligned} \tag{8.12}$$

This corollary shows that by simply replacing the first step of ISTA by the modified proximal step detailed in (8.5), one can obtain an improved bound at fixed q as soon as

$$\begin{aligned} & 2\|R_0\| \max(\|z^* - z^{(0)}\|_2^2, \|z^* - z^{(1)}\|_2^2) \\ & \quad + 4L_{A_0}(z^{(1)}) \max(\|z^* - z^{(0)}\|_2, \|z^* - z^{(1)}\|_2) \quad \leq \quad \|B\|\|z^* - z^{(0)}\|_2^2 \;, \end{aligned}$$

which, assuming $\|z^* - z^{(0)}\|_2 \geq \|z^* - z^{(1)}\|_2$, translates into

$$\|R_0\| + 2\frac{L_{A_0}\left(z^{(1)}\right)}{\|z^* - z^{(0)}\|_2} \leq \frac{\|B\|}{2} \;. \tag{8.13}$$

More generally, given a current estimate $z^{(q)}$, searching for a factorization (A_q, S_q) will improve the upper bound when

$$\|R_q\| + 2\frac{L_{A_q}\left(z^{(q+1)}\right)}{\|z^* - z^{(q)}\|_2} \leq \frac{\|B\|}{2} \;. \tag{8.14}$$

[1]This quantity exists as δ_A is a difference of convex. See proof of Proposition 8.7.1 in proofs section for details.

We emphasize that this is not a guarantee of acceleration, since it is based on improving an upper bound. However, it provides a simple picture on the mechanism that makes non-asymptotic acceleration possible.

8.2.3 Interpretation

In this section we analyze the consequences of Theorem 8.2 in the design of fast sparse coding approximations, and provide a possible explanation for the behavior observed numerically.

"Phase Transition" and Law of Diminishing Returns

(8.14) reveals that the optimum matrix factorization in terms of minimizing the upper bound depends upon the current scale of the problem, that is, of the distance $\|z^* - z^{(q)}\|$. At the beginning of the optimization, when $\|z^* - z^{(q)}\|$ is large, the bound (8.14) makes it easier to explore the space of factorizations (A, S) with A further away from the identity. Indeed, the bound tolerates larger increases in $L_A(z^{(q+1)})$, which is dominated by

$$L_A(z^{(q+1)}) \leq \lambda(\sqrt{\|z^{(q+1)}\|_0} + \sqrt{\|Az^{(q+1)}\|_0}) \ ,$$

i.e. the sparsity of both $z^{(1)}$ and $A_0(z^{(1)})$. On the other hand, when we reach intermediate solutions $z^{(q)}$ such that $\|z^* - z^{(q)}\|$ is small with respect to $L_A(z^{(q+1)})$, the upper bound is minimized by choosing factorizations where A is getting closer and closer to the identity, leading to the non-adaptive regime of standard ISTA ($A = Id$).

This is consistent with the numerical experiments, which show that the gains provided by learned sparse coding methods are mostly concentrated in the first iterations. Once the estimates reach a certain energy level, section 8.5 shows that LISTA enters a steady state in which the convergence rate matches that of standard ISTA.

The natural follow-up question is to determine how many layers of adaptive splitting are sufficient before entering the steady regime of convergence. A conservative estimate of this quantity would require an upper bound of $\|z^* - z^{(q)}\|$ from the energy bound $F(z^{(q)}) - F(z^*)$. Since in general F is convex but not strongly convex, such bound does not exist unless one can assume that F is locally strongly convex (for instance for sufficiently small values of F).

Improving the Factorization to Particular Input Distributions

Given an input data set $\mathcal{D} = (x_i, z_i^{(0)}, z_i^*)_{i \leq N}$, containing examples $x_i \in \mathbb{R}^n$, initial estimates $z_i^{(0)}$ and sparse coding solutions z_i^*, the factorization adapted to $\mathcal{D}$ is defined as

$$\min_{A,S;\ A^\mathsf{T}A = \boldsymbol{I}_K, A^\mathsf{T}SA - B \succ 0} \frac{1}{N} \sum_{i \leq N} \frac{1}{2} (z_i^{(0)} - z_i^*)^\mathsf{T} (A^\mathsf{T}SA - B)(z_i^{(0)} - z_i^*) + \delta_A(z_i^*) - \delta_A(z_{1,i}) \ . \tag{8.15}$$

Therefore, adapting the factorization to a particular data set, as opposed to enforcing it uniformly over a given ball $B(z^*; R)$ (where the radius R ensures that the initial value $z^{(0)} \in B(z^*; R)$), will always improve the upper bound (8.9). Studying the gains resulting from the adaptation to the input distribution is kept for future work.

8.3 Generic Gap Control

In this section, we consider the problem of accelerating the resolution of (8.1) in the case where D is a generic dictionary, *i.e.* its elements D_i are draw uniformly over the ℓ_2 unit-sphere.

Definition 8.4 (Generic dictionary)**.** *A dictionary $D \in \mathbb{R}^{p\times K}$ is a generic dictionary when its columns D_i are drawn uniformly over the ℓ_2 unit sphere $\mathcal{S}^{p-1}$.*

The results by Song & Gupta (1997) show that such dictionaries emerge when the atoms are drawn independently from normal distributions $\mathcal{N}(0, \boldsymbol{I}_p)$ and then normalized on the unit sphere. Thus, $D_i = \frac{d_i}{\|d_i\|_2}$ with $d_i \sim \mathcal{N}(0, \boldsymbol{I}_p)$ for all $i \in [\![1, K]\!]$. In this context, we consider the matrices A which are perturbations of the identity and highlight the conditions under which it is possible to find a perturbation of the identity A which is more advantageous than the identity to resolve (8.1). For a fixed integer $i \in [\![1, K]\!]$, e_i denotes the i-th canonical direction and we introduce $\mathcal{E}_{\delta,i}$, the ensemble such that

$$\mathcal{E}_{\delta,i} = \left\{ u \in \mathbb{R}^K : \exists \mu < \delta,\ \exists h_i \in Span(e_i)^\perp \cap \mathcal{S}^{K-1} \quad s.t \quad u = \sqrt{1-\mu^2} e_i + \mu h_i \right\} ,$$

This ensemble contains the vectors which are mainly supported by one of the canonical directions. Indeed, $\cup_{i=1}^K \mathcal{E}_{\delta,i} = \left\{ u \in \mathbb{R}^K : \|u\|_2 = 1, \|u\|_\infty > \sqrt{1-\delta^2} \right\}$ We will denote $A \subset \mathcal{E}_\delta$ when a matrix A is such that each of its columns A_i are in $\mathcal{E}_{\delta,i}$. These matrices are diagonally dominant and are close to the identity when δ is close to 0, as $\|A - I\|_F = K\delta$.

8.3.1 Controlling the Sparse Diagonalization for the Gram Matrix

First, we analyze the possible gain of replacing B by an approximate diagonalization $A^{-1}SA$ for a diagonally dominant matrix $A \subset \mathcal{E}_\delta$. We choose to study the case where S is chosen deterministicaly when A is fixed. For A, B fixed, we choose the matrix S which minimizes the frobenius norm of the diagonalization error, *i.e.*

$$S = \underset{S' diagonal}{\operatorname{argmax}} \|B - A^\mathsf{T} S' A\|_F \tag{8.16}$$

This matrix S can easily be computed as $S_{i,i} = A_i^\mathsf{T} B A_i$.

Lemma 8.5. *For a generic dictionary D and a diagonally dominant matrix $A \subset \mathcal{E}_\delta$,*

$$\begin{aligned}\mathbb{E}_D \left[\min_{A_i \in \mathcal{E}_{\delta,i}} \left\| A^{-1}SA - B \right\|_F^2 \right] \leq & \frac{K(K-1)}{p} - 4\delta(K-1)\sqrt{\frac{K}{p}} \\ & + \delta^2 \left(8\mathbb{E}_D \left[\|B\|_F^4 \right] - 6\frac{K(K-1)}{p} \right) + \underset{\delta\to 0}{\mathcal{O}} \left(\delta^3 \right) .\end{aligned}$$

Proof. **sketch for Lemma 8.5.** (The full proof can be found Subsubsection 8.7.2.2)
Using the properties of the matrix $A \subset \mathcal{E}_\delta$ we can show that

$$\left\| A^{-1}SA - B \right\|_F^2 \leq \|B\|_F^2 (1 + 8\delta^2 K) - \sum_{i=1}^K \|DA_i\|_2^4 + \underset{\delta\to 0}{\mathcal{O}} \left(\delta^3 \right) . \tag{8.17}$$

The first term is the squared Frobenius norm of a Wishart matrix and we can show

$$\mathbb{E}_D\left[\|B\|_F^2\right] = \frac{K(K-1)}{p} + K \ .$$

The columns A_i are chosen in $\mathcal{E}_{\delta,i}$, we can thus show that

$$\mathbb{E}_D\left[\max_{u\in\mathcal{E}_{\delta,i}} \|Du\|_2^4\right] \geq 1 + 4\delta\mathbb{E}_D\left[\sqrt{\|D^\mathsf{T} d_i\|_2^2 - 1}\right] + 6\delta^2\mathbb{E}_D\left[\|D^\mathsf{T} d_i\|_2^2 - 1\right] + \underset{\delta\to 0}{\mathcal{O}}\left(\delta^3\right) \ . \tag{8.18}$$

Denoting Y_i the random variable such that $pY_i^2 = p(\|D^\mathsf{T} d_i\|_2^2 - 1)$, we can compute the lower bounds

$$\mathbb{E}_D\left[Y_i\right] = \sqrt{\frac{2}{p}}\frac{\Gamma\left(\frac{K}{2}\right)}{\Gamma\left(\frac{K-1}{2}\right)} \geq \frac{K-1}{\sqrt{pK}} \quad \text{and} \quad \mathbb{E}_D\left[Y_i^2\right] = \frac{K-1}{p}$$

Combining these results with (8.18) yields the following lower bound when $\delta \to 0$,

$$\mathbb{E}_D\left[\max_{u\in\mathcal{E}_{\delta,i}} \|D^\mathsf{T} u\|_2^4\right] \gtrsim 1 + 4\delta\frac{K-1}{\sqrt{pK}} + 6\delta^2\frac{K-1}{p} + \underset{\delta\to 0}{\mathcal{O}}\left(\delta^3\right)$$

The final bound is obtained using these results with (8.17). □

8.3.2 Controlling the Rotation of the ℓ_1-norm

In this subsection, we analyze the deformation of the ℓ_1-norm due to a rotation of the code space with a diagonally dominant matrix $A \subset \mathcal{E}_\delta$.

Lemma 8.6. *Let $A \subset \mathcal{E}_\delta$ be a diagonally dominant matrix and let z be a random variable in $\mathbb{R}^K$ with iid coordinates z_i. Then*

$$\mathbb{E}_{z,D}\left[\delta_A(z)\right] \leq \lambda\mathbb{E}_z\left[\|z\|_1\right]\left(\delta\sqrt{K-1} - \frac{\delta^2}{2} + \underset{\delta\to 0}{\mathcal{O}}\left(\delta^4\right)\right)$$

Proof. **sketch for Lemma 8.6.** (The full proof can be found Subsubsection 8.7.2.3)
First, we show that if z is a *random variable* in $\mathbb{R}^K$ with *iid* coordinates z_i, then

$$\mathbb{E}_{z,D}\left[\frac{\|Az\|1}{\|z\|_1}\middle|\|z\|_1\right] \leq \frac{\mathbb{E}_D\left[\|A\|_{1,1}\right]}{K} \ .$$

This decouples the expectations and we obtain the following upper bound

$$\mathbb{E}_z\left[\delta_A(z)\right] \leq \lambda\mathbb{E}_z\left[\|z\|_1\right]\frac{\mathbb{E}_D\left[\|A\|_{1,1}\right] - \|\boldsymbol{I}_K\|_{1,1}}{K} \ .$$

Then, for $A \subset \mathcal{E}_\delta$, the ℓ_1-norm of the columns A_i is

$$\mathbb{E}_D\left[\|A_i\|_1\right] \leq \sqrt{1-\delta^2} + \delta\sqrt{K-1} \ .$$

Basic computations allow to show that

$$\frac{\mathbb{E}_D\left[\|A\|_{1,1}\right] - \boldsymbol{I}_K}{K} \leq \delta\sqrt{K-1} - \frac{\delta^2}{2} + \underset{\substack{\delta\to 0\\ \delta\to 0}}{\mathcal{O}}\left(\delta^4\right) \ .$$

□

8.3.3 Acceleration Conditions for Generic Dictionaries

The two previous results are used to control the upper bound of the cost update defined in Proposition 8.1 for generic dictionaries. It is interesting to see when this upper bound becomes smaller than the upper bound obtained using the identity $\boldsymbol{I}_K$.

Theorem 8.7 (Acceleration certificate)**.** *In expectation over the generic dictionary D, the factorization algorithm using a diagonally dominant matrix $A \subset \mathcal{E}_\delta$, has better performance for iteration $q+1$ than the normal ISTA iteration – which uses the identity – to solve (8.1) when*

$$\lambda \mathbb{E}_z \left[\|z^{(q+1)}\|_1 + \|z^*\|_1 \right] \leq \sqrt{\frac{K(K-1)}{p}} \mathbb{E}_z \left[\|z^{(q)} - z^*\|_2^2 \right]$$

We recall here that K denotes the number of atoms in the dictionary and p the dimension of the input space.

Proof. **sketch for Theorem 8.7.** (The full proof can be found Subsubsection 8.7.2.4) We denote $v = z^{(q)} - z^*$. For $A \subset \mathcal{E}_\delta$ with columns chosen greedily in $\mathcal{E}_{\delta,i}$, using results from Lemma 8.5 and Lemma 8.6,

$$\begin{aligned} \mathbb{E}_D \left[\min_{A \subset \mathcal{E}_\delta} \left\| A^{-1}SA - B \right\|_F^2 \|v\|_2^2 + \lambda \delta_A(z^*) - \delta_A(z^{(q+1)}) \right] \leq \\ \frac{(K-1)K}{p} \|v\|_2^2 + \delta\sqrt{K-1} \left(\lambda(\|z^{(q+1)}\|_1 + \|z^*\|_1) - \sqrt{\frac{K(K-1)}{p}} \|v\|_2^2 \right) \\ + \underset{\delta \to 0}{\mathcal{O}} \left(\delta^2 \right) \end{aligned} \tag{8.19}$$

Starting from the bound in Proposition 8.1, and using the results from (8.19), we obtain

$$\begin{aligned} \mathbb{E}_D \left[F(z^{(q+1)}) - F(z^*) \right] \leq \frac{(K-1)K}{p} \|z^{(q)} - z^*\|_2^2 \\ + \delta\sqrt{K-1} \underbrace{\left(\lambda \left(\|z^{(q+1)}\|_1 + \|z^*\|_1 \right) - \sqrt{\frac{K(K-1)}{p}} \|z^{(q)} - z^*\|_2^2 \right)}_{\leq 0} \\ + \underset{\delta \to 0}{\mathcal{O}} \left(\delta^2 \right) \end{aligned}$$

For small $\delta > 0$, this bound is better than with $\delta = 0$. Thus, in expectation, the performances of the algorithm based on the factorization with $A \subset \mathcal{E}_\delta$ are better than the one of ISTA on this iteration. □

From this theorem, we derive a bound on the maximal resolution where the factorization algorithm can provide an acceleration compared to ISTA. This bound only depends on the parameters of the problem.

Corollary 8.8 (Resolution gap)**.** *If the input distribution and the regularization parameter λ verify*

$$\frac{\lambda\sqrt{p}}{8} \leq \mathbb{E}_z \left[\|z^*\|_1 \right],$$

Then for any resolution $\mathbb{E}_z\left[\|z^{(q)} - z^*\|_2\right] = \epsilon > 0$ *at iteration* q*, the performance of our factorization algorithm is better than the performance of ISTA, in expectation over the generic dictionaries.*

Proof. We denote the expected resolution at iteration q by $\epsilon = \mathbb{E}_z\left[\|z^{(q)} - z^*\|_2\right]$. We first remark that for $z^* \in \mathbb{R}^K$ and $z^{(q+1)} \in \mathbb{R}^K$,

$$\|z^{(q+1)}\|_1 \leq \|z^*\|_1 + \|z^{(q+1)} - z^*\|_1 \leq \|z^*\|_1 + \sqrt{K}\|z^{(q+1)} - z^*\|_2$$

Assuming that $\mathbb{E}_z\left[\|z^{(q+1)} - z^*\|_2\right] \leq \mathbb{E}_z\left[\|z^{(q)} - z^*\|_2\right]$, we have

$$\mathbb{E}_z\left[\|z^{(q+1)}\|_1\right] \leq \mathbb{E}_z\left[\|z^*\|_1\right] + \sqrt{K}\epsilon$$

Using this in the condition of Theorem 8.7, we obtain the condition

$$0 \leq \sqrt{\frac{K(K-1)}{p}}\epsilon^2 - \lambda\sqrt{K}\epsilon - 2\lambda\mathbb{E}_z\left[\|z^*\|_1\right] \tag{8.20}$$

If

$$\lambda^2 K \leq 8\lambda\mathbb{E}_z\left[\|z^*\|_1\right]\frac{\sqrt{K(K-1)}}{\sqrt{p}} \ ,$$

then the condition (8.20) is verified for all $\epsilon \in \mathbb{R}$. Simplifying the expression, we obtain the result as $\frac{\sqrt{K}}{\sqrt{K-1}} \geq 1$. □

8.4 Network Architectures for Adaptive Optimization

8.4.1 Learned ISTA

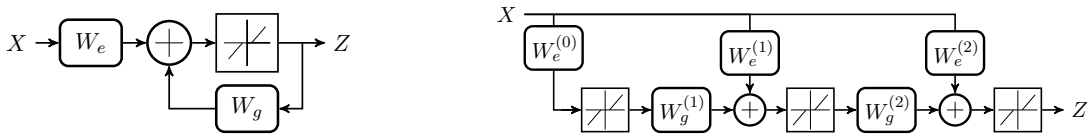
In Gregor & Lecun (2010), the authors introduced LISTA, a neural network constructed by considering ISTA as a recurrent neural net. At each step, ISTA performs the following 2-step procedure:

$$\left.\begin{array}{ll} 1. & u^{(q+1)} = z^{(q)} - \frac{1}{L}D^\mathsf{T}(Dz^{(q)} - x) = \underbrace{(\boldsymbol{I}_K - \frac{1}{L}D^\mathsf{T}D)}_{W_g} z^{(q)} + \underbrace{\frac{1}{L}D^\mathsf{T}}_{W_e} x \ , \\ 2. & z^{(q+1)} = \mathrm{Sh}\left(u^{(q+1)}, \frac{\lambda}{L}\right) \ \text{where } \mathrm{Sh}(u,\theta) = \mathrm{sign}(u)(|u| - \theta)_+ \ , \end{array}\right\} \text{ step } q \text{ of ISTA} \tag{8.21}$$

This procedure combines a linear operation to compute $u^{(q+1)}$ with an element-wise non-linearity. It can be summarized as a recurrent neural network, presented in Figure 8.1a., with tied weights. The authors in Gregor & Lecun (2010) considered the architecture Φ_Θ^Q with parameters $\Theta = (W_g^{(q)}, W_e^{(q)}, \theta^{(q)})_{q=1\ldots Q}$ obtained by unfolding Q times the recurrent network, as presented in Figure 8.1b. The layers ϕ_Θ^q are defined as

$$z^{(q+1)} = \phi_\Theta^q(z^{(q)}) := \mathrm{Sh}\left(W_g^{(q)}z^{(q)} + W_e^{(q)}x, \theta^{(q)}\right) \ . \tag{8.22}$$

If $W_g^{(q)} = \boldsymbol{I}_K - \frac{D^\mathsf{T}D}{L}$, $W_e^{(q)} = \frac{D^\mathsf{T}}{L}$ and $\theta^{(q)} = \frac{\lambda}{L}$ are fixed for all the Q layers, the output of this neural net is exactly the vector $z^{(Q)}$ resulting from Q steps of ISTA. With LISTA, the parameters Θ are learned using back propagation to minimize the cost function: $f(\Theta) = \mathbb{E}_x\left[F_x(\Phi_\Theta^Q(x))\right]$.



(a) **ISTA** - Recurrent Neural Network (b) **LISTA** - Unfolded network

Figure 8.1: Network architecture for ISTA/LISTA. The unfolded version (b) is trainable through backpropagation and approximates the sparse coding solution efficiently.

8.4.2 Learned FISTA

A similar algorithm can be derived from FISTA, the accelerated version of ISTA to obtain LFISTA (see Figure 8.2). The architecture is very similar to LISTA, now with two memory taps: it introduces a momentum term to improve the convergence rate of ISTA as follows:

1. $y^{(q)} = z^{(q)} + \frac{t_{q-1}-1}{t_q}(z^{(q)} - z^{(q-1)})$,
2. $z^{(q+1)} = \mathrm{Sh}\left(y^{(q)} - \frac{1}{L}\nabla E(y^{(q)}),\ \ \frac{\lambda}{L}\right) = \mathrm{Sh}\left((\boldsymbol{I}_K - \frac{1}{L}B)y^{(q)} + \frac{1}{L}D^{\mathsf{T}}x,\ \ \frac{\lambda}{L}\right)$,
3. $t_{q+1} = \frac{1+\sqrt{1+4t_q^2}}{2}$.

By substituting the expression for $y^{(q)}$ into the first equation, we obtain a generic recurrent architecture very similar to LISTA, now with two memory taps, that we denote by LFISTA:

$$z^{(q+1)} = \mathrm{Sh}\left(W_g^{(q)} z^{(q)} + W_m^{(q)} z^{(q-1)} + W_e^{(q)} x, \theta\right) .$$

This model is equivalent to running K-steps of FISTA when its parameters are initialized with

$$\begin{aligned} W_g^{(q)} &= \left(1 + \frac{t_{q-1}-1}{t_q}\right)\left(\boldsymbol{I}_K - \frac{1}{L}B\right) , \\ W_m^{(q)} &= \left(\frac{1-t_{q-1}}{t_q}\right)\left(\boldsymbol{I}_K - \frac{1}{L}B\right) , \\ W_e^{(q)} &= \frac{1}{L}D^{\mathsf{T}} . \end{aligned}$$

The parameters of this new architecture, presented in Figure 8.2 , are trained analogously as in the LISTA case.

8.4.3 Factorization Network

Our analysis in Section 8.2 suggests a re-factorization of LISTA in a more structured class of parameters. Following the same basic architecture, and using (8.5), the network FacNet, Ψ_Θ^K is formed using layers such that:

$$z^{(q+1)} = \psi_\Theta^q(z^{(q)}) := A_q^{\mathsf{T}} \mathrm{Sh}\left(A_q z^{(q)} - S_q^{-1} A_q (D^{\mathsf{T}} D z^{(q)} - D^{\mathsf{T}} x),\ \ \lambda S_Q^{-1}\right) , \qquad (8.23)$$

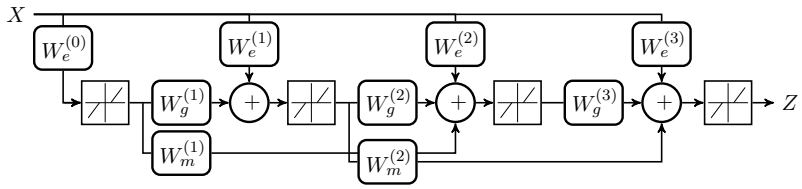


Figure 8.2: Network architecture for LFISTA. This network is trainable through back-propagation and approximates the sparse coding solution efficiently.

with S_q diagonal and A_q unitary, the parameters of the q-th layer. The parameters obtained after training such a network with back-propagation can be used with the theory developed in Section 8.2. Up to the last linear operation $A_q^\intercal$ of the network, this network is a re-parametrization of LISTA in a more constrained parameter space. Thus, LISTA is a generalization of this proposed network and should have performances at least as good as FacNet, for a fixed number of layers.

The optimization can also be performed using backpropagation. To enforce the unitary constraints on A_q, the cost function is modified with a penalty:

$$f(\Theta) = \mathbb{E}_x \left[F_x(\Psi_\Theta^Q(x)) \right] + \frac{\mu}{Q} \sum_{q=1}^{Q} \left\| \boldsymbol{I}_K - A_q^\intercal A_q \right\|_2^2 , \tag{8.24}$$

with $\Theta = (A_q, S_q)_{q=1\dots Q}$ the parameters of the Q layers and μ a scaling factor for the regularization. The resulting matrix $A^{(q)}$ is then projected on the Stiefel Manifold using a SVD to obtain final parameters, coherent with the network structure.

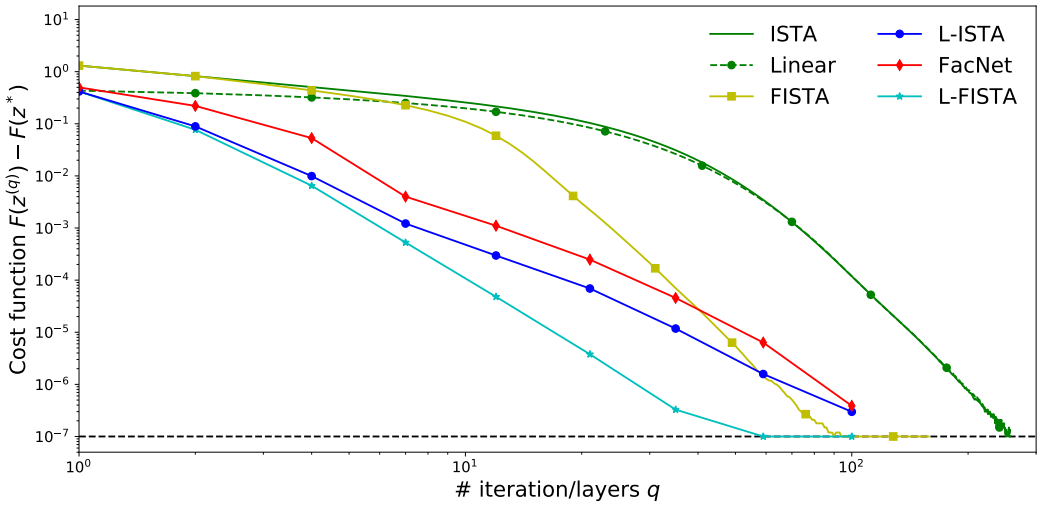
8.4.4 Linear Model

Finally, it is important to distinguish the performance gain resulting from choosing a suitable starting point and the acceleration resulting from our algorithm. To highlights the gain obtained by changing the starting point, we considered a linear model with one layer such that $z_{out} = A^{(0)}x$. This model is learned with SGD and the convex cost function $f(A^{(0)}) = \|(\boldsymbol{I}_P - DA^{(0)})x\|_2^2 + \lambda\|A^{(0)}x\|_1$. It computes a tradeoff between starting from the sparsest point $\mathbf{0}$ and a point with minimal reconstruction error y. Then, we observe the performance of the classical iteration of ISTA using z_{out} as a starting point instead of $\mathbf{0}$.

8.5 Numerical Experiments

This section provides numerical arguments to analyze adaptive optimization algorithms and their performances, and relates them to the theoretical properties developed in the previous section. All the experiments were run using Python and Tensorflow. For all the experiments, the training is performed using Adagrad (Duchi et al., 2011). The code to reproduce the figures is available online[2].

(a) Sparsity level $\rho = \frac{1}{20}$.



(b) Sparsity level $\rho = \frac{1}{4}$.

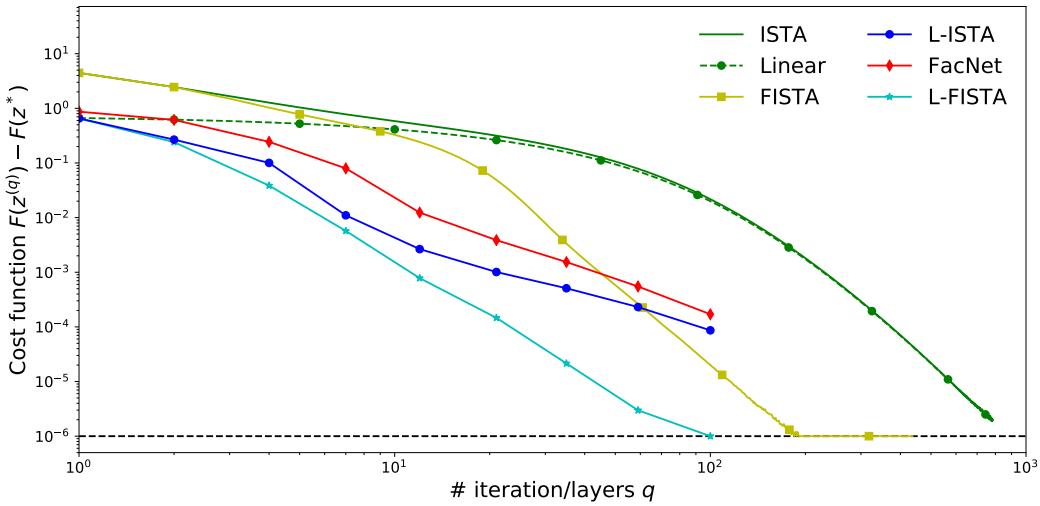


Figure 8.3: Evolution of the cost function $F(z^{(q)}) - F(z^*)$ with the number of layers or the number of iterations q for different sparsity level.

8.5.1 Acceleration for Generic Dictionaries

In order to disentangle the role of dictionary structure from the role of data distribution structure, the minimization problem is tested using a synthetic generative model with no structure in the weights distribution. First, K atoms $d_k \in \mathbb{R}^p$ are drawn *iid* from a multivariate Gaussian with mean $\mathbf{0}$ and covariance $\boldsymbol{I}_p$ and the dictionary D is defined as $\left(d_k/\|d_k\|_2\right)_{k=1\dots K}$. The data points are generated from its sparse codes following a Bernoulli-Gaussian model. The coefficients $z = (z_1, \dots, z_K)$ are constructed with $z_i = b_i a_i$, where $b_i \sim \mathcal{B}(\rho)$ and $a_i \sim \mathcal{N}(0, \sigma \boldsymbol{I}_K)$, where ρ controls the sparsity of the data. The values are set to $K = 100$, $p = 64$ for the dictionary dimension, $\rho = 5/K$ for the sparsity level and $\sigma = 10$ for the activation coefficient generation parameters. The

[2]The code can be found at https://github.com/tomMoral/AdaptiveOptim

sparsity regularization is set to λ=0.01. The batches used for the training are generated with the model at each step and the cost function is evaluated over a fixed test set, not used in the training.

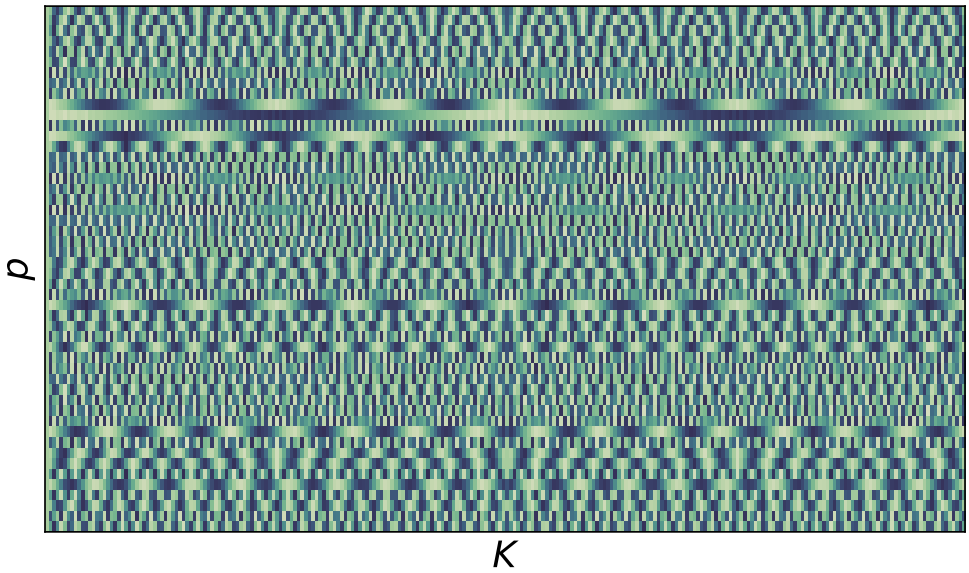
Figure 8.3 displays the cost performance for methods ISTA/FISTA/Linear relatively to their iterations and for methods LISTA/LFISTA/FacNet relatively to the number of layers used to solve our generated problem. Linear has performances comparable to learned methods with the first iteration but a gap appears as the number of layers increases, until a point where it achieves the same performances as non adaptive methods. This highlights that the adaptation is possible in the subsequent layers of the networks, going farther than choosing a suitable starting point for iterative methods. The first layers achieve a large gain over the classical optimization strategy, by leveraging the structure of the problem. This appears even with no structure in the sparsity patterns of input data, in accordance with the results in the previous section. We also observe diminishing returns as the number of layers increases. This results from the phase transition described in Section 8.2.3, as the last layers behave as ISTA steps and do not speed up the convergence. The 3 learned algorithms are always performing at least as well as their classical counterpart, as it was stated in Theorem 8.2. We also explored the effect of the sparsity level in the training and learning of adaptive networks. In the denser setting, the arbitrage between the ℓ_1-norm and the squared error is easier as the solution has a lot of non-zero coefficients. In this setting, the approximate method is more precise than in the very sparse setting where the approximation must perform a fine selection of the coefficients. But it also yields lower gain at the beggining as the sparser solution can move faster.

There is a small gap between LISTA and FacNet in this setup. This can be explained from the extra constraints on the weights that we impose in the FacNet, which effectively reduce the parameter space by half. Also, we implement the unitary constraints on the matrix A by a soft regularization (see (8.24)), involving an extra hyper-parameter μ that also contributes to the small performance gap. In any case, these experiments show that our analysis accounts for most of the acceleration provided by LISTA, as the performance of both methods are similar, up to optimization errors.

8.5.2 Adverse Dictionary

In this experiment, we would like to show that the limits of FacNet are also limits for LISTA. The idea is thus to design a dictionary for which we know that FacNet will fail, and see if LISTA is able to accelerate the resolution of (8.1). The results from Section 8.2 show that problems with a gram matrix composed of large eigenvalues associated to non sparse eigenvectors are harder to accelerate. Indeed, it is not possible in this case to find a quasi diagonalization of the matrix B that does not distort the ℓ_1 norm. It is possible to generate such a dictionary using Harmonic Analysis. The Discrete Fourier Transform (DFT) distorts a lot the ℓ_1 ball, since a very sparse vector in the temporal space is transformed in widely spread spectrum in the Fourier domain. We can thus design a dictionary for which FacNet performances should be degraded. $D = \left(d_k / \|d_k\|_2\right)_{k=1\dots K}$ is constructed such that $d_{k,j} = e^{-2\pi i k \zeta_j}$, with $\left(\zeta_j\right)_{j \leq p}$ randomly selected from $\left\{1/K, \dots, K/2/K\right\}$ without replacement.

(a) Adverse dictionary



(b) Performances

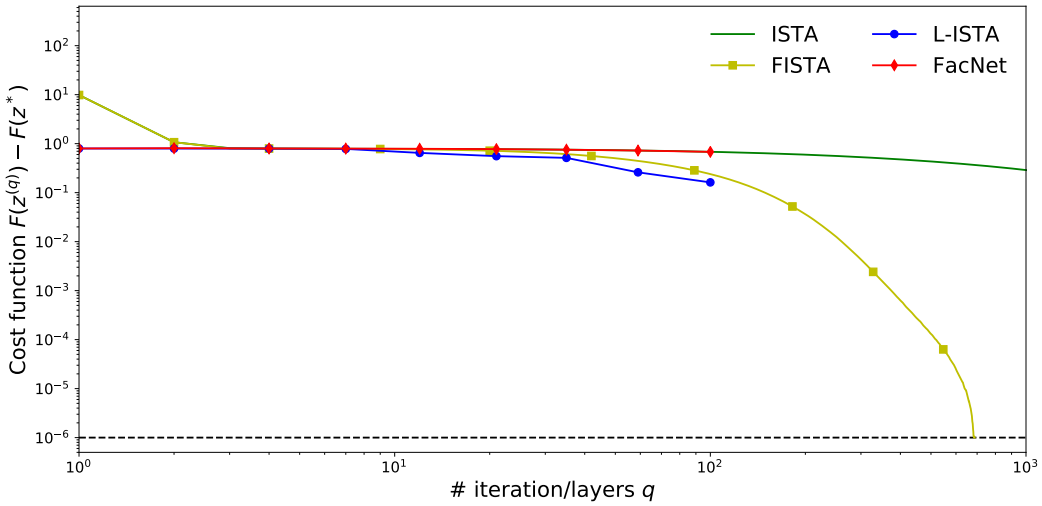


Figure 8.4: Evolution of the cost function $F(z^{(q)}) - F(z^*)$ with the number of layers or the number of iterations q for a problem generated with an adversarial dictionary.

The resulting performances are reported in Figure 8.4. The first layer provides a big gain by changing the starting point of the iterative methods. It realizes an arbitrage of the tradeoff between starting from $\mathbf{0}$ and starting from y . But the next layers do not yield any extra gain compared to the original ISTA algorithm. After 4 layers, the cost performance of both adaptive methods and ISTA are equivalent. It is clear that in this case, FacNet does not accelerate efficiently the sparse coding, in accordance with our result from Section 8.2. LISTA also displays poor performances in this setting. This provides further evidence that FacNet and LISTA share the same acceleration mechanism as adversarial dictionaries for FacNet are also adversarial for LISTA.

8.5.3 Sparse Coding with Over Complete Dictionary on Images

Wavelet Encoding for Natural Images. A highly structured dictionary composed of translation invariant Haar wavelets is used to encode 8x8 patches of images from the PASCAL VOC 2008 data set. The network is used to learn an efficient sparse coder for natural images over this family. 500 images are sampled from data set to train the encoder. Training batches are obtained by uniformly sampling patches from the training image set to feed the stochastic optimization of the network. The encoder is then tested with 10000 patches sampled from 100 new images from the same data set.

Learned Dictionary for MNIST. To evaluate the performance of LISTA for dictionary learning, LISTA was used to encode MNIST images over an unconstrained dictionary, learned *a priori* using classical dictionary learning techniques. The dictionary of 100 atoms was learned from 10000 MNIST grayscale images, scaled to 17x17 using the implementation of Mairal et al. (2010) proposed in scikit-learn, with $\lambda = 0.05$. Then, the networks were trained through backpropagation using all the 60000 images from the training set of MNIST. Finally, the performance of these encoders were evaluated with the 10000 images of the training set of MNIST.

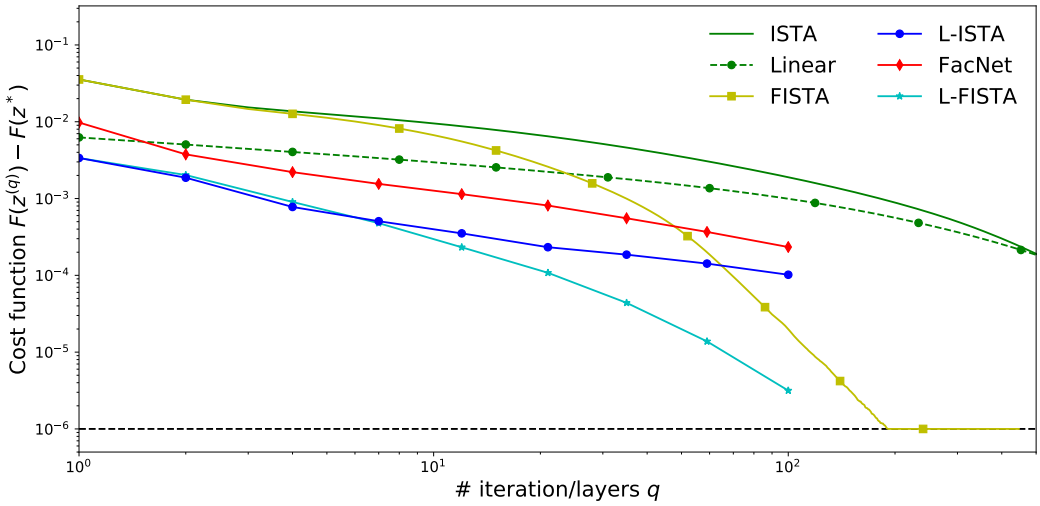
The Figure 8.5 displays the cost performance of the adaptive procedures compared to non-adaptive algorithms. In both scenario, FacNet has performances comparable to the one of LISTA and their behavior are in accordance with the theory developed in Section 8.2. The gains become smaller for each added layer and the initial gain is achieved for dictionary either structured or unstructured. The MNIST case presents a much larger gain compare to the experiment with natural images. This results from the difference of structure of the input distribution, as the MNIST digits are much more constrained than patches from natural images and the network is able to leverage it to find a better encoder. In the MNIST case, a network composed of 12 layers is sufficient to achieve performance comparable to ISTA with more than 1000 iterations.

8.6 Conclusion

In this chapter we studied the problem of finite computational budget approximation of sparse coding. Inspired by the ability of neural networks to accelerate over splitting methods on the first few iterations, we have studied which properties of the dictionary matrix and the data distribution lead to such acceleration. Our analysis reveals that one can obtain acceleration by finding approximate matrix factorizations of the dictionary which nearly diagonalize its Gram matrix, but whose orthogonal transformations leave approximately invariant the ℓ_1 ball. By appropriately balancing these two conditions, we show that the resulting rotated proximal splitting scheme has an upper bound which improves over the ISTA upper bound under appropriate sparsity.

In order to relate this specific factorization property to the actual LISTA algorithm, we have introduced a re-parametrization of the neural network that specifically computes the factorization, and incidentally provides reduced learning complexity (fewer parameters) from the original LISTA. Numerical experiments of Section 8.5 show that such re-parametrization recovers the same gains as the original neural network, providing evidence that our theoretical analysis is partially explaining the behavior of the LISTA neural network. Our acceleration scheme is inherently transient, in the sense that once the iterates are sufficiently close to the optimum, the factorization is not effective anymore. This transient effect is also consistent with the performance observed numerically,

(a) Pascal VOC 2008



(b) MNIST

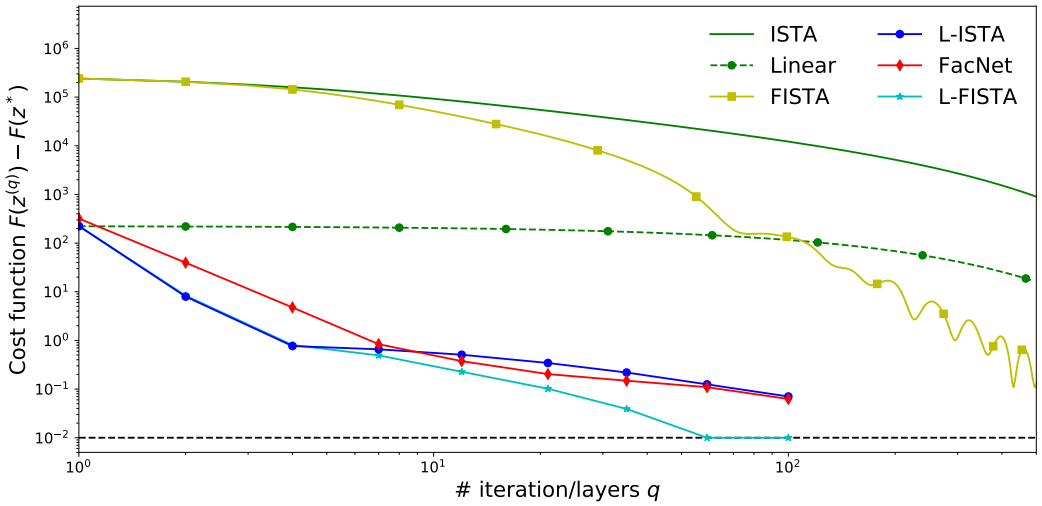


Figure 8.5: Evolution of the cost function $F(z_k) - F(z^*)$ with the number of layers or the number of iterations k for two image data sets.

although the possibility remains open to find alternative models that further exploit the particular structure of the sparse coding. Finally, we provide evidence that successful matrix factorization is not only sufficient but also necessary for acceleration, by showing that Fourier dictionaries are not accelerated.

Despite these initial results, a lot remains to be understood on the general question of optimal trade offs between computational budget and statistical accuracy. Our analysis did not take into account any probabilistic consideration so far, e.g. obtain approximations that hold with high probability for a given model. Another area of further study is the extension of our analysis to the FISTA case, and more generally to other inference tasks that are currently solved via iterative procedures compatible with neural network parametrization, such as inference in Graphical Models using Belief Propagation or other ill-posed inverse problems. Finally, specializing this results for the convolutional sparse coding would be a very interesting research direction. The convolutional case

is a specific case of our study and the results can be applied directly but they may be pessimistic as they do not account for the band-circulant structure of the dictionary. Thus, the analysis of this specific case could lead to improved bounds, which would better explain the acceleration in the convolutional case.

8.7 Proofs

8.7.1 Proofs for the Convergence Rate of FacNet

Proposition 8.7.1. *Suppose that $R = A^\mathsf{T} SA - B$ is positive definite, and define*

$$z^{(q+1)} = \operatorname*{argmin}_z \widetilde{F}(z, z^{(q)}) \ , \ \textit{and} \tag{8.25}$$

$\delta_A(z) = \|Az\|_1 - \|z\|_1$. *Then we have*

$$\begin{aligned} F(z^{(q+1)}) - F(z^*) \leq & \frac{1}{2}\left((z^* - z^{(q)})^\mathsf{T} R(z^* - z^{(q)}) - (z^* - z^{(q+1)})^\mathsf{T} R(z^* - z^{(q+1)})\right) \\ & + \langle \partial\delta_A(z^{(q+1)}), z^{(q+1)} - z^* \rangle \ . \end{aligned} \tag{8.26}$$

Proof. We define

$$f(t) = F\left(tz^{(q+1)} + (1-t)z^*\right) \ , \ t \in [0,1] \ .$$

Since F is convex, f is also convex in $[0,1]$. Since $f(0) = F(z^*)$ is the global minimum, it results that $f'(t)$ is increasing in $(0,1]$, and hence

$$F(z^{(q+1)}) - F(z^*) = f(1) - f(0) = \int_0^1 f'(t)dt \leq f'(1) \ ,$$

where $f'(1)$ is any element of $\partial f(1)$. Since $\delta_A(z)$ is a difference of convex functions, its subgradient can be defined as a limit of infimal convolutions Hiriart-Urruty (1991). We have

$$\partial f(1) = \langle \partial F(z^{(q+1)}), z^{(q+1)} - z^* \rangle \ ,$$

and since

$$\partial F(z) = \partial \widetilde{F}(z, z^{(q)}) - R(z - z^{(q)}) - \partial\delta_A(z) \ \text{ and } 0 \in \partial \widetilde{F}(z^{(q+1)}, z^{(q)})$$

it results that

$$\partial F(z^{(q+1)}) = -R(z^{(q+1)} - z^{(q)}) - \partial\delta_A(z^{(q+1)}) \ ,$$

and thus

$$F(z^{(q+1)}) - F(z^*) \leq (z^* - z^{(q+1)})^\mathsf{T} R(z^{(q+1)} - z^{(q)}) + \langle \partial\delta_A(z^{(q+1)}), (z^* - z^{(q+1)}) \rangle \ . \tag{8.27}$$

(8.10) is obtained by observing that

$$(z^* - z^{(q+1)})^\mathsf{T} R(z^{(q+1)} - z^{(q)}) \leq \frac{1}{2}\left((z^* - z^{(q)})^\mathsf{T} R(z^* - z^{(q)}) - (z^* - z^{(q+1)})^\mathsf{T} R(z^* - z^{(q+1)})\right) \ , \tag{8.28}$$

thanks to the fact that $R \succ 0$. □

Theorem 8.2. *Let A_q, S_q be the pair of unitary and diagonal matrices corresponding to iteration q, chosen such that $R_q = A_q^\mathsf{T} S_q A_q - B \succ 0$. It results that*

$$F(z^{(q)}) - F(z^*) \leq \frac{(z^* - z^{(0)})^\mathsf{T} R_0 (z^* - z^{(0)}) + 2L_{A_0}(z^{(1)})\|z^* - z_1\|_2}{2q} + \frac{\alpha - \beta}{2q} \ , \tag{8.11}$$

$$\begin{aligned} \textit{with} \quad \alpha = & \sum_{i=1}^{q-1} \left(2L_{A_i}(z^{(i+1)})\|z^* - z^{(i+1)}\|_2 + (z^* - z^{(i)})^\mathsf{T}(R_{i-1} - R_i)(z^* - z^{(i)})\right) \ , \\ \beta = & \sum_{i=0}^{q-1} (i+1)\left((z^{(i+1)} - z^{(i)})^\mathsf{T} R_i (z^{(i+1)} - z^{(i)}) + 2\delta_{A_i}(z^{(i+1)}) - 2\delta_{A_i}(z^{(i)})\right) \ , \end{aligned}$$

where $L_A(z)$ denotes the local lipschitz constant of δ_A at z.

Proof. The proof is adapted from (Beck & Teboulle, 2009, Theorem 3.1).
From Proposition 8.7.1, we start by using (8.26) to bound terms of the form $F(z^{(n)}) - F(z^*)$:

$$F(z^{(n)}) - F(z^*) \leq \langle \partial\delta_{A_n}(z^{(n+1)}), (z^* - z^{(n+1)})\rangle + \frac{1}{2}\left((z^* - z^{(n)})^\mathsf{T} R_n (z^* - z^{(n)}) - (z^* - z^{(n+1)})^\mathsf{T} R_n (z^* - z^{(n+1)})\right) .$$

Adding these inequalities for $n = 0 \dots k-1$ we obtain

$$\begin{aligned}\left(\sum_{n=0}^{q-1} F(z^{(n)})\right) - qF(z^*) \quad &\leq \quad \sum_{n=0}^{q-1} \langle \partial\delta_{A_n}(z^{(n+1)}), (z^* - z^{(n+1)})\rangle \\ &+\frac{1}{2}\left((z^* - z^{(0)})^\mathsf{T} R_0 (z^* - z^{(0)}) - (z^* - z^{(q)})^\mathsf{T} R_{q-1} (z^* - z^{(q)})\right) \\ &+\frac{1}{2}\sum_{n=1}^{q-1} (z^* - z^{(n)})^\mathsf{T} (R_{n-1} - R_n)(z^* - z^{(n)}) .\end{aligned} \tag{8.29}$$

On the other hand, we also have

$$\begin{aligned}F(z^{(n)}) - F(z^{(n+1)}) \quad &\geq \quad F(z^{(n)}) - \tilde{F}(z^{(n)}, z^{(n)}) + \tilde{F}(z^{(n+1)}, z^{(n)}) - F(z^{(n+1)}) \\ &= \quad -\delta_{A_n}(z^{(n)}) + \delta_{A_n}(z^{(n+1)}) + \frac{1}{2}(z^{(n+1)} - z^{(n)})^\mathsf{T} R_n (z^{(n+1)} - z^{(n)}) ,\end{aligned}$$

which results in

$$\begin{aligned}\sum_{n=0}^{q-1}(n+1)(F(z^{(n)}) - F(z^{(n+1)})) \quad &\geq \quad \frac{1}{2}\sum_{n=0}^{q-1}(n+1)(z^{(n+1)} - z^{(n)})^\mathsf{T} R_n (z^{(n+1)} - z^{(n)}) \\ &+\sum_{n=0}^{q-1}(n+1)\left(\delta_{A_n}(z^{(n+1)}) - \delta_{A_n}(z^{(n)})\right)\end{aligned} \tag{8.30}$$

$$\begin{aligned}\left(\sum_{n=0}^{q-1} F(z^{(n)})\right) - qF(z^{(q)}) \quad &\geq \quad \sum_{n=0}^{q-1}(n+1)\Big(\frac{1}{2}(z^{(n+1)} - z^{(n)})^\mathsf{T} R_n (z^{(n+1)} - z^{(n)}) \\ &+\delta_{A_n}(z^{(n+1)}) - \delta_{A_n}(z^{(n)})\Big) .\end{aligned} \tag{8.31}$$

Combining (8.29) and (8.30) we obtain

$$\begin{aligned}F\left(z^{(q)}\right) - F(z^*) \leq \frac{(z^* - z^{(0)})^\mathsf{T} R_0 (z^* - z^{(0)}) + 2\langle \nabla\delta_{A_0}(z^{(1)}), (z^* - z^{(1)})\rangle}{2q} \\ + \frac{\alpha - \beta}{2q}\end{aligned} \tag{8.32}$$

with

$$\alpha = \sum_{n=1}^{q-1}\left(2\langle \nabla\delta_{A_n}(z^{(n+1)}), (z^* - z^{(n+1)})\rangle + (z^* - z^{(n)})^\mathsf{T}(R_{n-1} - R_n)(z^* - z^{(n)})\right) ,$$

$$\beta = \sum_{n=0}^{q-1}(n+1)\left((z^{(n+1)} - z^{(n)})^\mathsf{T} R_n (z^{(n+1)} - z^{(n)}) + 2\delta_{A_n}(z^{(n+1)}) - 2\delta_{A_n}(z^{(n)})\right) .$$

□

Corollary 8.3. *If $A_q = \boldsymbol{I}_K$, $S_q = \|B\|\boldsymbol{I}_K$ for $q \geq 1$ then*

$$F(z^{(q)}) - F(z^*) \leq \frac{(z^* - z^{(0)})^\mathsf{T} R_0 (z^* - z^{(0)}) + (z^* - z^{(1)})^\mathsf{T} R_0 (z^* - z^{(1)})^\mathsf{T}}{2q} + \frac{L_{A_0}(z_1)(\|z^* - z^{(1)}\| + \|z^{(1)} - z^{(0)}\|)}{q} . \tag{8.12}$$

Proof. We verify that in that case, $R_{n-1} - R_n \equiv 0$ and for $n > 1$ and $\delta_{A_n} \equiv 0$ for $n > 0$. □

8.7.2 Existence of a Gap for Generic Dictionaries.

8.7.2.1 Properties of $\mathcal{E}_\delta$

Proposition 8.7.2. *If a matrix A has its columns in $\mathcal{E}_{\delta,i}$, then it is almost unitary for small value of δ. More precisely, denoting $\nu = A^\mathsf{T} A - \boldsymbol{I}_K$, when $\delta \to 0$*

$$\|\nu\|_F = \mathcal{O}\left(\delta\right)$$

Proof. Let $\nu = A^\mathsf{T} A - \boldsymbol{I}_K$. As A_i are in $\mathcal{E}_{\delta,i}$,

$$\nu_{i,i} = A_i^\mathsf{T} A_i - 1 = 0$$

We can verify that for $i \neq j$

$$\begin{aligned} \nu_{i,j} = A_i^\mathsf{T} A_j &= \delta\sqrt{1-\delta^2}(e_i^\mathsf{T} h_j + e_j^\mathsf{T} h_i) + \delta^2 h_i^\mathsf{T} h_j \\ &= \delta(e_i^\mathsf{T} h_j + e_j^\mathsf{T} h_i) + \mathcal{O}\left(\delta^2\right) \end{aligned}$$

This permits to bound the Frobenius norm of ν *i.e.*

$$\|\nu\|_F^2 = \sum_{1 \leq i,j \leq K} \nu_{i,j}^2 = \delta^2 \sum_{\substack{1 \leq i,j \leq K \\ i \neq j}} (e_i^\mathsf{T} h_j + e_j^\mathsf{T} h_i)^2 + \mathcal{O}\left(\delta^3\right) .$$

$$\begin{aligned} \|\nu\|_F^2 = \sum_{1 \leq i,j \leq K} \nu_{i,j}^2 &= \delta^2 \sum_{\substack{1 \leq i,j \leq K \\ i \neq j}} (e_i^\mathsf{T} h_j + e_j^\mathsf{T} h_i)^2 + \mathcal{O}\left(\delta^3\right) , \\ &= \delta^2 \sum_{1 \leq i,j \leq K} (h_{i,j} + h_{j,i})^2 + \mathcal{O}\left(\delta^3\right) , \\ &= \delta^2 \|H + H^\mathsf{T}\|_F^2 + \mathcal{O}\left(\delta^3\right) , \\ &= 4\delta^2 \|H\|_F^2 + \mathcal{O}\left(\delta^3\right) = 4\delta^2 K + \mathcal{O}\left(\delta^3\right) . \qquad \text{as } \|h_i\|_2^2 = 1 \end{aligned}$$

□

Proposition 8.7.3. *For $A \subset \mathcal{E}_\delta$, and for any symmetric matrix $U \in \mathbb{R}^{K \times K}$, when $\delta \to 0$,*

$$\left\| A^{-1} U A \right\|_F^2 \leq \|U\|_F^2 + \mathcal{O}\left(\delta^3\right)$$

Proof. For $U \in \mathbb{R}^{K\times K}$ symmetric, as A is quasi unitary,

$$\begin{aligned}\left\|A^{-1}UA\right\|_F^2 &= \mathbf{Tr}\left[A^\mathsf{T}U(A^{-1})^\mathsf{T}A^{-1}UA\right] = \mathbf{Tr}\left[U(A^\mathsf{T}A)^{-1}UAA^\mathsf{T}\right]\\ &= \mathbf{Tr}\left[U(\boldsymbol{I}_K+\nu)^{-1}U(\boldsymbol{I}_K+\nu)\right] = \mathbf{Tr}\left[U(\boldsymbol{I}_K-\nu+\nu^2+\mathcal{O}\left(\nu^3\right))U(\boldsymbol{I}_K+\nu)\right]\\ &= \mathbf{Tr}\left[UU+U\nu^2U-U\nu U\nu+\mathcal{O}\left(\nu^3\right))\right]\\ &= \|U\|_F^2+\|\nu U\|_F^2-\|\nu^{\frac{1}{2}}U\nu^{\frac{1}{2}}\|_F^2+\mathcal{O}\left(\|\nu^{3/2}\|_F^2\right)\end{aligned}$$

Notice that

$$\|\nu^{\frac{1}{2}}U\nu^{\frac{1}{2}}\|_F = \|(U\nu)^{\frac{1}{2}\mathsf{T}}(U\nu)^{\frac{1}{2}}\|_F = \|(U\nu)^{\frac{1}{2}}\|_F^2 \geq \|U\nu\|_F$$

Thus $\|\nu U\|_F^2 - \|\nu^{\frac{1}{2}}U\nu^{\frac{1}{2}}\|_F^2 \leq 0$ and by submultiplicativity of $\|\cdot\|_F^2$,

$$\|\nu^{3/2}\|_F^2 \leq \|\nu\|_F^3 = \mathcal{O}\left(\delta^3\right) \quad \Rightarrow \quad \mathcal{O}\left(\|\nu^{3/2}\|_F^2\right) = \mathcal{O}\left(\delta^3\right) .$$

By combining all these results, we get:

$$\left\|A^{-1}UA\right\|_F^2 \leq \|U\|_F^2 + \mathcal{O}\left(\delta^3\right)$$

□

Proposition 8.7.4. *For a matrix $A \subset \mathcal{E}_\delta$ and any matrices $X, Y \in \mathbb{R}^{K\times K}$, when $\delta \to 0$,*

$$\left\|A^{-1}XA - Y\right\|_F^2 \leq \left\|X - AYA^\mathsf{T}\right\|_F^2 + \|Y\|_F^2\,\|\nu\|_F^2 + \mathcal{O}\left(\delta^3\right) .$$

Proof. First, we split the error of replacing $\left\|A^{-1}XA - Y\right\|_F^2$ by $\left\|X - AYA^\mathsf{T}\right\|_F^2$ in two terms. Both are linked to the quasi unitarity of A. The first term arises as we replace A^{-1} by A^T,

$$\begin{aligned}\left\|A^{-1}XA - Y\right\|_F^2 = \left\|A^{-1}\left(X - AYA^{-1}\right)A\right\|_F^2 &= \left\|A^{-1}\left(X - AY\underbrace{\left(A^\mathsf{T}A-\nu\right)}_{\boldsymbol{I}_K}A^{-1}\right)A\right\|_F^2\\ &= \left\|A^{-1}\left(X - AYA^\mathsf{T} + AY\nu A^{-1}\right)A\right\|_F^2\\ &\leq 2\left\|A^{-1}\left(X - AYA^\mathsf{T}\right)A\right\|_F^2 + 2\left\|A^{-1}\left(AY\nu A^{-1}\right)A\right\|_F^2 \qquad \text{(use } (a+b)^2 < 2a^2+b^2\text{)}\\ &\leq 2\left\|X - AYA^\mathsf{T}\right\|_F^2 + 2\left\|Y\nu\right\|_F^2 + \mathcal{O}\left(\delta^3\right) \qquad \text{(Proposition 8.7.3)}\\ &\leq 2\left\|X - AYA^\mathsf{T}\right\|_F^2 + 2\left\|Y\right\|_F^2\left\|\nu\right\|_F^2 + \mathcal{O}\left(\delta^3\right) \qquad \text{(submultiplicativity } \|\cdot\|_F \text{)}\\ &\leq 2\left\|X - AYA^\mathsf{T}\right\|_F^2 + 8\delta^2K\left\|Y\right\|_F^2 + \mathcal{O}\left(\delta^3\right) \qquad \text{(Proposition 8.7.2)}\end{aligned}$$

□

Proposition 8.7.5. *For $A \subset \mathcal{E}_\delta$, and for any matrix $U \in \mathbb{R}^{K\times K}$, when $\delta \to 0$,*

$$\left\|A^\mathsf{T} U A\right\|_F^2 = \|U\|_F^2 \left(1 + \|\nu\|_F^2\right) + \mathcal{O}\left(\delta^3\right)$$

Proof.

$$\begin{aligned}
\|AUA^\mathsf{T}\|_F = \|AXA^\mathsf{T}AA^{-1}\|_F^2 &= \|AU(\boldsymbol{I}_K + \nu)A^{-1}\|_F^2 \\
&= \|U + U\nu\|_F^2 + \mathcal{O}\left(\delta^3\right) && \text{(Proposition 8.7.3)} \\
&= 2\|U\|_F^2 + 2\|U\nu\|_F^2 + \mathcal{O}\left(\delta^3\right) && \text{(triangular inequality)} \\
&= 2\|U\|_F^2 + 2\|U\|_F^2\|\nu\|_F^2 + \mathcal{O}\left(\delta^3\right) \\
& && (\|\cdot\|_F \text{ is sub multiplicative}) \\
&\leq 2\|U\|_F^2 + 8\delta^2 K\|U\|_F^2 + \mathcal{O}\left(\delta^3\right) && \text{(Proposition 8.7.2)}
\end{aligned}$$

□

8.7.2.2 Controling the Deviation of $\|\cdot\|_B$

Lemma 8.5. *For a generic dictionary D and a diagonally dominant matrix $A \subset \mathcal{E}_\delta$,*

$$\begin{aligned}
\mathbb{E}_D\left[\min_{A_i \in \mathcal{E}_{\delta,i}} \left\|A^{-1}SA - B\right\|_F^2\right] \leq &\frac{K(K-1)}{p} - 4\delta(K-1)\sqrt{\frac{K}{p}} \\
&+ \delta^2\left(8\mathbb{E}_D\left[\|B\|_F^4\right] - 6\frac{K(K-1)}{p}\right) + \underset{\delta\to 0}{\mathcal{O}}\left(\delta^3\right) .
\end{aligned}$$

Proof. First, we use the results from Proposition 8.7.4 to remove the inverse matrix A^{-1}

$$\left\|A^{-1}SA - B\right\|_F^2 \leq \left\|S - ABA^\mathsf{T}\right\|_F^2 + \|B\|_F^2\|\nu\|_F^2 + \mathcal{O}\left(\delta^3\right) .$$

Using Proposition 8.7.2 with $A \subset \mathcal{E}_\delta$,

$$\|\nu\|_F^2 = 4\delta^2 K + \mathcal{O}\left(\delta^3\right)$$

and

$$\left\|A^{-1}SA - B\right\|_F^2 \leq \left\|S - ABA^\mathsf{T}\right\|_F^2 + 4\delta^2 K\|B\|_F^2 + \mathcal{O}\left(\delta^3\right) .$$

Then, we only need to control $\left\|S - ABA^\mathsf{T}\right\|_F^2$. First we note that this can be split into 2 terms

$$\begin{aligned}
\left\|S - ABA^\mathsf{T}\right\|_F^2 = \sum_{i=1}^K \sum_{\substack{j=1 \\ j\neq i}}^K (A_i^\mathsf{T} B A_j)^2 &= \sum_{i=1}^K \sum_{j=1}^K (A_i^\mathsf{T} B A_j)^2 - \sum_{i=1}^K (A_i^\mathsf{T} B A_i)^2 \\
&= \left\|ABA^\mathsf{T}\right\|_F^2 - \sum_{i=1}^K \|DA_i\|_2^4 \\
&= \|B\|_F^2\left(1 + 4\delta^2 K\right) - \sum_{i=1}^K \|DA_i\|_2^4 + \mathcal{O}\left(\delta^3\right) && \text{(Proposition 8.7.5)}
\end{aligned}$$

The first term is the squared Frobenius norm of a Wishart matrix and can be controlled by

$$\begin{aligned}
\mathbb{E}_D\left[\|B\|_F^2\right] &= \mathbb{E}_D\left[\sum_{i=1}^{K}\sum_{j=1}^{K} B_{i,j}^2\right] = \sum_{i=1}^{K}\sum_{j=1}^{K}\mathbb{E}_D\left[\left(\sum_{l=1}^{p} d_{i,k}d_{j,k}\right)^2\right] \\
&= \sum_{i=1}^{K}\sum_{\substack{j=1\\ i\neq j}}^{K}\mathbb{E}_D\left[\left(\sum_{l=1}^{p} d_{i,l}d_{j,l}\right)^2\right] + \sum_{i=1}^{K}\mathbb{E}_D\left[\|d_i\|_2^4\right] \\
&= \sum_{i=1}^{K}\sum_{\substack{j=1\\ i\neq j}}^{K}\left[\sum_{l=1}^{p}\mathbb{E}_D\left[d_{j,l}^2 d_{i,l}^2\right] + \sum_{l=1}^{p}\sum_{\substack{m=1\\ m\neq l}}^{p}\underbrace{\mathbb{E}_D\left[d_{i,l}d_{i,m}d_{j,l}d_{j,m}\right]}_{=0}\right] + K \\
&= \sum_{i=1}^{K}\sum_{\substack{j=1\\ i\neq j}}^{K}\sum_{l=1}^{p}\mathbb{E}_{d_j}\left[d_{j,l}^2\right]\mathbb{E}_{d_i}\left[d_{i,l}^2\right] + K \qquad (d_i \text{ are independent}) \\
&= \sum_{i=1}^{K}\sum_{\substack{j=1\\ i\neq j}}^{K}\sum_{l=1}^{p}\frac{1}{p^2} + K = \frac{K(K-1)}{p} + K\ . \quad (\mathbb{E}_d\left[d_{i,j}^2\right] = \tfrac{1}{p}, \text{(Song \& Gupta, 1997)})
\end{aligned}$$

For the second term, consider $u \in \mathcal{E}_{\delta,i}$, such that $u = \sqrt{1-\mu^2}e_i + \mu h$ for $0 < \mu < \delta$, $h \in Span(e_i)^\perp$. Given $i \in [K]$, Be_i can be decomposed as $z_1 e_i + z_2 h_i$, with $h_i \in Span(e_i)^\perp \cap \mathcal{S}^{K-1}$. Using basic algebra, z_1 and z_2 are:

$$z_1 = e_i^\mathsf{T} B e_i = \|De_i\|_2^2 = \|d_i\|_2^2 = 1\ . \tag{8.33}$$

$$z_2^2 = \|Be_i\|_2^2 - z_1^2 = \|D^\mathsf{T} d_i\|_2^2 - 1 \tag{8.34}$$

Also, for all $i, j \in [K]$, if $i \neq j$ then $h_i^\mathsf{T} e_j = \begin{cases} 0 & \text{if } \|D^\mathsf{T} d_i\|_2^2 = 1 \\ \frac{d_i^\mathsf{T} d_j}{\|D^\mathsf{T} d_i\|_2^2 - 1} & \text{elsewhere} \end{cases}$.

Then

$$\begin{aligned}
\|Du\|_2^2 &= u^\mathsf{T} D^\mathsf{T} D u = u^\mathsf{T} B u \\
&= (1-\mu^2)\underbrace{\|De_i\|_2^2}_{\|d_i\|_2=1} + \mu^2\|Dh\|_2^2 + 2\mu\sqrt{1-\mu^2}\underbrace{h^\mathsf{T} B e_i}_{z_2 h^\mathsf{T} h_i}
\end{aligned}$$

Thus, with the notation from (8.34), $h^\mathsf{T} B e_i = z_2 h^\mathsf{T} h_i$ and

$$\|Du\|_2^2 = (1-\delta^2) + \delta^2\|Dh\|_2^2 + 2\delta\sqrt{1-\delta^2}\sqrt{\|D^\mathsf{T} d_i\|_2^2 - 1}\, h^\mathsf{T} h_i \tag{8.35}$$

Now we can use this to derive a lower bound on $\max_{u\in\mathcal{E}_{\delta,i}} \|D^\mathsf{T} u\|_2^2$ when $\delta \to 0$,

$$\max_{u\in\mathcal{E}_{\delta,i}} \|Du\|_2^2 \geq 1 + 2\delta\sqrt{\|D^\mathsf{T} d_i\|_2^2 - 1} + \delta^2\left(\|D^\mathsf{T} d_i\|_2^2 - 1\right) + \mathcal{O}\left(\delta^3\right)$$

$$(u=\hat{A}_i)$$

Taking the square of this relation yields

$$\max_{u\in\mathcal{E}_{\delta,i}} \|Du\|_2^4 \geq 1 + 4\delta\sqrt{\|D^\mathsf{T} d_i\|_2^2 - 1} + 6\delta^2 \left(\|D^\mathsf{T} d_i\|_2^2 - 1\right) + \mathcal{O}\left(\delta^3\right)$$

Taking the expectation yields

$$\mathbb{E}_D\left[\max_{u\in\mathcal{E}_{\delta,i}} \|Du\|_2^4\right] \geq 1 + 4\delta\mathbb{E}_D\left[\sqrt{\|D^\mathsf{T} d_i\|_2^2 - 1}\right] + 6\delta^2\mathbb{E}_D\left[\|D^\mathsf{T} d_i\|_2^2 - 1\right] + \mathcal{O}\left(\delta^3\right) \ .$$

The random variable $pY_i^2 = p(\|D^\mathsf{T} d_i\|_2^2 - 1)$ are distributed as χ^2_{K-1}. Indeed, the atoms d_i are uniformly distributed over $\mathcal{S}^{K-1}$. As this distribution is rotational invariant, without loss of generality, we can take $d_i = e_1$. Then Y_i is simply sum of $K-1$ squared normal gaussians rv with variance $\frac{1}{p}$,

$$Y_i^2 = \|D^\mathsf{T} e_1\|_2^2 - 1 = \sum_{j=2}^{K} d_{j,1}^2 \ ,$$

and $\sqrt{p}Y_i$ is distributed as χ_{K-1}. A lower bound for its expectation is

$$\mathbb{E}_D\left[Y_i\right] = \sqrt{\frac{2}{p}}\frac{\Gamma\left(\frac{K}{2}\right)}{\Gamma\left(\frac{K-1}{2}\right)} \geq \frac{K-1}{\sqrt{pK}} \quad \text{and} \quad \mathbb{E}_D\left[Y_i^2\right] = \frac{K-1}{p}$$

We derive a lower bound for the second term when $\delta \to 0$,

$$\mathbb{E}_D\left[\max_{u\in\mathcal{E}_{\delta,i}} \|D^\mathsf{T} u\|_2^4\right] \gtrsim 1 + 4\delta\frac{K-1}{\sqrt{pK}} + 6\delta^2\frac{K-1}{p} + \mathcal{O}\left(\delta^3\right)$$

Using these results, we derive an upper bound for the expected distortion of B with A with columns in $\mathcal{E}_{\delta,i}$,

$$\begin{aligned}
\mathbb{E}_D\left[\min_{A_i\in\mathcal{E}_{\delta,i}} \left\|S - ABA^\mathsf{T}\right\|_F^2\right] &\leq \mathbb{E}_D\left[\left\|B\right\|_F^2\right] - \sum_{i=1}^{K}\mathbb{E}_D\left[\max_{A_i\in\mathcal{E}_\delta} \|DA_i\|_2^4\right] + C_1\delta^2 + \mathcal{O}\left(\delta^3\right) \\
&\leq K + \frac{K(K-1)}{p} - \sum_{i=1}^{K} 1 + 4\delta\frac{K-1}{\sqrt{pK}} + C_2\delta^2 + \mathcal{O}\left(\delta^3\right) \\
&\leq \frac{K(K-1)}{p} - 4\delta(K-1)\sqrt{\frac{K}{p}} + C'\delta^2 + \mathcal{O}\left(\delta^3\right)
\end{aligned}$$

And

$$C' = \delta^2\left(4K\mathbb{E}_D\left[\|B\|_F^2\right] - 6\frac{K(K-1)}{p}\right)$$

This concludes our proof as

$$\mathbb{E}_D\left[\min_{A_i\in\mathcal{E}_{\delta,i}}\left\|A^{-1}SA-B\right\|_F^2\right]=\mathbb{E}_D\left[\min_{A_i\in\mathcal{E}_{\delta,i}}\left\|S-ABA^\intercal\right\|_F^2\right]+4\delta^2K\mathbb{E}_D\left[\|B\|_F^2\right]$$

(Proposition 8.7.4)

$$+\mathcal{O}\left(\delta^3\right)$$

$$\leq\frac{K(K-1)}{p}-4\delta(K-1)\sqrt{\frac{K}{p}}+C\delta^2+\mathcal{O}\left(\delta^3\right)\ .$$

(Lemma 8.5)

And

$$C=8K\mathbb{E}_D\left[\|B\|_F^2\right]-6\frac{K(K-1)}{p}$$

□

8.7.2.3 Controling $\mathbb{E}_{z\sim\mathcal{Z}}\left[\delta_A(z)\right]$

Lemma 8.6. *Let $A\subset\mathcal{E}_\delta$ be a diagonally dominant matrix and let z be a random variable in $\mathbb{R}^K$ with iid coordinates z_i. Then*

$$\mathbb{E}_{z,D}\left[\delta_A(z)\right]\leq\lambda\mathbb{E}_z\left[\|z\|_1\right]\left(\delta\sqrt{K-1}-\frac{\delta^2}{2}+\underset{\delta\to0}{\mathcal{O}}\left(\delta^4\right)\right)$$

Proof. For any random variable $z=(z_1,\ldots,z_K)\sim\mathcal{Z}\in\mathbb{R}^K$ *s.t.* the z_i are rotational invariant, then

$$1=\mathbb{E}_{z\sim\mathcal{Z}}\left[1\right]=\mathbb{E}_{z\sim\mathcal{Z}}\left[\frac{\|z\|_1}{\|z\|_1}\middle|\|z\|_1\right]=\sum_{i=1}^K\mathbb{E}_{z\sim\mathcal{Z}}\left[\frac{|z_i|}{\|z\|_1}\middle|\|z\|_1\right]=K\mathbb{E}_{z\sim\mathcal{Z}}\left[\frac{|z_1|}{\|z\|_1}\middle|\|z\|_1\right]$$

Thus we get:

$$\mathbb{E}_{z\sim\mathcal{Z}}\left[\frac{|z_1|}{\|z\|_1}\middle|\|z\|_1\right]=\frac{1}{K}\ .\tag{8.36}$$

Let $z=\left(z_1,\ldots z_K\right)$ be a vector of $\mathbb{R}^K$. Then

$$\|Az\|_1=\sum_{i=1}^K\left|\sum_{j=1}^K A_{i,j}z_j\right|\leq\sum_{i=1}^K\sum_{j=1}^K\left|A_{i,j}\right|\left|z_j\right|\leq\sum_{j=1}^K\left\|A_i\right\|_1\left|z_j\right|$$

Using (8.36), we can compute an upper bound for $\mathbb{E}_{z\sim\mathcal{Z}}\left[\frac{|Az|_1}{\|z\|_1}\right]$:

$$\mathbb{E}_{z\sim\mathcal{Z}}\left[\frac{\|Az\|1}{\|z\|_1}\middle|\|z\|_1\right]\leq\sum_{i=1}^K\|A_i\|_1\mathbb{E}_{z\sim\mathcal{Z}}\left[\frac{|z_i|}{\|z\|_1}\middle|\|z\|_1\right]=\frac{\|A\|_{1,1}}{K}$$

Finally, we can get an upper bound on $\mathbb{E}_{z\sim\mathcal{Z}}\left[\|Az\|_1\right]$:

$$\begin{aligned}\mathbb{E}_{z\sim\mathcal{Z}}\left[\|Az\|_1\right] &= \mathbb{E}_{z\sim\mathcal{Z}}\left[\mathbb{E}_{z\sim\mathcal{Z}}\left[\|A\frac{z}{\|z\|_1}\|_1\middle|\|z\|_1\right]\|z\|_1\right]\\ &\leq \mathbb{E}_{z\sim\mathcal{Z}}\left[\frac{\|A\|_{1,1}}{K}\|z\|_1\right]\\ &\leq \frac{\|A\|_{1,1}}{K}\mathbb{E}_{z\sim\mathcal{Z}}\left[\|z\|_1\right]\end{aligned}$$

This permits to control $\mathbb{E}_{z\sim\mathcal{Z}}\left[\delta_A(z)\right]$ with

$$\mathbb{E}_{z\sim\mathcal{Z}}\left[\delta_A(z)\right] = \mathbb{E}_{z\sim\mathcal{Z}}\left[\|Az\|_1 - \|z\|_1\right] \leq \frac{\|A\|_{1,1} - \|\boldsymbol{I}_K\|_{1,1}}{K}\mathbb{E}_{z\sim\mathcal{Z}}\left[\|z\|_1\right]$$

Then, for $A \in \mathcal{E}_\delta$, the ℓ_1-norm of the columns A_i is

$$\|A_i\|_1 \leq \sqrt{1-\delta^2} + \delta\sqrt{K-1}$$

We can derive an expression of $\frac{\|A\|_{1,1}-\|\boldsymbol{I}_K\|_{1,1}}{K}$ for $\delta \to 0$,

$$\begin{aligned}\frac{\|A\|_{1,1}}{K} - 1 = \frac{1}{K}\sum_{i=1}^{K}\|A_i\|_1 - 1 &\leq \sqrt{1-\delta^2} + \delta\sqrt{K-1} - 1\\ &\leq \delta\sqrt{K-1} - \frac{\delta^2}{2} + \underset{\delta\to 0}{\mathcal{O}}\left(\delta^4\right)\end{aligned} \tag{8.37}$$

□

8.7.2.4 Acceleration Conditions for Generic Dictionaries

Proposition 8.7.6. *For A with columns chosen greedily in $\mathcal{E}_{\delta,i}$ and for $v, z \in \mathbb{R}^K$,*

$$\begin{aligned}\mathbb{E}_D\left[\min_{A\subset\mathcal{E}_\delta}\left\|A^{-1}SA - B\right\|_F^2\|v\|_2^2 + \lambda\delta_A(z)\right] \leq\\ \frac{(K-1)K}{p}\|v\|_2^2 + \delta\sqrt{K-1}\left(\lambda\|z\|_1 - \sqrt{\frac{K(K-1)}{p}}\|v\|_2^2\right)\\ + \mathcal{O}\left(\delta^2\right)\end{aligned}$$

Proof. Let $A \subset \mathcal{E}_\delta$ be a unitary matrix with each column i chosen greedily in $\mathcal{E}_{\delta,i}$. Using results from Lemma 8.5 and Lemma 8.6, we show

$$\begin{aligned}
\mathbb{E}_D\left[\left\|A^{-1}SA - B\right\|_F^2 \|v\|_2^2 + \lambda\delta_A(z)\right] \leq & \|v\|_2^2 \left(\frac{K-1}{\sqrt{p}}\left(\frac{K}{\sqrt{p}} - 4\delta\sqrt{K}\right) + \mathcal{O}\left(\delta^2\right)\right) \\
& + \lambda\|z\|_1\left(\delta\sqrt{K-1} + \mathcal{O}\left(\delta^2\right)\right) \\
\leq & \frac{(K-1)K}{p}\|v\|_2^2 \\
& + \delta\sqrt{K-1}\left(\lambda\|z\|_1 - \sqrt{\frac{K(K-1)}{p}}\|v\|_2^2\right) \\
& + \mathcal{O}\left(\delta^2\right)
\end{aligned}$$

□

Theorem 8.7 (Acceleration certificate)**.** *In expectation over the generic dictionary D, the factorization algorithm using a diagonally dominant matrix $A \subset \mathcal{E}_\delta$, has better performance for iteration $q+1$ than the normal ISTA iteration – which uses the identity – to solve (8.1) when*

$$\lambda\mathbb{E}_z\left[\|z^{(q+1)}\|_1 + \|z^*\|_1\right] \leq \sqrt{\frac{K(K-1)}{p}}\mathbb{E}_z\left[\|z^{(q)} - z^*\|_2^2\right]$$

Proof. Let $A \subset \mathcal{E}_\delta$ be a unitary matrix with columns chosen greedily in $\mathcal{E}_{\delta,i}$. We start from Proposition 8.1,

$$\mathbb{E}_D\left[F(z^{(q+1)}) - F(z^*)\right] \leq \mathbb{E}_D\left[\|A^{-1}SA - B\|_F\|z^{(q)} - z^*\|_2^2 + \delta_A(z^*) - \delta_A(z^{(q+1)})\right]$$

Using the results from Proposition 8.7.6, with $v = z^{(q)} - z^*$, we can write

$$\begin{aligned}
\mathbb{E}_D\left[F(z^{(q+1)}) - F(z^*)\right] \leq & \frac{(K-1)K}{p}\|z^{(q)} - z^*\|_2^2 \\
& + \delta\sqrt{K-1}\underbrace{\left(\lambda\left(\|z\|_1 + \|z^*\|\right) - \sqrt{\frac{K(K-1)}{p}}\|z^{(q)} - z^*\|_2^2\right)}_{\leq 0} \\
& + \underset{\delta\to 0}{\mathcal{O}}\left(\delta^2\right) .
\end{aligned}$$

Taking the expectation over the input distribution of z, we get the desired result. □

Part III

Application to physiological signals

During my PhD, I collaborated with medical doctors for clinical research purposes, developing tools to help them analyze their physiological signal data. This collaboration has been centered around two projects, the study of the walk for adults and the study of the eye movements for young infants. This part presents results obtained as part of this collaboration.

9 Extracting Steps from Human Gait Signals

Contents

In clinical context, gait assessment is usually performed by visual examination. Extracting the information relevant to the doctors from inertial sensors would change the way patients are followed, as it would improve the comparison of their gait in time and with other patients. In this chapter, we present the gait signals collected with the Cognac-G group and show that convolutional representations – described in Chapter 3 – can be applied to these signals to extract step like patterns and to summarize the signals. Finally, we introduce a novel algorithm to identify steps in gait signal. This algorithm relies on template matching between the signals and a set of chosen steps.

9.1 Context

Pathologies affecting posture, balance, and gait control are threatening the autonomy of patients not to mention the risk of fall and therefore require rehabilitation intervention as early as possible. However, it remains difficult to accurately evaluate the various specific interventions during the rehabilitation process and the optimal content of exercise interventions they should involve. If only for these reasons, it would be interesting to learn how to monitor sensorimotor behavior at large and locomotion in particular which is a growing area in medical engineering science (Mariani, 2012; Marschollek et al., 2008; Willemsen et al., 1990; Dijkstra et al., 2008; Han et al., 2006; Ayachi et al., 2016; Williamson & Andrews, 2000). It requires several steps: first, we wish to investigate how

to monitor sensorimotor behaviors for patients in the doctor office and the resulting cognitive load it implies. Second, we want to learn how to construct databases with the quantitative variables recorded in that process, in order to make longitudinal studies of behaving individuals. Third, we would like to merge these individual databases in large data banks to define statistical norms, which is mandatory to detect dysfunctions or pathologies at the earliest stage possible. In that process we encounter at least three main challenges: the need for pervasive or ubiquitous computation to collect data; handling large inter-individual variability in the studied Human motion captures; and aggregating highly heterogeneous data to build the data bank.

There exist many software applications on the market that use wearable sensors – namely accelerometers, gyroscopes, magnetometers and/or GPS – and are useful for rehabilitations. They calculate the number of steps made in a day (Tran et al., 2012; Naqvi et al., 2012), the distance traveled in a day (Renaudin et al., 2012; Kim et al., 2004), the average speed or the daily amount of time spent walking, running, sitting, standing, laying (Oner et al., 2012; Brajdic & Harle, 2013). Most of the algorithms published in this context are either dedicated to one specific terminal or mobile phone, or they are copyrighted and not freely available for research.

This chapter is organized as follows: Section 9.2 describes the gait data used in this chapter. Section 9.3 presents the application of convolutional dictionary learning to these signals. Section 9.4 introduces a novel step detection algorithm, discusses the influence of the parameters and compares it to state-of-the-art methods. In Section 9.5, we briefly summarize a medical study, done with these signals and Section 9.6 concludes this chapter.

9.2 Gait Signals

9.2.1 What is a step ?

Locomotion is a hierarchical and complex phenomenon composed of different entities such as strides, steps, and phases (Auvinet et al., 2002; Mariani, 2012).

- Considering one foot, the stride is the succession of two phases: the *swing phase* (when the foot is in the air), and the *stance phase* (when the foot is in contact with the ground). The stance phase occurs between the heel-strike (moment when the foot hits the ground) and the toe-off (moment when the toes go off the ground), while the swing phase occurs between the toe-off and the next heel-strike.

- A *stride* is defined as the event that occurs between two heel-strikes of the same foot.

- A *step* is defined as the event that occurs between successive heel strikes of opposite feet. A step is therefore composed of two strides: one for the right foot, one for the left foot.

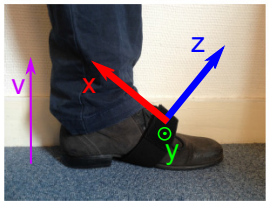
In the formal medical definition, a step is supposed to start when the heel strikes the ground and to finish somewhere in the end of the stance phase. It is not related to the foot activity since the foot is also moving in the swing phase. We choose in this chapter another definition: a step is defined in the following as the whole period of activity of a foot (when the foot is moving). The beginning of the step is defined as

Group	Number of exercises	Number of subjects	Sex (M/F)	Age (yr)	Height (cm)	Weight (kg)
Healthy subjects	242	52	35/17	36.4 (20.6)	173.4 (10.8)	70.7 (12.2)
Orthopedic diseases	243	53	26/27	60.1 (19.3)	169.2 (10.2)	77.4 (16.8)
Neurologic diseases	535	125	80/45	61.6 (13.2)	169.8 (8.7)	72.7 (15.5)
Total	1020	230	141/89	55.5 (19.6)	170.5 (9.7)	73.4 (15.3)

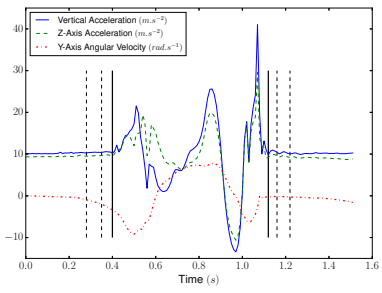
Table 9.1: Subjects' characteristics. For the age, height and weight, the mean and the standard deviations are displayed.

the heel-off (moment when the heel leaves the floor) and end of the step is defined as the foot-flat (moment when the foot is stabilized on the floor). This new definition allows considering the whole period of activity of a foot as a step, which makes it more adapted to step detection. Note that it does not change the number of steps and that it is easy to switch back to the medical definition once the heel-off and foot-flat instants have been detected.

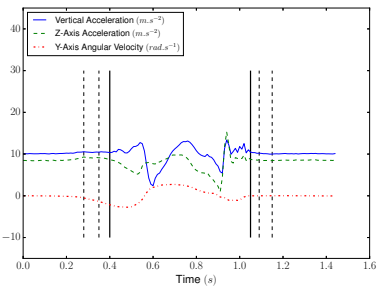
9.2.2 Data Acquisition and First Observations



(a) Definition of the axis for the XSens^{TM} sensor located at the left foot



(b) Healthy patient



(c) Hip affected patient

Figure 9.1: (a) XSens^{TM} sensor - (b,c) Vertical acceleration, Z-axis acceleration and the Y-axis angular velocity recorded from the right foot. The vertical lines display the different possibilities for start/end times.

The data used for the conception and testing of the method presented in this chapter has been provided by the following medical departments: Service de chirurgie orthopédique et de traumatologie de l'Hôpital Européen Georges Pompidou, Assistance Publique des Hôpitaux de Paris, Service de médecine physique et de réadaptation de l'Hôpital Fernand Widal, Assistance Publique des Hôpitaux de Paris, Service de neurologie de l'Hôpital d'Instruction des Armées du Val de Grâce, Service de Santé des Armées. The study was validated by a local ethic comity (Comité de Protection des Personnes Ile de France II, CPP 2014-10-04 RNI) and both patients and control subjects gave their written consent to participate. All signals have been acquired at 100 Hz with

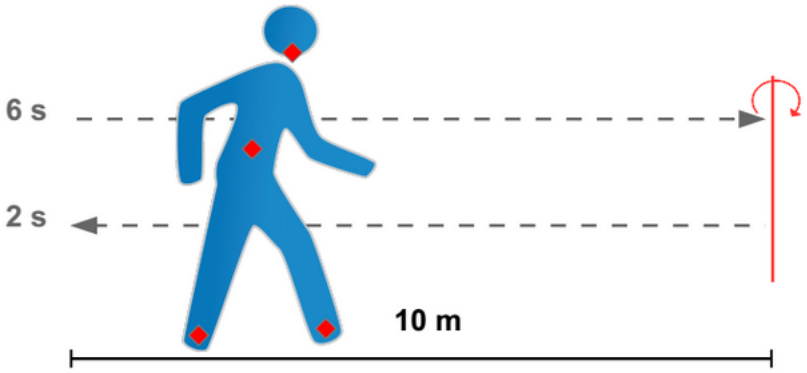


Figure 9.2: Clinical protocol for the walk study

wireless XSens MTw^{TM} sensors located at the right and left foot and fixed using a velcro band designed by XSens^{TM}. The signals obtained with both sensors were automatically synchronized by the acquisition software. All subjects were asked to:

1. stand quiet for 6 seconds
2. Walk 10 meters at preferred walking speed on a level surface
3. Make a U turn
4. Walk back
5. Stand quiet 2 seconds

Figure 9.2 illustrates this protocol. For practical reasons, patients kept their own shoes. The database is composed of 230 subjects who performed the protocol between 1 and 10 times, which leads to 1020 recordings. The subjects' characteristics are presented in Table 9.1. Healthy subjects had no known medical impairment. The orthopedic group is composed of 2 cohorts of distinct pathologies: lower limb osteoarthrosis and cruciate ligament injury. The neurologic group is composed of 4 cohorts: hemispheric stroke, Parkinson's disease, toxic peripheral neuropathy and radiation induced leukoencephalopathy.

The protocol includes 2 sensors (left and right foot), and each of them records a 9-dimensional signal (3D accelerations, 3D angular velocities, 3D magnetic fields), possibly with some recalibrated data provided by the XSens^{TM} software (such as the vertical acceleration in the direction of the gravity). Instead of considering all these dimensions, we decided to only use a subset of them, and select the most relevant in the context of step detection. This decision has been made based on observations of real data and physiological reasons provided by doctors. We decided to select only the components that are the most reflective of the locomotion process (see Figure 9.1a for the definition of the axis): the Z-axis acceleration, the recalibrated vertical acceleration (vertical movements of the foot) and the Y-axis angular velocity (swing in the direction of the walk). We expect these components to strongly react to the steps, making them identifiable.

Figure 9.3: Vertical acceleration of the right foot for an healthy subject walking. The vertical dashed line represent the beginning of the steps, annotated by a medical doctor.

Examples of these 3 components (Z-axis acceleration, vertical acceleration and Y-axis angular velocity) recorded for the right foot are presented on Figure 9.1b and Figure 9.1c for respectively a healthy and hip-injured patient. It appears on these figures that the amplitudes of the signals are clearly different and it is likely that classical threshold-based methods would hardly perform well on both subjects. However, the structure and shape of the step is roughly the same for both subjects so it might be relevant to use a template-based method. Nevertheless, these examples also display the main difficulties in conceiving an automatic algorithm for step detection:

- The uncertainties in the definition of the starts and ends of the steps. Indeed, we can see on Figure 9.1b, that many choices would be acceptable: depending on the considered definition, the results may be different.
- The variability of the step patterns according to the pathology, the age, the weight, etc. For example, on Figure 9.1c, the subject is dragging his feet, causing an abrupt change in the step pattern (noisy part at the end of the step).

9.3 Convolutional Representations for Gait Signals

In this section, we use the convolutional representation, described in Chapter 3, to test the capacity of the model to extract step patterns. We focus on the vertical acceleration of the right foot. An example of a vertical acceleration signal during walking is given in Figure 9.3. All the experiments are run using a basis of 100 recordings of healthy subjects and split between a train set of 50 recordings denoted X_{train} and a test set X_{test} with 50 recordings from healthy subjects that are not in the train set.

9.3.1 Encoding a Signal with a Dictionary of Steps

To test the encoding capacity of the convolutional sparse coding, we computed the embedding of signals from human walking on a dictionary $\boldsymbol{D}^m$ of steps. To construct this dictionary, we select 25 recordings in X_{train} where the steps are annotated manually by a medical doctor and draw one step uniformly in each of these signals. These patterns are them normalized and zero-padded such that p is the k-th pattern in $\mathcal{P}$, *i.e.* the vertical acceleration of the step selected in the k-th recording, then

$$\boldsymbol{D}_k^m[t] = \begin{cases} \frac{p[t]}{\|p\|_2} & \forall t \in [\![0, |p| - 1]\!] \\ 0 & \forall t \in [\![|p|, W - 1]\!] \end{cases}$$

with $W = \max_{p \in \mathcal{P}} |p|$. The blue curves in Figure 9.4 present these step patterns used in $\boldsymbol{D}^m$. Then, the activation for the signals in X_{test} are computed using SeqDICOD, presented in Algorithm 5.2, with $M = 5$. We fixed the regularization parameter to $\lambda = 5$ for this experiment.

Figure 9.5 presents the activation signal computed for one signal X. Unsurprisingly, the activation coefficients are concentrated around the beginning of the steps. The activated coefficients are not unique for each step, but the steps are a linear combination

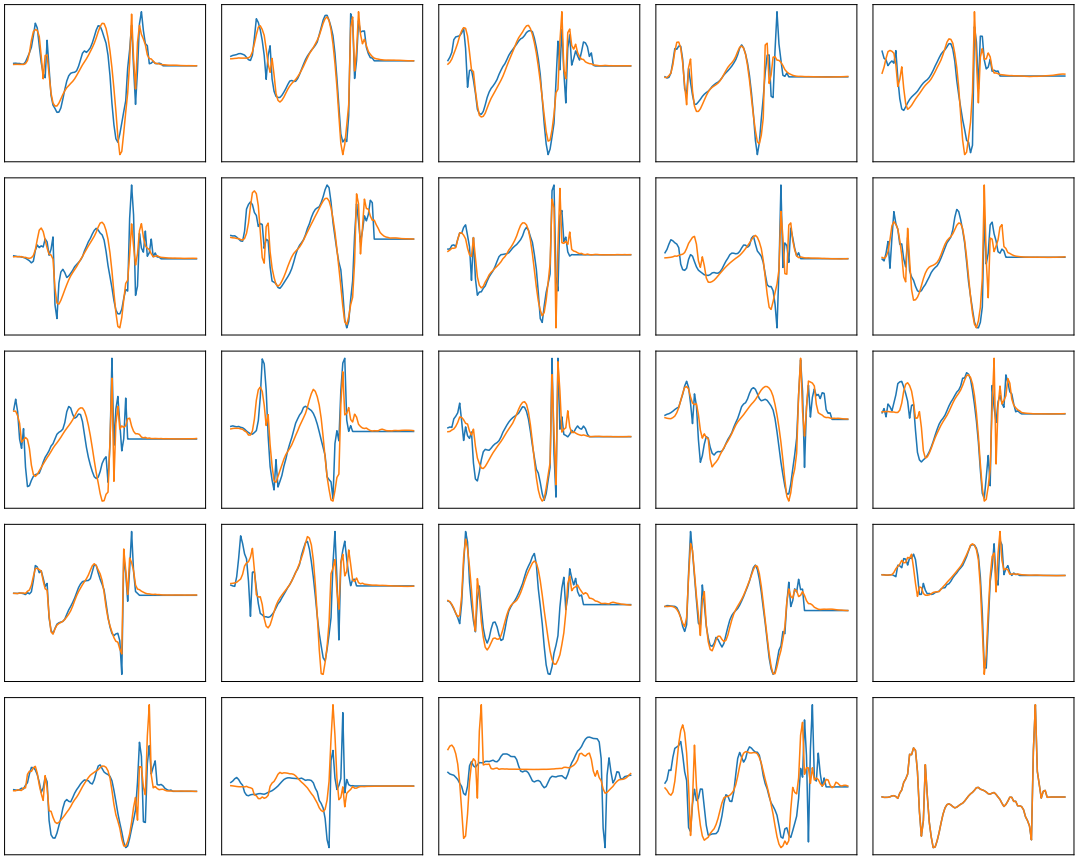


Figure 9.4: (*blue*) Initial steps in $\boldsymbol{D}^m$. (*orange*) Patterns in $\boldsymbol{D}^{(50)}$ learned with convolutional dictionary learning on a set of 50 walk exercises with healthy subjects, starting from the steps patterns $\boldsymbol{D}^m$.

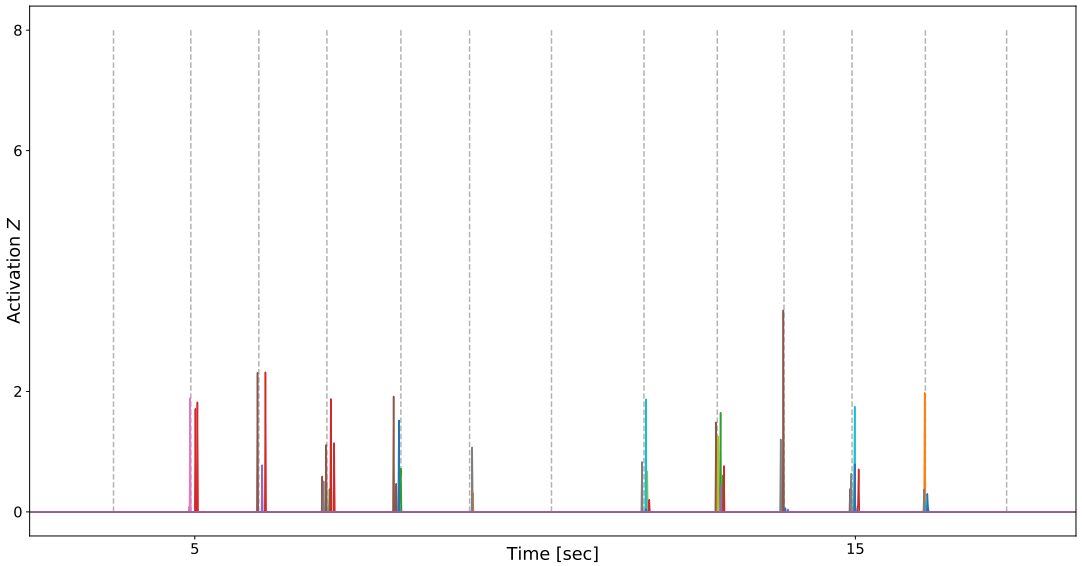


Figure 9.5: Convolutional sparse coding for the right foot vertical acceleration signal with the dictionary $\boldsymbol{D}^m$ composed of steps randomly selected from different subjects' exercises. The vertical dashed lines represent the beginning of each step, annotated by a medical doctor in the original signal.

of multiple patterns, sometimes slightly shifted in time.The slight shifts in time result from annotation imprecision for the steps patterns. Indeed, from the signal, there is no clear definition of the start of the step in the signal and the uncertainty of this annotation is evaluated to less than 0.2 s (20 samples) for each step. This explains that the coefficients might be spread out around the step boundaries. To be able to use convolutional sparse coding as a step detection algorithm, an extra pattern alignment step would be required. No pattern seems to be dominant to encode the signal as all patterns are activated for some signals. In Figure 9.5, three steps are not encoded by the method: the steps at the beginning and at the end of the walk and the central step. The first and last steps play a specific role in the recording as they capture the transition between the stand still phase and the walking phase. In our specific protocol, the central step is associated with the about turn of the exercise (*cf* Figure 9.2). These three steps have specific shapes and amplitudes and are different from other steps. The dictionary $\boldsymbol{D}^m$ being drawn uniformly, it contains steps from the edge and turn-about but as they are more specific, the convolutional sparse coding is not able to summarize the three missed steps using the given step patterns. Another issue is that the amplitudes of these three steps are smaller than the amplitude of other steps. Lower amplitude steps are not captured by the convolutional sparse coding because encoding them does not reduce the reconstruction cost as much as for the other steps. The tradeoff between the reconstruction cost and the sparse regularization is controlled by the regularization parameter λ which is fixed for the whole signal X. If λ is lowered, convolutional sparse coding encodes these specific steps but also add more coefficients for the other steps. This example illustrates an open problem for sparse coding: how can convolutional sparse coding capture local variations in the signals adaptively to the local amplitude. For step detection, we designed a robust algorithm based on the Pearson coefficient to detect steps in recordings of human walking which handle theses problems with the amplitude (see Section 9.4 and Appendix A). This algorithm can be seen as a greedy sparse coding algorithm with an amplitude normalization.

9.3.2 Updating the Dictionary for a Set of Signals

Then, we use convolutional dictionary learning (CDL) to update the dictionary $\boldsymbol{D}^m$ with X_{train}. The signal used for the unsupervised learning of the dictionary $\boldsymbol{D}^{(50)}$ are thus the 25 recordings from which the original steps in $\boldsymbol{D}^m$ are extracted plus 25 extra recordings. The patterns are updated with 50 iterations of alternate minimization with SeqDICOD$_5$ for sparse coding steps and accelerated proximal gradient descent (APGD) to update the dictionary elements (see Subsection 3.4.1 and Algorithm 3.9 for details on APGD).

Figure 9.6: Evolution of the cost function for the train set relatively of the number of CDL iteration q run. (*dashed*) Cost for dictionary $\boldsymbol{D}^m$. (*blue*) Cost for the CDL starting from $\boldsymbol{D}^m$. (*orange*) Cost for the CDL starting with the SSA initialization. (*green*) Cost for the CDL with 5 random initializations.

Figure 9.6 shows the evolution of the cost function for the train set relatively of the number of convolutional dictionary learning iteration q run. The CDL improves the encoding cost on the train set compared to the initial dictionary of steps $\boldsymbol{D}^m$. This procedure also improves the cost function value on the test set. Figure 9.4 presents in orange the patterns obtained after the dictionary learning, ordered using the ℓ_1-norm of

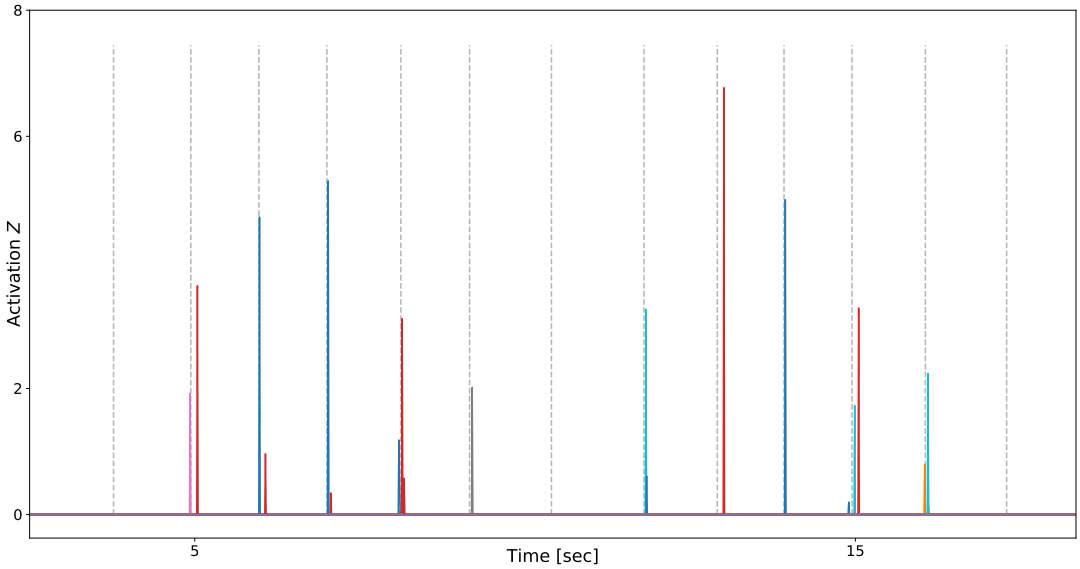


Figure 9.7: Convolutional sparse coding for the right foot vertical acceleration signal with a dictionary of steps randomly extracted from other patients exercises.

the associated coding signals on the test set. Thus, the first pattern is the one which is used the most to encode the test signals. The first observation is that the final patterns are not very different from the original steps and the local structure of the steps is preserved. The original steps are smoothed in their final version as it can be seen with the first pattern where the final oscillations are attenuated and the central part is more regular. Some other local variations are amplified. For instance the vibration at the beginning of the sixteenth learned pattern is stronger than in the original step. Finally, the last pattern is equal to the original ones. This pattern is not updated because it is not used to encode any part of the train sample signals.

The patterns learned with CDL improve the signal representation computed with this model. Figure 9.7 illustrates the coding signal obtained with this new dictionary for the same signal X as in Figure 9.5. Compared to the codes obtained with $\boldsymbol{D}^m$, the activated coefficients are more concentrated around the steps and fewer coefficients are activated. The signal is summarized in a sparser way using the new dictionary. However, no pattern is adapted by the algorithm to capture the three steps that were missed using $\boldsymbol{D}^m$. This can be explained by the amplitude issue discussed in Subsection 9.3.1. Indeed, these type of steps are not encoded by the sparse coding step as for the given regularization, the coding coefficients are estimated to be null in these areas. Thus, the dictionary updates cannot adapt the patterns to better capture these local variations and this part of the signal is ignored when updating the dictionary elements. This behavior is linked to the regularization level adaption problem for convolutional sparse coding. The design of an algorithm which could adapt the regularization level to the local amplitude of the signal, up to certain levels, would improve the usage of this technique for time series representation.

Figure 9.8: (*top*) Examples of generic dictionary patterns in $\boldsymbol{D}^{(0)}$, drawn from (9.1). (*bottom*) Top five patterns learn with 50 iterations of CDL. These patterns captures the step local structures.

9.3.3 Learning a Dictionary of Steps from Accelerometer Signals

In the previous experiments, the structure of the steps is given in the initial dictionary $\boldsymbol{D}^m$. An interesting question is to see if the CDL is able to learn the local structure of the steps without supervision, *i.e.* starting from an initial dictionary computed automatically. Here, we will compare two initialization approaches to learn a set of patterns $\boldsymbol{D}$ with 50 iterations of CDL: the random initialization with a generic dictionary and the initialization with the PCA.

For the random initialization, we draw uniformly generic atoms in the unit sphere of $\mathbb{R}^W$. The elements $\boldsymbol{D}_k^{(0)}$ of the dictionary are generated using a normalized Gaussian distribution, such that

$$\boldsymbol{D}_k^{(0)} = \frac{u_k}{\|u_k\|_2} \quad \text{with } u_k \sim \mathcal{N}_W\left(0, \boldsymbol{I}_W\right) \tag{9.1}$$

The top part of Figure 9.8 presents 5 examples of generic atoms. These atoms do not have any structure, contrarily to the initial dictionary used in the previous experiment. The CDL alternate minimization is used, starting with this random dictionary, with a set X_{train} of 50 recordings of healthy patients walking. The final dictionary elements are displayed in the bottom part of Figure 9.8. Out of the 25 learned patterns, only six are used to encode the signals in X_{train}. The other ones are not updated as they are never activated. The six patterns learned are similar to step patterns from Figure 9.4. The general shape of the patterns are very similar. The main differences are located at the end of the steps, when the foot is touching the ground. Also, the patterns are not aligned. The first pattern starts with a small phase delay and is faster compared to the second one. This experiment shows that convolutional dictionary learning is able to learn patterns which capture the local dynamic of a step in an unsupervised setting starting from random initialization.

For signal encoding, these patterns are also able to summarize the walk in interpretable way. Figure 9.9 displays the encoding of the same walk signal presented in Figure 9.3 and encoded in the previous figures with the patterns learned with CDL from generic initialization. The activation coefficients are also concentrated around the beginning of the steps but there is a bit more variations. Indeed, as no annotations were given at the beginning, some patterns such as the 4th and 5th ones capture a part of either one (the 4th) or two steps (the 5th). These random shifts are selected by the initialization of the dictionaries and the delays between the localization of the components are linked to the delay observed in the learned patterns. A nice way to resolve this would be to design a dictionary updates which tries to compute aligned dictionaries by trying to regroup the activated coefficients at the same time in the activation signal. Indeed, a dictionary and its associated activated coefficient can both be shifted by an inverse lag τ without changing the value of the reconstruction cost. With this idea, the patterns with a zero part such as the 4th one could be shifted back to learn more interesting part of the signal during the dictionary update.

Figure 9.9: Activation signal for the right foot vertical acceleration signal with a dictionary learned from a set of 50 exercises by healthy patient. Each color corresponds to one activation signal.

Figure 9.10: (*top*) Examples of initial patterns in the dictionary $\boldsymbol{D}^{(0)}$ computed using SSA. (*bottom*) Top five patterns learn with 50 iterations of CDL. These patterns captures the step local structures.

Figure 9.11: Activation signal for the right foot vertical acceleration signal with a dictionary learned from a set of 50 exercises by healthy patient. (*red*) first pattern, (*orange*) second pattern.

A second experiment makes use of the SSA to initialize the patterns. The idea is to make a PCA on all subseries of length $W = 100$ in the train set. The initial dictionary $D^{(0)}$ is constructed by taking the K first singular vectors (or principal components). The top part of Figure 9.10 presents 5 examples of generic atoms. These atoms do not have any structure, in contrary with the initial dictionary used in the previous experiment. The CDL alternate minimization is used, starting with this random dictionary, with a set X_{train} of 50 recordings of healthy patients walking. The final dictionary elements are displayed in the bottom part of Figure 9.10. In this case, the learned patterns can also be linked to step patterns. The same remarks can be made as for the random initialization. Out of the 25 learn patterns, only 15 are used to encode the signals from X_{train} and 10 are not updated from their original value. The patterns are not aligned and some of them capture the end of a step and the beginning of another one. Figure 9.11 presents the encoding of the same signal presented before. The encoding is less concentrated compared to random initialization. Indeed, the patterns are selected to capture the variance in all shifted position to they are not well localized with SSA. But this method is also the only one to be able to capture the last step patterns in the signal. The first and middle patterns are not encoded by any of the learned dictionary we tried.

As the CDL problem is non-convex, the initialization plays a tremendous role in the capacity of the method to compute a good representation of the signals in the basis. To assess the impact of the initialization, we look at the function error for X_{train} and X_{test}. Figure 9.6 displays the training cost function evolution during the CDL for initialization with the dictionary of manually annotated patterns $\boldsymbol{D}^m$, with the SSA and with 5 random dictionary drawn from (9.1). We can see that the manual initialization gives better results than all the other initialization methods. But with the number of iteration growing, CDL obtains dictionaries which have similar error level as the initial dictionary $\boldsymbol{D}^m$. The dictionary obtained with CDL starting from $\boldsymbol{D}^m$ beats all the other for the other dictionaries. However, the error levels are not very different and random initialization gives results that can be compared to manual initialization. The same observations can be done with the test error, displayed in Table 9.2 and the order of the different methods is preserved. The best test error is obtained for manual initialization with CDL but random and SSA initialization are able to reach levels lower than the initial manual dictionary after few iterations. The SSA initialization does not seem to provide a big advantage compared to the random initialization but it is better than the

Initialization	Manual	Manual	SSA	best random	mean on 5 random initialization
# iteration	0	50	50	50	50
Train Error	0.1882 (0.08)	0.1802 (0.07)	0.1832 (0.07)	0.1820 (0.07)	0.1837 (0.07)
Test Error	0.1806 (0.05)	0.1777 (0.05)	0.1791 (0.05)	0.1786 (0.05)	0.1792 (0.05)

Table 9.2: Value of the cost function for X_{train} and X_{test} for various initialization and number of iterations. The last column is the performance averaged over 5 random iterations.

mean random initialization.

Finally, one parameter which is hard to fix is the regularization parameter. Indeed, this parameter controls the sensitiviy of the method and the quality of the extracted patterns. To see the effect of this parameter, Figure 9.12 displays the ratio between the cost function value on X_{train} for dictionaries learned with CDL and the one obtained with a generic dictionary drawn at random relatively to the regularization parameter value λ. We clearly see on this plot that there is a trade off for this parameter. If it is set too high, a lot of information in the signal is discarded as noise and possibly no patterns are learned. If it is too low, the learned patterns are not be informative as the different set of patterns have the same reconstruction error. Around the optimal λ, there is little difference between the manual initialization and unsupervised strategies but as λ gets lower, the manual strategy seems to lead to a better dictionary.

Figure 9.12: Ratio between final training cost for a dictionary learned with CDL ; from an initialization (*blue*) manual to $\boldsymbol{D}^m$, (*orange*) with SSA and (*green*) with a generic dictionary; compared to the training cost for a random dictionary as a function of the regularization parameter λ.

These preliminary results are very encouraging on the usefulness of convolutional representation method to capture the same information as the step detection in walk signals. Indeed, the activation coefficients indicate the approximate localization of the steps. An extra step of pattern alignment is necessary to get a unique coefficient localized in time. This concentration of the coefficients could be obtained using group sparsity technique which would penalize activated coefficients that are close to be grouped on the same time step.

9.4 Robust Step Detection

This section quickly describes an algorithm developed using our signal basis to robustly detect the steps in humane walk signals. The full study can be found in Appendix A.

In the context of dynamic equilibrium quantification, it is important to be able to robustly extract the steps from inertial sensor recording of a human walking. In our study, we present a method for step detection from accelerometer signals based on template matching. Due to the constraints from the considered medical application, our algorithm has not been directly developed using the convolutional sparse coding as this method does not robustly detect the steps with various amplitude from our

data base. The principle of our step detection algorithm is to recognize the start and end times of the steps in the signal thanks to a predefined set of templates (library of steps). The algorithm is tested on a database of 1020 recordings, composed of healthy patients and patients with various neurological or orthopedic troubles. Simulations on more than 40000 steps show that even with a library of only 5 templates, our method achieves remarkable results with a 98% recall and a 98% precision. The method is robust to parameter changes, adapts well to pathological subjects and can be used in a medical context for robust step estimation and gait characterization.

9.5 An Automated Recording Method in Clinical Consultation to Rate the Limp in Lower Limb Osteoarthritis

This section quickly describes a study performed using our signal basis to rate the Limp in Lower Limb Osteoarthritis. The full study can be found in Appendix B.

For diagnosis and follow up, it is important to be able to quantify limp in an objective, and precise way adapted to daily clinical consultation. The purpose of this exploratory study was to determine if an inertial sensor-based method could provide simple features that correlate with the severity of lower limb osteoarthritis evaluated by the WOMAC index without the use of step detection in the signal processing. Forty-eight patients with lower limb osteoarthritis formed two severity groups separated by the median of the WOMAC index (G1, G2). Twelve asymptomatic age-matched control subjects formed the control group (G0). Subjects were asked to walk straight 10 meters forward and 10 meters back at self-selected walking speeds with inertial measurement units (IMU) (3-D accelerometers, 3-D gyroscopes and 3-D magnetometers) attached on the head, the lower back (L3-L4) and both feet. Sixty parameters corresponding to the mean and the root mean square (RMS) of the recorded signals on the various sensors (head, lower back and feet), in the various axes, in the various frames were computed. Parameters were defined as discriminating when they showed statistical differences between the three groups. In total, four parameters were found discriminating: mean and RMS of the norm of the acceleration in the horizontal plane for contralateral and ipsilateral foot in the doctor's office frame. No discriminating parameter was found on the head or the lower back. No discriminating parameter was found in the sensor linked frames. This study showed that two IMUs placed on both feet and a step detection free signal processing method could be an objective and quantitative complement to the clinical examination of the physician in everyday practice. Our method provides new automatically computed parameters that could be used for the comprehension of lower limb osteoarthritis. It may not only be used in medical consultation to score patients but also to monitor the evolution of their clinical syndrome during and after rehabilitation. Finally, it paves the way for the quantification of gait in other fields such as neurology and for monitoring the gait at a patient's home.

9.6 Conclusion

In this chapter, we showed that sparse convolutional representation can be used to summarize walk signals. A library of patterns can be used to encode the signal and by optimizing it, it is possible to get information about the step localization. In addition, steps patterns can be learned from a set of signals in an unsupervised setting, capturing

the common shape of the patterns in the different exercises for different initialization strategies.

Despise these promising initial results, certain questions should be addressed to improve the practicality of this method. The question of the local adaption of the regularization to capture patterns with different amplitude in the signal is critical to enable the usage of such technique on non-segmented signals, with different intensity due to variation in the speed of the walk or to different phases in the protocol. Another question is the cleaning of the obtained representations. The different patterns shifted, with activation coefficient localized at different time and the grouping of such coefficients would greatly improve the interpretability of the method and its results. Finally, the question of pattern balance is also an open problem. Indeed, some patterns are less frequent than the other, such as the boundaries steps or the steps performed during a turn-about. The capacity to learn such patterns greatly depends on the capacity to learn patterns that are under-represented in the signal.

We also described in this chapter a template-based method for step detection. This method, based on a greedy algorithm and a library of annotated step templates, achieves good and robust performances even with a small number of templates. When used on a large database composed of healthy and pathological subjects walking at different speeds, the method obtains a 98% recall and 98% precision. This method shows that it is possible to improve the pattern detection for specific application but the automatization of such process would be of tremendous interest for many applications.

ACKNOWLEDGMENTs

The authors would like to thank R. Barrois, D. Ricard, A. Yelnik, C. De Waele and T. Grégory for the thorough discussions, the design of the experiment, the data acquisition and clinical annotation. This work was supported by SATT Ile-de-France Innov.

10 Recording Eye Movements in Young Children

Contents

The eye movements result from a set of complex interactions between different parts of the nervous system. Indeed, the eyes move to adapt to what they look at but also to stabilize the vision during movements, or to react to other inputs. Through the study of eye movements, it is thus possible to understand the normal and pathological behavior of many other parts in the human body. The development of cheap sensors to record eye movements opened the field to new opportunities to quantify these movements.

In this chapter, we focus on a particular eye movement, the nystagmus. This movement – which can hinder the vision of young infants – is not clearly understood by ophthalmologists. In this chapter, we suggest using the SSA to remove the normal movement of the eye – resulting from the gaze movement. With this method, it is possible to study a cleaner movement resulting from the nystagmus. This method is then used in a study of the nystagmus associated to optic pathway gliomas.

10.1 Context

The eye movements are the set of voluntary or involuntary movements of the eyes, used to accommodate the vision. They are involved in many visual tasks, like target-tracking, reading or stabilizing the gaze during movement. The control mechanisms involved are complex and come from different parts of the nervous system. The study of these movements and their pathologies helps researchers better understand these mechanisms.

The nystagmus is a specific movement of the eyes, which is normally observed while the head is rotating. With a coupling from the vestibular area, the eyes move in the opposite direction slowly – matching the rotation speed – and then perform a saccadic movement – or high speed movement – in the sense of rotation. This movement stabilizes distant images while the head moves, by keeping the eye in a fix direction long enough to see

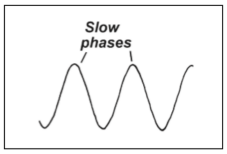
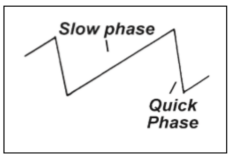
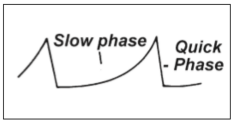
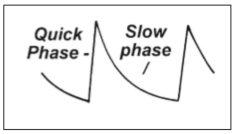
Pendular	**Jerk**: slow phase+saccade		
Only slow phases	Constant velocity	Increasing velocity	Decreasing velocity
			
• Spasmus Nutans (SN) • Infantile Nystagmus Syndrome (INS)	• Certain neurological nystagmus • Optokinetic nystagmus	• Certain neurological nystagmus • Infantile Nystagmus Syndrome (INS)	• Certain neurological nystagmus • Fusional Maldevelopment Nystagmus Syndrome (FMNS)

Table 10.1: Classification of the nystagmus shapes

the scene clearly. This type of nystagmus – associated to the head rotation – is called the optokinetic nystagmus.

In some cases, unwanted nystagmus movements develop in infancy – and sometime later in life – with the infant's eyes constantly moving, and his visual perception is hindered. The reasons behind the appearance of such movements are not clear but this condition can be associated to congenital disorders, central nervous system disorders or retinal dysfunction. The causes of the nystagmus are linked to the characteristics of the movements, like their frequency, their shapes and their variations when the gaze moves. The general principles to classify nystagmus are given in Table 10.1. The focus is on the presence of slow phases, with varying velocity, or saccades. These movements do not result from the same controlling mechanism and it is important to distinguish them. Then, inside each class, the type of movement can also be sub-classified. The eye movements in INS can take specific shapes which are sorted in different categories, as described in Figure 10.1.

If some classes of nystagmus, like the INS, have been largely studied, we have less information about other classes of nystagmus. The Spasmus Nutans (SN) is difficult to study as it is a condition which appears early in the childhood and can then disappear or change into another class of nystagmus. This forces researchers to work with very young children, from 3 months to 10 years old. The design of adequate protocols to record the infant eye movements in various settings is challenging in itself. As this condition is not very well-known, it is often misdiagnosed and treated as another type of nystagmus. In collaboration with Matthieu Robert, from the Hôpital universitaire Necker-Enfants malades, we designed a protocol to record and study eye movement from early childhood. The stimuli were adapted to get the children attention using cartoons and we used a device specially designed for young infants. We used the Eyefant sensor, developed with Ober consulting (Poland), which tracks the binocular eye movements using infrared photoreflectometry with a sampling rate of 1000Hz. The sensor was designed to be very light, in order not to affect the behavior of the child during recording.

The aim of these recordings is to quantify the eye movement properties. With the extracted information, the medical doctor is able to better understand the nystagmus and to classify it in the right category, from the previous classification. Indeed, characteristics like the frequency or the shape are easy to infer on specific signals. But

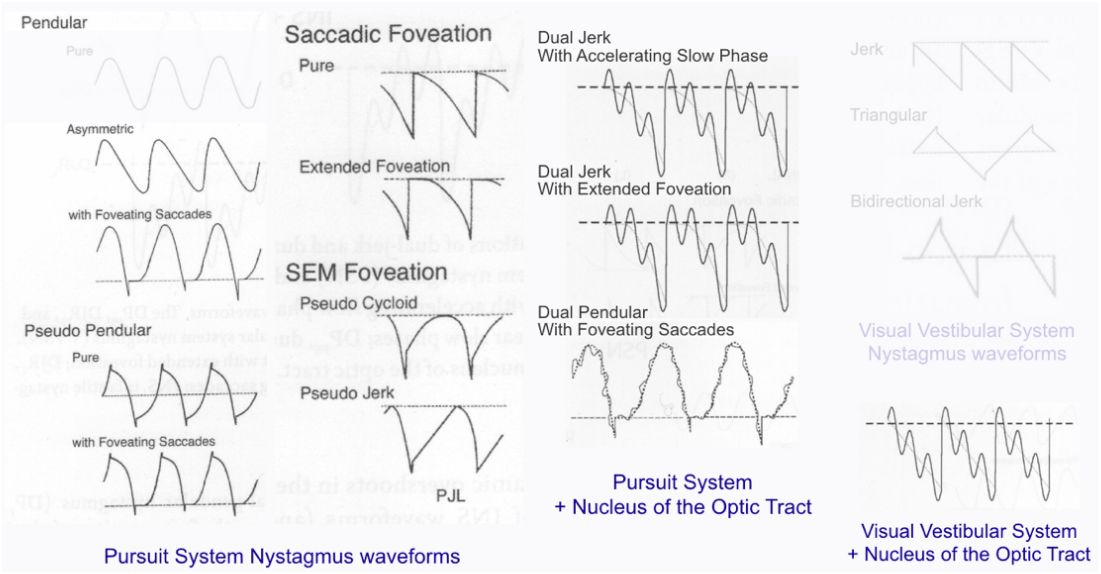


Figure 10.1: Sixteen possible waveforms in INS, among which 12 pathognomonic of INS (adapted from Hertle & Dell'Osso 2013).

these quantified characteristics also highlight some behaviors not well documented in the nystagmus literature.

The rest of this chapter is organized as follows. In Section 10.2, we describe the tools used to study the recorded nystagmus movements. These tools have been used for two clinical studies around the nystagmus, one of which is briefly described in Section 10.3. Section 10.4 concludes this chapter with some remarks on this clinical research collaboration.

10.2 Extracting Movement Properties with SSA

The Eyefant recorder needs a calibration phase to correctly identify the horizontal and vertical axis. This calibration relies on a sequence, where the recorded subject needs to look at green dots moving on the screen. This is not possible with infants in early childhood. The calibration is thus made *a posteriori*, using the movements of the eyes doing saccades from a central cue to four to eight eccentric locations. Saccades are provoked using a moving cartoon and this approach gives good results in practice.

To study the nystagmus, a detrending step is necessary to separate the wide eye movements – provoked by the gaze movement – from the ones associated to the nystagmus. The eyes are not looking in a fix position but scanning the visual field, with slow movement, and reaching some precise target point with saccadic movement. These two types of movement are produced without a specific structure and form the trend of the series. The movements linked to the nystagmus are more structured, in the sense that they are repeated and have a characteristic shape.

In this context, the usage of the Singular Spectrum Analysis (SSA) reliably removes the trend and facilitates the characterization of the signal. This technique is efficient to decompose signals composed of a trend and periodic components. We propose to use it to determine the gaze movement as the trend of the series and substract it from the signal to recover the nystagmus movement. As mentioned in Chapter 4, the quality

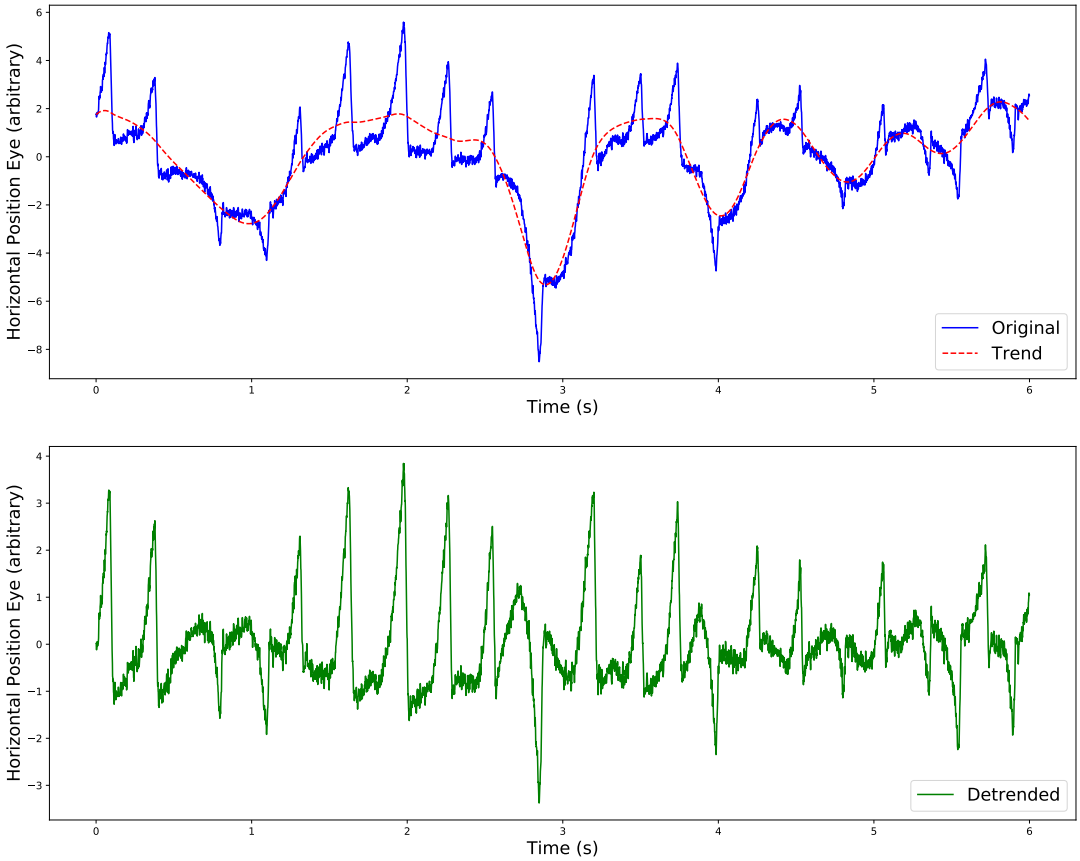


Figure 10.2: (*top*) Oculographic signal and its estimated trend. (*bottom*) Detrended signal. The SSA preserves the characteristic local patterns but discards the large variations. The result is visually satisfying, with minor overshoot, for instance on the large saccade between 2.5s and 3s.

of the components is tied to the grouping strategy chosen. The experimental results show that (**GG3**)-(HM) is a good method to estimate precisely the trend of the signals. The critical parameter for this method is the window length W. If too small, all the variations are included in the trend and if too big, the trend is not estimated correctly, because it is spread on too many components. A good choice is $W = 500$, as it gives a resolution high enough (around 0.5Hz) to correctly separate the trend and it is not too large for the computation to be unmanageable but the trend is correctly estimated.

The signals for the nystagmus movement are then analyzed using traditional signal processing tools. We estimate for each eye the main frequency of the movement using the maximum of the correlogram. Then to compute the phase delay of the movements of the eyes, we use the same principle based on the maximum of the cross-correlation function between the two movement signals. We use this processing tools with sliding windows selection of the signal in order to highlights the temporal variations of the frequency and delays.

The results are presented using two visualizations. The first one uses heatmaps to present the results relatively to the direction of the gaze. Figure 10.3 presents such representations for four characteristics of the nystagmus. This representation highlights the calm zone that the infant has in the right gaze. One possible treatment for the

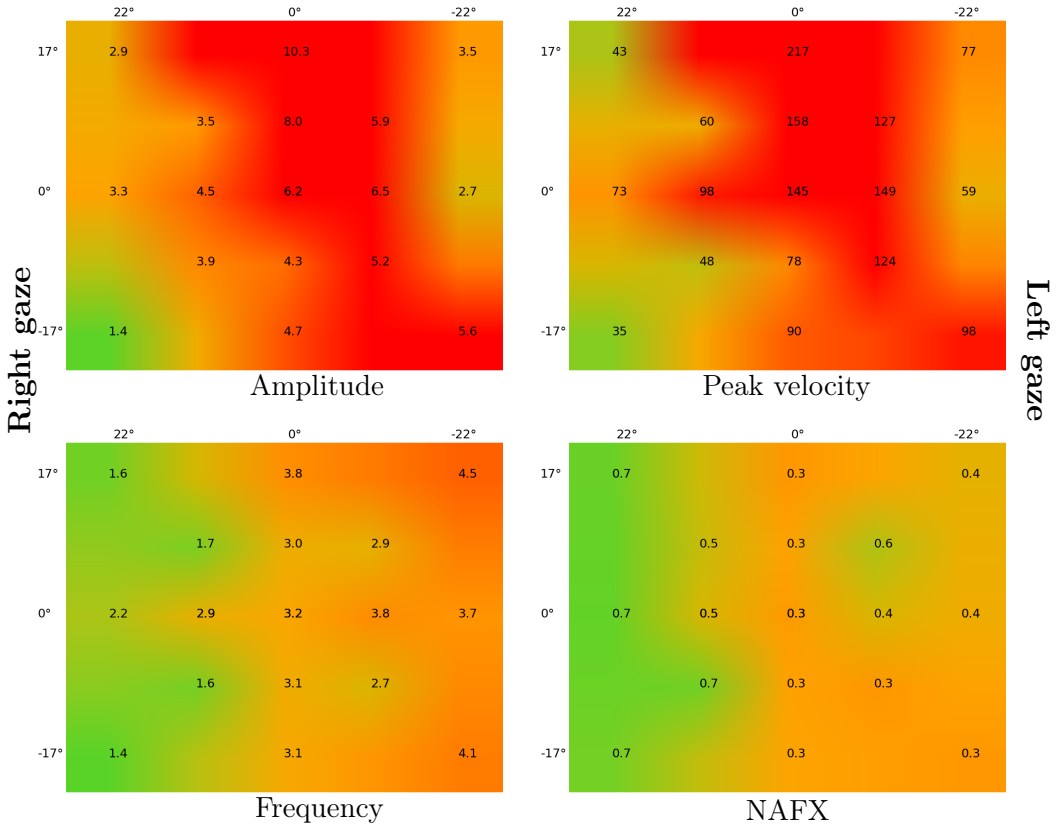


Figure 10.3: Spatial presentation of the nystagmus characteristics: (*from top-left to bottom-right*) Amplitude, peak velocity, frequency and NAFX). These maps highlight a calm zone for the nystagmus in the right gaze.

nystagmus is to perform a surgery to move the eye, such that this calm zone is located in the central gaze of the patient. This heatmap could be used to help the ophthalmologist decide of such interventions and then follow the evolution of the nystagmus in the different gaze directions.

The second visualization, presented in Figure 10.4, presents the delay between the two eyes as a function of time. This visualization is important as it indicates to the ophthalmologist the variation in the phase between the two eyes. On the signal presented, it can be seen that the two eyes are moving with a phase shift of around 180 degrees most of the time. But in some parts of the signal, the movements of both eyes are in-phase again. This behavior is typical of a Spasmus Nutans and it can only be seen by studying this quantity relatively to time. This type of visualization helps the practitioner understand the characteristics of the eye movement.

10.3 Nystagmus Associated to Optic Pathway Gliomas

This section quickly describes the Optic pathway gliomas-associated nystagmus study which can be found in Appendix C.

Nystagmus associated with optic pathway gliomas (OPG) are of crucial interest, both from a clinical point of view –they are possibly the only type of nystagmus in infants to lead to the diagnosis of a potentially lethal tumor– and from a theoretical perspective.

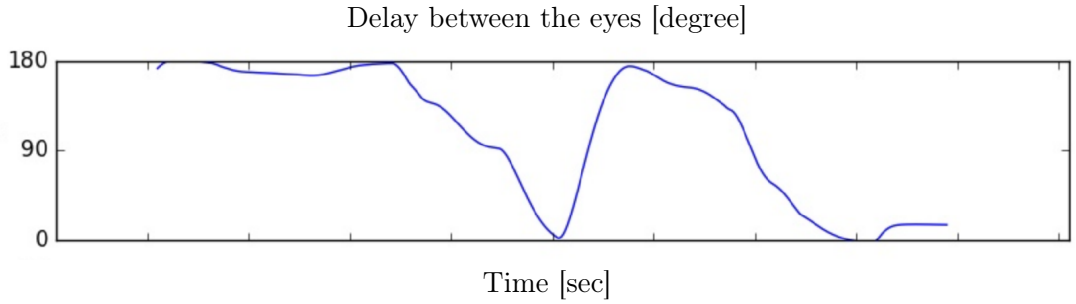


Figure 10.4: (*top*) Delay between the eyes movement (*bottom*) Frequency of each eye. The temporal representation shows that while the frequency is very stable, around 2Hz, the delay between the eyes can vary. This is typical of the SN nystagmus.

However, the precise characteristics and mechanism of OPG-associated gliomas remain unknown. Cases of nystagmus having led to a diagnosis of OPG seen in the pediatric neuro-ophthalmology clinic of a tertiary referral hospital between 2009 and 2014 were identified. Cases with reproducible nystagmus recordings available (video recordings and/or infrared photoreflectometry using the Eyefant*, Ober consulting, Poland) were included. Eleven cases were identified; eight were included. Age at nystagmus onset was 2.5-10 months (mean=5.8, median=5.5, SD=2.4). The associated OPG always involved the chiasm, always exhibited post-gadolinium enhancement –either peripheral or global–; in seven cases the OPG was big (from 28x28x20 to 51x30x47mm); in two cases it was metastatic. Clinically, the nystagmus was always classified as Spasmus Nutans (SN) type (always pendular, medium or high-frequency, low-amplitude, possibly multidirectional and possibly dissociated); it could never be mistaken for an Infantile Nystagmus Syndrome (INS) or a Fusion Maldevelopment Nystagmus Syndrome (FMNS); it was associated with head tilt and head oscillations in one case, with head tilt alone in two cases and with head oscillations alone in two cases. Analyses from oculographic recordings showed frequencies of 2.7-5 Hz (mean=3.7 Hz, median=3.6, SD=0.8), sinusoidal waveforms, dissociation and a special type of disconjugacy, both eyes oscillating with a 180 degree horizontal phase shift and no vertical phase shift, therefore exhibiting a "convection-like" movement pattern, close to the convergent-divergent pendular variety of nystagmus. Rarely and for short periods of time, the phase shift could change. These characteristics point towards oscillations in the vergence system, which could possibly result from the specific disruption of the vergence centers afferences in the brainstem, induced by the OPG during the sensitive period of visual development. This is the first study to provide a systematic description of the specific type of nystagmus associated with OPG. Its clinical and oculographic characteristics are unique among nystagmus in infanthood and cast light on its still incompletely elucidated mechanisms.

10.4 Conclusion

The collaboration with medical doctors requires to develop comprehensive tools to explore the signals. As there is no ground truth for the results, they need to be validated by an expert. The expert needs to be able to understand the limitations of the obtained results, in order to give a feedback, which can subsequently be used to improve the tool. The usage of the SSA to remove the gaze movement in the oculometric signals has been discussed with the clinician in order to ensure we did not add any artefact to the signal

which would hinder the following treatments. Also, comprehensive tools helped the expert to understand the issues of signal processing. While discussing the re-calibration issues for the signal, a change in the protocol was introduced in order to improve our capacity to validate the results. The representations of the quantified properties in the signal have been developed in order to link them to clinical observation. The heatmaps highlight the calm zone of the subject's nystagmus and the temporal presentation of the delay is a critical information for the identification of Spasmus Nutans. For clinical research, the goal is less to classify the data between categories than to combine the right information in order to understand the signal characteristics. The development of data driven signal representations for physiological signals is an exciting direction to tackle this challenge.

Conclusion and Perspectives

The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'Eureka!' but 'That's funny...'."

— *Isaac Asimov*

Convolutional representations are data-driven tools which can be used to summarize time series and to study their local structures. These methods emerge naturally when trying to understand physiological signals which are often formed with repetitive patterns such as the steps in human locomotion. The first part of this manuscript studies computational aspects of these representations for time series. We showed that for unitary dictionary, it is possible to compute such a representation using the Singular Spectrum Analysis. The decomposition computed with SSA can be used to compute a solution of a convolutional representation problem with dense activations and orthonormal patterns. Since this solution is dense, an extra step is necessary to ensure a good interpretability of the representation. We proposed a general framework to automatize this step which is usually done manually and we presented novel metrics to group the components with this step. Another way to improve the interpretability of this representation is to use sparse activation signals. In this context, we proposed a novel algorithm to solve the convolutional sparse coding. This algorithm runs in both distributed and sequential setting, and it accelerates the resolution of the optimization problem. We proved that this algorithm converges to the optimal solution of the considered problem and that it has a super-linear speed up compared to the greedy coordinate descent. This acceleration is sub-linear compared to the proposed sequential algorithm. The theoretical results were confirmed with numerical experiments. These two works described efficient algorithms to compute interpretable convolutional representations.

In the second part of this manuscript, we explored the link between deep learning models and signal representations. We first proposed an extra training step, which relies on the idea that the first layers of a network compute a representation of the dataset, to improve the training strategy. This step can be used after the training of a network. The weights of the first layers are fixed and we train the weights of the last layer for a small number of iterations. This improves the way the internal representation – computed with the first layers – is used to solve the considered task. This step is connected with results from kernel methods and we showed that this extra step provides consistent performance boost for multiple architectures. Then, we focus on recent works which presented certain optimization algorithms as neural network (Gregor & Lecun, 2010). These studies show that some common algorithms for sparse representation can be accelerated using trained neural networks. We presented a theoretical analysis of this acceleration for Learned ISTA (LISTA) networks and linked the acceleration to a quasi-diagonalization of the Gram matrix of the dictionary in a sparse basis. We showed that using this basis, we can derive an efficient algorithm, with same convergence rate than ISTA but potentially better constant factors. This algorithm can be shown to be a re-parametrization of

the LISTA network. Thus, LISTA is also able to accelerate the resolution when this factorization exists. Moreover, we designed an adverse example where the factorization was not possible and showed that LISTA also failed to accelerate the resolution on these examples. We also highlight under which conditions the performance of our factorization could be better than those of ISTA, in expectation over the generic dictionaries. With this second part, we study the deep learning models as two parts models, where first layers map the input to internal representations and the last layers compute a statistical model. The post-training ensures that the task-driven representations computed by the first layers are used optimally to solve the considered task. And the aim of the study of LISTA is to highlight the properties of dictionaries for which it is possible to efficiently compute sparse codes with neural networks. Combining these two ideas could help bring more interpretability for deep models.

Finally, we illustrated in the third part some of the results obtained on physiological signals. Robustly extracting the steps from a walk signal is a core block to automatize walk quantification. The early experiments with sparse convolutional dictionary learning for walk signal show that this technique is able to highlight local patterns with an unsupervised algorithm. The computed representation summarizes the signals in an interpretable way: on one side, patterns that look similar to steps and on the other side, activation signals which describe the regularity of the steps taken by the patient. We also presented a novel algorithm to detect steps robustly in walk signals. This algorithm is based on template matching with a step library and was evaluated on over 1000 walk signals, for both healthy and pathological subjects. The walk signal study was used in a medical publication. For eye movement quantification, we developed various tools to help the doctor study the nystagmus movement. We showed that the SSA could be used to remove the trend component from the registered signals and presented two representations which helped the ophthalmologist characterize the type of movement which was recorded. These tools were used to write a communication about the relationship between certain nystagmus and optical path-way gliomas. These two illustrations showed that convolutional representations can be used to highlight interpretable information in a signal.

These different works shed light on the properties of convolutional representations. This model is able to extract local structure in signal with unsupervised methods. All the presented results were produced using an ℓ_2-norm to compare the original signal with the reconstruction. For some applications, it is not the best way to compare the signals, for instance when the additive noise has some known structure. The recent work of Jas et al. (2017) proposed a model using an alpha-stable noise model instead of the Gaussian model, and showed that it was possible to solve it using an EM algorithm. Developing efficient algorithms for other types of noise is an interesting direction for future work. Another issue with the ℓ_2-norm is that all the channels of the original signal have the same weight. When learning a dictionary for signals with heterogeneous channels, parts of the signal are ignored. Finding the proper way to handle such signals would broaden the possibilities of convolutional dictionary learning. Also, the length of the extracted patterns is chosen manually, by selecting the shapes of the dictionary elements. With the ℓ_2 norm, small changes in the pattern scales can lead to large distances. Finding a way to extract scale invariant patterns would be useful for studies of unconstrained recording of physiological signals. A possible solution would be to introduce an extra parameter in the model to encode a scaling of the pattern used when it is activated. Using coordinate descent, a greedy algorithm can be used to solve the

resulting optimization problem in an efficient manner.

Appendices

A

Template-based step detection from accelerometer signals

Contents

Authors: Laurent Oudre[1,2], Rémi Barrois-Müller[2], Thomas Moreau[2,3], Charles Truong[2,3], Stéphane Buffat[2], Pierre-Paul Vidal[2]

Abstract: This article presents a method for step detection from accelerometer signals based on template matching. The principle of our step detection algorithm is to recognize the start and end times of the steps in the signal thanks to a predefined set of templates (library of steps). The algorithm is tested on a database of 1020 recordings, composed of healthy patients and patients with various neurological or orthopaedic troubles. Simulations on more than 40000 steps show that even with a library of only 5 templates, our method achieves remarkable results with a 98% recall and a 98% precision. The method is robust to parameter changes, adapts well to pathological subjects and can be used in a medical context for robust step estimation and gait characterization.

Keywords: gait analysis, biomedical signal processing, pattern recognition, step detection, physiological signals

[1]L2TI, Université Paris 13, France.
[2]COGNAC-G (UMR 8257), CNRS Université Paris Descartes, France.
[3]CMLA (UMR 8536), CNRS ENS Cachan, France .

A.1 Introduction

Pathologies affecting posture, balance, and gait control are threatening the autonomy of patients not to mention the risk of fall and therefore require rehabilitation intervention as early as possible. However, it remains difficult to accurately evaluate the various specific interventions during the rehabilitation process and the optimal content of exercise interventions they should involve. If only for these reasons, it would be interesting to learn how to monitor motor sensorimotor behavior at large and locomotion in particular which is a growing area in medical engineering science (Mariani, 2012; Marschollek et al., 2008; Willemsen et al., 1990; Dijkstra et al., 2008; Han et al., 2006; Ayachi et al., 2016; Williamson & Andrews, 2000). It requires several steps: first, we wish to investigate how to monitor sensorimotor processing in behaving patients in the doctor office and the resulting cognitive load it implies. Second, we want to learn how to construct databases with the quantitative variables recorded in that process, in order to make longitudinal studies of behaving individuals. Third, we would like to merge these individual databases in large data banks to define statistical norms, which is mandatory to detect dysfunctions or pathologies at the earliest stage possible. In that process we meet at least three main problems: using pervasive or ubiquitous computing to collect data; facing large inter-individual variability in the studied HMCs; aggregating highly heterogeneous data to build the databank.

There exist many software applications on the market that use wearable sensors (namely accelerometers, gyroscopes, magnetometers and/or GPS) to calculate the number of steps made in a day (Tran et al., 2012; Naqvi et al., 2012), the traveled distance in a day (Renaudin et al., 2012; Kim et al., 2004), the average speed, the daily amount of time spent in walking, running, sitting, standing, laying (Oner et al., 2012; Brajdic & Harle, 2013), useful for rehabilitation. Most of the algorithms published in this context are either dedicated to one specific terminal or mobile phone, or they are copyrighted and not freely available for research.

The main idea behind the algorithm presented in this paper is to automatically detect the steps from inertial sensor signals thanks to a library of templates extracted from real signals. It provides a novel, robust and precise step detection method which allows the user not only to count the steps, but also to locate when they occurred, how long they lasted, etc. These features can be useful either for personal or medical use. In particular, the algorithm has been tested on a large database containing 1020 walk exercises performed by healthy and pathological subjects at unconstrained speeds, which confirms the robustness of the presented method.

This article is organized as follows: Section A.2 defines the task of step detection and gives an overview of state-of-the-art methods. Section A.3 describes the data used for training and testing, the method, and the evaluation metrics. Section A.4 presents the results of our method, the influence of the parameters and compares the algorithm to state-of-the art methods. Section A.5 provides a discussion on the robustness of the method and several insights for the possible use of this algorithm in a clinical context.

A.2 Background

A.2.1 What is a step ?

Locomotion is a hierarchical and complex phenomenon composed of different entities such as strides, steps, and phases (Auvinet et al., 2002; Mariani, 2012).

- Considering one foot, the stride is the succession of two phases: the *swing phase* (when the foot is in the air), and the *stance phase* (when the foot is in contact with the ground). The stance phase occurs between the heel-strike (moment when the foot hits the ground) and the toe-off (moment when the toes go off the ground), while the swing phase occurs between the toe-off and the next heel-strike.
- A *stride* is defined as the event that occurs between two heel-strikes of the same foot.
- A *step* is defined as the event that occurs between successive heel strikes of opposite feet. A stride is therefore composed of two steps: one for the right foot, one for the left foot.

In the formal medical definition, a step is supposed to start when the heel strikes the ground and to finish somewhere in the end of the stance phase. It is not related to the foot activity since the foot is also moving in the swing phase. We choose in this article another definition: a step is defined in the following as the whole period of activity of a foot (when the foot is moving). The beginning of the step is defined as the heel-off (moment when the heel leaves the floor) and end of the step is defined as the foot-flat (moment when the foot is stabilized on the floor).This new definition allows to consider the whole period of activity of a foot as a step, which makes it more adapted to step detection. Note that it does not change the number of steps and that it is easy to switch back to the medical definition once the heel-off and foot-flat instants have been detected.

A.2.2 Existing methods

Current algorithms can be classified in two categories:

- Step counting algorithms: the aim is only to know the number of steps performed by the subject
- Step detection algorithms: the aim is to locate when the step occurred, and eventually to give specific timings (heel-strike, toe-off, etc.). These algorithms can also be used for step counting.

Among step detection algorithms, two main approaches have been proposed: the use of filtering/thresholding/peak detection techniques and the use of template matching. The former aims to recognize one specific event, supposedly characteristic of the step (such as a local maximum or the time when the signal exceeds a threshold). Most of the time, these algorithms include a preprocessing step where the signal is filtered so as to emphasize the event that they seek to detect or to remove other events. The most well-known pre-processing stage was designed by Pan & Tompkins (1985) and is composed of several signal processing blocks (bandpass filtering, derivation, squaring,

etc.). Designed at first for ECG signals, this pre-precessing has been used in various step detection methods (Ying et al., 2007; Libby, 2012; Marschollek et al., 2008; Thüer & Verwimp, 2008). After this possible processing stage, the steps are detected with empirical or dynamic thresholds, peak detection methods, of a combination of both (Mladenov & Mock, 2009; Dijkstra et al., 2008; Fortune et al., 2012). Other methods seek to detect each phase of the walking process by using dedicated signal processing techniques (such as peak detection, zero-crossing, etc.) (Willemsen et al., 1990; Han et al., 2006). Unfortunately, these methods heavily rely on the calibration of several parameters (width of the bandpass filter, window length, thresholds, etc.) (Ying et al., 2007; Libby, 2012; Marschollek et al., 2008; Thüer & Verwimp, 2008) which are difficult to estimate and thus set according to empirical experience. Moreover, these methods often assume some prior knowledge on the shape of a step (Willemsen et al., 1990; Han et al., 2006), which significantly limits the detection of unconventional patterns found with mobility-impaired patients.

For these reasons, we have decided in this article to focus on the second type of step detection methods, based on template matching. The main intuition behind this is that there are several types of steps (according to interpersonal variability, age, speed and pathology). Therefore, it is irrelevant to try to detect steps with one specific model (which is basically what is done with other methods since they only consider one set of parameters, thresholds, detection criteria, etc.). In order to overcome this issue, it is necessary to use a library of models (in our case a library of patterns) which represent typical step cycles. Hopefully, the use of this library can improve the robustness of the detection and paradoxically, prevent the overfitting induced by the choice of many parameters. Note that while commonly used in several other fields, this approach is novel in the context of step detection. We are aware of only one article mentioning the use of templates for step detection. Ying et al. (2007) is using one single template automatically extracted with filtering/thresholding/peak detection methods (thus relying on many parameters) and not from raw data. Also, in their paper, a different template is extracted for each subject, and only used for this particular subject. The novelty of the algorithm presented in this paper is that it uses a limited set of parameters whose influence is carefully studied and analysed. Also, our method is tested on a large database, with healthy and pathological subjects, at various speeds and in a rigorous cross-validation context.

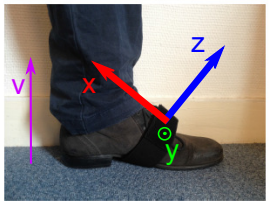
A.3 Data, method and evaluation

A.3.1 Data Acquisition and First Observations

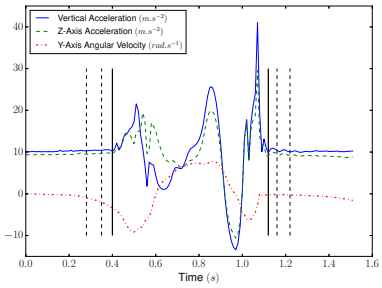
The data used for the conception and testing of the method presented in the article has been provided by the following medical departments: Service de chirurgie orthopédique et de traumatologie de l'Hôpital Européen Georges Pompidou, Assistance Publique des Hôpitaux de Paris, Service de médecine physique et de réadaptation de l'Hôpital Fernand Widal, Assistance Publique des Hôpitaux de Paris, Service de neurologie de l'Hôpital d'Instruction des Armées du Val de Grâce, Service de Santé des Armées. The study was validated by a local ethic comity (Comité de Protection des Personnes Ile de France II, CPP 2014-10-04 RNI) and both patients and control subjects gave their written consent to participate. All signals have been acquired at 100 Hz with wireless XSens MTw^{TM} sensors located at the right and left foot and fixed using a velcro band designed by XSens^{TM}. The signals obtained with both sensors were automatically

Group	**Number of exercises**	**Number of subjects**	**Sex (M/F)**	**Age (yr)**	**Height (cm)**	**Weight (kg)**
Healthy subjects	242	52	35/17	36.4 (20.6)	173.4 (10.8)	70.7 (12.2)
Orthopedic diseases	243	53	26/27	60.1 (19.3)	169.2 (10.2)	77.4 (16.8)
Neurologic diseases	535	125	80/45	61.6 (13.2)	169.8 (8.7)	72.7 (15.5)
Total	1020	230	141/89	55.5 (19.6)	170.5 (9.7)	73.4 (15.3)

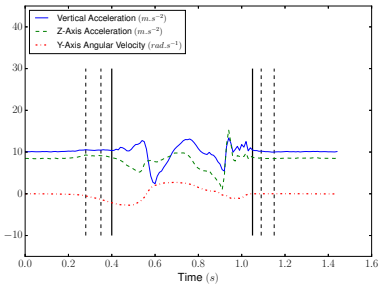
Table A.1: subjects' characteristics. For the age, height and weight, the mean and the standard deviations are displayed.



(a) Definition of the axis for the XSens^{TM} sensor located at the left foot



(b) Healthy patient



(c) Hip affected patient

Figure A.1: (a) XSens^{TM} sensor - (b,c) Vertical acceleration, Z-axis acceleration and the Y-axis angular velocity recorded from the right foot. The vertical lines display the different possibilities for start/end times.

synchronized by the acquisition software. All subjects were asked to:

- stand quiet for 6 seconds
- walk 10 meters at preferred walking speed on a level surface
- make a U turn
- walk back
- stand quiet 2 seconds

For practical reasons, patients kept their own shoes. The database is composed of 230 subjects who performed the protocol between 1 and 10 times, which leads to 1020 recordings. The subject's characteristics are presented in Table A.1. Healthy subjects had no known medical impairment. The orthopedic group is composed of 2 cohorts of distinct pathologies: lower limb osteoarthrosis and cruciate ligament injury. The neurologic group is composed of 4 cohorts: hemispheric stroke, Parkinson's disease, toxic peripheral neuropathy and radiation induced leukoencephalopathy.

The protocol includes 2 sensors (left and right foot), and each of them records a 9-dimensional signal (3D accelerations, 3D angular velocities, 3D magnetic fields), possibly with some recalibrated data provided by the XSens^{TM} software (such as the vertical acceleration in the direction of the gravity). Instead of considering all these dimensions, we decided to only use a subset of them, and select the most relevant in the context of step detection. This decision has been made based on observations of real data and physiological reasons provided by doctors. We decided to only select the components that are the most reflective of the locomotion process (see Figure 9.1a for the definition of the axis): the Z-axis acceleration, the recalibrated vertical acceleration (vertical movements of the foot) and the Y-axis angular velocity (swing in the direction of the walk). We expect these components to strongly react to the steps, making them identifiable.

Examples of these 3 components (Z-axis acceleration, vertical acceleration and Y-axis angular velocity) recorded at the right foot are presented on walk/ Figure A.1b and Figure A.1c for respectively an healthy and hip-injured patient. It appears on these walk/ that the amplitudes of the signals are clearly different and it is likely that classical threshold-based methods would hardly perform well on both subjects. However, the structure and shape of the step is roughly the same for both subjects so it might be relevant to use a template-base method. Nevertheless, these examples also display the main difficulties in conceiving an automatic algorithm for step detection:

- The uncertainties in the definition of the starts and ends of the steps. Indeed, we can see on Figure A.1b, that many choices would be acceptable: depending on the considered definition, the results may be different.

- The variability of the step patterns according to the pathology, the age, the weight, etc. For example, on Figure A.1c, the subject is dragging his feet, causing an abrupt change in the step pattern (noisy part at the end of the step).

A.3.2 Description of the method

The principle of our step detection algorithm is to recognize the steps in the signals thanks to a predefined set of templates. More precisely, our method uses a set of templates $\mathcal{P}$: these templates have been manually extracted from real accelerometer data and checked by doctors and specialists of locomotion. Each template $p \in \mathcal{P}$ is a three-dimensional signal of length $|p|$ (vertical acceleration, Z-axis acceleration and Y-axis angular velocity) corresponding to one step.

These templates are to be compared to the signal we want to study by calculating some correlation coefficients. As the sequences we want to detect are variable in duration as well as in amplitude, we want to use a measure of fit that is independent of the scale but is able to identify the correspondences in shape. Also, we want the comparison to be independent of the orientation of the sensor, so any DC component should be removed. In this context, it seems natural to use the Pearson correlation coefficient, which satisfies all these conditions, and defined for two one-dimensional vectors y and z of length n as

$$\rho_{y,z} = \frac{\operatorname{cov}(y,z)}{\sigma_y \sigma_z} = \frac{E[(y-\mu_y)(z-\mu_z)]}{\sigma_y \sigma_z} \tag{A.1}$$

where (μ_y, μ_z), (σ_y, σ_z) are respectively the mean and standard deviation of y and z.

Let x be a three-dimensional signal: we want to detect the steps by using the set of templates $\mathcal{P}$. Let us introduce the following notations:

- $|\mathcal{P}|$ is the number of three-dimensional templates
- $|x|$ (resp. $|p|$) is the length of the three-dimensional vector x (resp. p)
- $x^{(k)}$ (resp. $p^{(k)}$) is the k^{th} component of x (resp. p). In our case we have $k \in \{1, 2, 3\}$
- $x^{(k)}[t_1 : t_1]$ is the portion of $x^{(k)}$ between time samples t_1 and t_2 (we therefore have $x^{(k)}[1 : |x|] = x^{(k)}$)

The first step of the algorithm consists in calculating the Pearson correlation coefficients between the templates and the signal, for all possible time positions and all three components:

$$\begin{aligned} \forall k \in \{1, 2, 3\}, \quad \forall p \in \mathcal{P}, \quad \forall t \in [\![1, |x| - |p| + 1]\!] \\ r(k, p, t) = \rho\left(p^{(k)}, x^{(k)}[t : t + |p| - 1]\right) \end{aligned} \tag{A.2}$$

$r(k, p, t)$ is the correlation between the k^{th} component of template p and the k^{th} component of the signal at time sample t.

The second step is a local maxima search among the $r(k, p, t)$ coefficients in order to extract the possible steps. $r(k, p, t)$ is selected as a local maximum if it is greater than its nearest temporal neighbors. We define the set $\mathcal{L}$ of possible steps as:

$$\begin{aligned} \mathcal{L} = \Big\{ (k, p, t) \text{ s.t. } r(k, p, t) > r(k, p, t - 1) \\ \text{and } r(k, p, t) > r(k, p, t + 1) \Big\} \end{aligned} \tag{A.3}$$

The $\mathcal{L}$ contains all acceptable positions for the steps, and the coefficients $r(k, p, t)$ with $(k, p, t) \in \mathcal{L}$ can be interpreted as the likelihood of having a step similar to the pattern p at time sample t.

Our step detection algorithm takes as input the set $\mathcal{L}$ and works as a greedy process. At each iteration, the largest value $r(k^*, p^*, t^*)$ with $(k^*, p^*, t^*) \in \mathcal{L}$ is selected: if the step p^* positioned at time sample t^* overlaps with a previously detected step, it is discarded and we switch to the next largest value. Otherwise, if step p^* can be positioned at time t^*, the step is detected and all time samples between t^* and $t^* + |p^*| - 1$ are forbidden for the next iterations. The process is stopped when all time samples are forbidden, when the set of possible steps $\mathcal{L}$ is empty, or when all values $r(k, p, t)$ with $(k, p, t) \in \mathcal{L}$ are lower than a threshold λ. Note that in practice, the main purpose of threshold λ is to speed up the algorithm, as it reduces the size of set $\mathcal{L}$. The algorithm is summarized on Algorithm A.1.

A last post-processing step can be performed so as to discard the steps detected when the patient was actually not moving. These false detections occur when a fit is found with one template, even though the signal is almost equal to zero after DC component removal: this is in fact due to the invariance in scale provided by the Pearson correlation coefficients. A solution can be found by processing the final list of detected steps, and removing the steps whose standard deviation is way lower than the one of the template

Algorithm A.1 Step Detection Algorithm

Input: Set of possible steps $\mathcal{L}$
Output: Set of start times $\mathcal{T}_{start}$, set of end times $\mathcal{T}_{end}$
Set of forbidden time positions $\mathcal{F} = \emptyset$;
$\mathcal{T}_{start} = \emptyset$, $\mathcal{T}_{end} = \emptyset$
while $\mathcal{F} \neq \left\{1, \ldots, |x|\right\}$ or $\mathcal{L} \neq \emptyset$ or $\max \mathcal{L} > \lambda$ **do**
  $(k^*, p^*, t^*) = \underset{(k,p,t)\in\mathcal{L}}{\operatorname{argmax}}\ r(k, p, t)$;
  if $\left\{t^*, \ldots, t^* + |p^*| - 1\right\} \notin \mathcal{F}$ **then**
    $t^* \rightarrow \mathcal{T}_{start}$;
    $t^* + |p^*| - 1 \rightarrow \mathcal{T}_{end}$;
    $\left\{t^*, \ldots, t^* + |p^*| - 1\right\} \rightarrow \mathcal{F}$;
    $\mathcal{L} = \mathcal{L} \backslash (k^*, p^*, t^*)$;
  end if
end while

that was used for the detection. Formally, this step involves a threshold μ: given a detected step with start and end times t_{start} and t_{end}, detected thanks to the pattern $p^{(k)}$, the step is to be discarded if

$$\sigma_{x^{(k)}[t_{start}:t_{end}]} < \mu\ \sigma_{p^{(k)}} \tag{A.4}$$

where $\sigma_{\cdot}$ stands for the empirical standard deviation operator.

A.3.3 Evaluation

All steps were manually annotated by specialists using a software allowing to point with the mouse the starts (foot-flat) and the ends (heel-off) of the foot flat periods during which the sensor is not moving. The annotations were performed thanks to the Z-axis acceleration (normal to the upper foot surface) which is the most sensitive direction to detect the movements of the foot with respect to the floor. For the tricky cases of pathological gaits, a first gross annotation was made and then refined by zooming on each step. The uncertainty of this annotation is evaluated to less than 0.2 s (20 samples) for each mouse click. In total, the database is composed of 40453 steps (20233 extracted on the right foot and 20220 on the left foot). Even though they had a distinct shape, the U-turn steps were also taken into account.

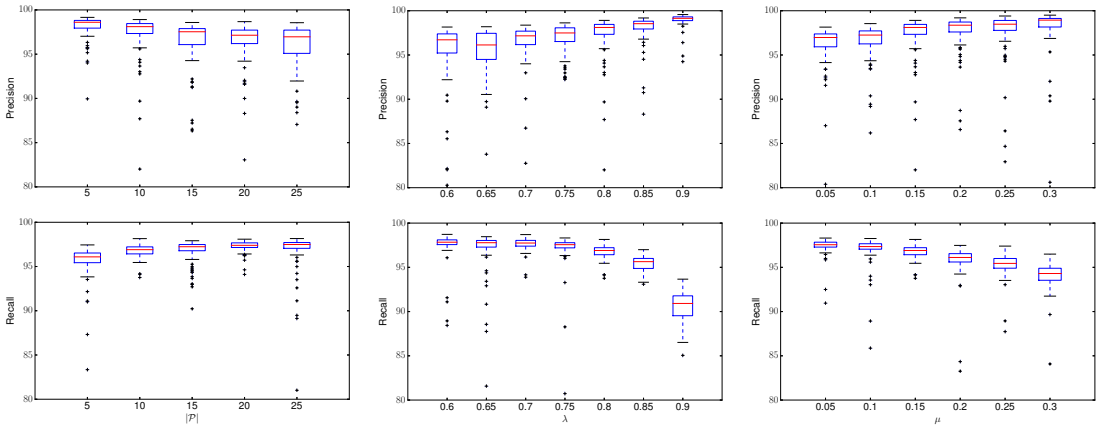
The following precision/recall metrics are used for the evaluation of our method based on the annotations provided by the specialists.

Precision. A detected step is counted as correct if the mean of its start and end times lies inside an annotated step. An annotated step can only be detected one time. If several detected steps correspond to the same annotated step, all but one are considered as false. The precision is the number of correctly detected steps divided by the total number of detected steps.

Recall. An annotated step is counted as detected if the mean of its start and end times lies inside a detected step. A detected step can only be used to detect one annotated step. If several annotated steps are detected with the same detected step, all but one are considered undetected. The recall is the number of detected annotated step divided by the total number of annotated steps.

A.4 Results

A.4.1 Influence of the Parameters



(a) Influence of $|\mathcal{P}|$ on the precision and recall (in %)

(b) Influence of λ on the precision and recall (in %)

(c) Influence of μ on the precision and recall (in %)

Figure A.2: Influence of the parameters (on 100 simulations). By default, $|\mathcal{P}| = 10$, $\lambda = 0.8$ and $\mu = 0.15$. Boxes correspond to quartiles and median, whiskers to 5 and 95 percentiles. Outliers are represented as $+$

The algorithm depends on 3 numerical parameters:

- The size of the pattern library $|\mathcal{P}|$
- The stopping criterion λ
- The threshold for discarding periods of no activity μ

Note that the algorithm is also influenced by the choice of the templates composing the library $\mathcal{P}$: this will be studied in the next section.

In order to study the scope of influence of these 3 parameters, a cross validation process is used:

- $|\mathcal{P}|$ three-dimensional step patterns are randomly chosen, so as to form the pattern library $\mathcal{P}$
- In order to avoid overfitting, all exercises performed by subjects that are used in the pattern library are then discarded from the test database.
- For each exercise of the test database, the step detection is performed with the $|\mathcal{P}|$ templates, and the detected steps are compared to the annotations

For each simulation, the mean and standard deviation of the precision/recall scores on the test database are calculated, as described in Subsection A.3.3. This process is performed 100 times.

The parameters are studied with the following grid search:

- $|\mathcal{P}| : [5, 10, 15, 20, 25]$

- $\lambda : [0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9]$
- $\mu : [0.05, 0.1, 0.15, 0.2, 0.25, 0.3]$

In total, 210 different configurations are considered.

The configuration giving the best average results on 100 simulations is using $|\mathcal{P}| = 10$ templates, $\lambda = 0.8$ and $\mu = 0.15$, with an average recall of 96.59% (std: 4.91) and an average precision of 97.03% (std: 3.69). Note that these values correspond to the average on 100 simulations with randomly chosen templates: they do not reflect the optimal performances of the algorithm.

We propose to use this configuration as a reference and study the influence of the parameters from this grid node. Figure A.2 presents the influence of the parameters on the precision and recall: on each figure, two of the parameters are fixed while the last one varies. The plots displays as boxplots the results obtained on the 100 simulations corresponding to the considered configuration.

On Figure A.2a, it is visible that adding more templates to the library tends to increase the recall, but it has a negative effect on the precision. This is probably due to the cross-validation process used for testing. Since the templates are randomly chosen, it is unknown if they belong to healthy or pathological subjects, to forward walking or U-turn, etc. Therefore, when $|\mathcal{P}|$ increases, it also increases the probability that a pathological step is used for detection. This is one of the predictable effect of this experiment: if a step within the library is *unadapted* for the task, it causes false detection and thus lowers the performances. However, this does not mean that adding *appropriate* steps in the library would degrade the performances: this problem will be investigated in the next section (as well as the questionable notion of *appropriate steps*). When $|\mathcal{P}| = 5$, the limits of the algorithm are reached: due to the small number of templates, the method crucially depends on the choice of the templates used for detection, thus causing a large number of outliers. The best compromise between precision and recall is obtained for $|\mathcal{P}| = 10$, but this might only be due to the cross-validation setting: rather than an optimal number of templates to be used, it is likely that the composition of the library is more crucial to the performances of the algorithm.

The plot on Figure A.2b is coherent with the definition of the parameter: when λ increases, only steps that are very correlated to the templates are selected: this increases the precision, but decreases the recall. On the contrary, when λ decreases, all possible steps are considered: the recall increases and the precision decreases. These results also confirm the utility of parameter λ: by increasing λ to an appropriate value (around 0.6-0.8), it is possible to increase the precision (and the robustness of the precision) while keeping the recall constant. Also, λ has an impact on the computational cost: for example, using $\lambda = 0.8$ instead of $\lambda = 0$ allows to compute the results approximately 2 times faster. It is therefore interesting to use the largest value of λ as possible. The best average performances are obtained for $\lambda = 0.8$, which constitutes a good compromise between recall and precision: indeed, with $\lambda = 0.85$ some annotated steps are discarded and the recall drops.

Figure A.2c shows that parameter μ mainly influences the recall. Indeed, when μ is too large, all steps whose amplitude are too different from those of the templates are discarded. This has a double effect: if one of the templates corresponds to a pathological patient whose steps have small amplitude, then it will not be able to detect steps on healthy patients. The opposite situation can also occur. In fact, when μ increases, the

normalization effect provided by the Pearson correlation coefficient (A.1) is neutralized. Figure A.2c shows that μ should be no greater than 0.2 so that the recall does not drop.

A.4.2 Influence of the composition of the library

The performances of the algorithm are intuitively dependent of the library of templates used for detection. As previously seen, when inappropriate steps are added to the library, the performances may drop. What would happen if the library of templates is composed only of healthy steps, but is to be used on patients with degraded walking abilities ? In order to correctly detect steps for a patient having e.g. an orthopedics disease, is it necessary to have patients with similar pathologies in the library of templates ?

To investigate this question, we propose to define two classes of subjects within the database: class A is typically composed of subjects who have no problem for walking, and class B is composed of subjects with severe pathologies that critically affect their locomotion. The idea is to study the cross-performances of the method on these two classes. The definition of these classes are non-trivial since the database contains gait recordings of patients cared for lower limb osteoarthritis, anterior cruciate ligament injury, hemispheric stroke, Parkinson's disease and neuropathy. In each nosologic class, patients were quoted by the medical doctors of our group with clinical scales specific to each pathology (WOMAC index : lower limb osteoarthritis ; Tegner Lysholm Knee Scoring Scale : anterior cruciate ligament injury ; Lower Limb Fugel Meyer scale : stroke ; UPDRS III : Parkinsons Disease ; TNSc : neuropathy). To allow the between pathology comparison, a transversal walking score (between 0 and 4) was assigned to each patient by the medical doctors of our group. Subjects with no problem for walking were graded 0, while other were graded from 1 to 4 (4 being the most severe degradation of locomotion). To have an idea, lower limb osteoarthritis patients with high functional manifestation walking troubles (use of cane, unable to climb stairs) were graded 4. Class A is defined as subjects with a locomotion grade of 0 (no problem) and Class B as subjects with locomotion grade of 3 or 4. In total 116 subjects are isolated from the database: 72 subjects in Class A (322 exercises, 4877 left steps, 4846 right steps), and 35 subjects in Class B (111 exercises, 3554 left steps, 3567 right steps).

In each simulation, the library is composed of templates belonging to only one class, and the test is performed on exercises belonging to only one class. All simulations are run with the default parameters $|\mathcal{P}| = 10$, $\lambda = 0.8$ and $\mu = 0.15$ (that gave the best average performances on 100 simulations in the grid search). Table A.2 presents the results (recall/precision) averaged on 100 simulations. A first observation is that Class A and Class B templates give similar (and good) performances on Class A subjects. This confirms the intuitive idea that it is easier to detect steps for healthy subjects. However, Class B templates used on Class B subjects do not perform so well: it might be due to the definition of the class which involves several types of pathologies. In fact, these severe pathologies might affect the steps shapes in a different way, so even though some pathological templates are used for detection, they might not correspond to the particular pathology of the test subject. To increase the scores, two strategies can be implemented: either introduce all types of degradations within the library, or add several healthy (or less pathological) steps which could smooth the results by introducing less specific examples. Interestingly, the results obtained on Class B subjects with random templates and with the exact same parameters (see Subsection A.4.1) are better than those obtained by using only Class B templates. This tends to show that in order to

		Test data	
		Class A	Class B
Template data	Class A	R : 97.64 (1.17) P : 97.45 (4.46)	R : 89.74 (3.82) P : 95.75 (5.09)
	Class B	R : 97.80 (1.32) P : 97.28 (2.17)	R : 93.25 (4.17) P : 93.13 (5.76)

Table A.2: Influence of the composition of the library of templates in the step detection ($|\mathcal{P}| = 10$, $\lambda = 0.8$ and $\mu = 0.15$). Average recall and precision on 100 simulations (with standard deviation). Class A: subjects who have no problem for walking. Class B: subjects with severe pathologies that critically affect their locomotion.

	Best simulation		Pan-Tomkins		One template	
Group	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**
Healthy subjects	98.93 (2.22)	98.98 (2.43)	99.14 (1.71)	97.09 (3.60)	99.03 (2.06)	99.33 (1.76)
Orthopedic diseases	97.54 (2.92)	98.77 (2.12)	98.78 (2.09)	94.87 (5.09)	97.37 (3.06)	98.85 (2.23)
Neurological diseases	98.55 (3.05)	98.05 (3.02)	96.80 (3.52)	95.49 (4.55)	98.11 (3.31)	98.58 (2.55)
Total	98.40 (2.89)	98.44 (2.72)	97.82 (3.07)	95.72 (4.56)	98.15 (3.05)	98.82 (2.33)

Table A.3: Detailed performances of the best step detection method ($|\mathcal{P}| = 5$, $\lambda = 0.75$ and $\mu = 0.1$), the best Pan-Tomkins method, and the best step detection method with one template ($|\mathcal{P}| = 1$, $\lambda = 0.6$ and $\mu = 0.15$). Means and standard deviations are displayed.

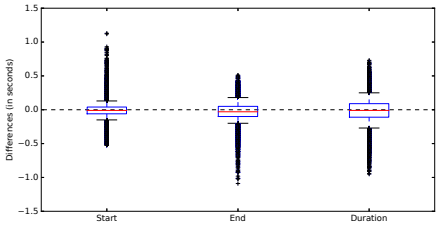
detect steps on severe pathological subjects, it is necessary to use a library composed of both healthy (or slightly pathological) and pathological steps.

As far as cross-class detection is concerned, it seems that using only Class A templates for detecting Class B steps is not appropriate : the recall drops while the precision decreases. It is likely that these results are due to the amplitudes of the steps that greatly vary between healthy and pathological subjects. Due to parameter μ, steps with low amplitude are hardly detectable with high amplitude templates (and vice-versa). Also, the durations of the steps might be inappropriate for detection, since pathological steps are in general longer than healthy steps.

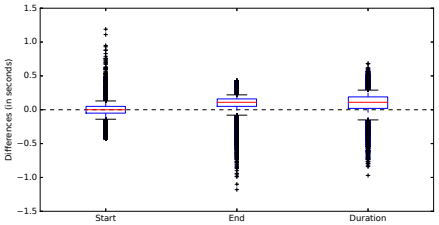
To summarize, two trends can be identified: as far as healthy subjects are concerned, the choice of templates is not crucial for the detection. But if the algorithm is to be used on pathological subjects, it appears that the best compromise would be to use a combination of healthy and pathological templates.

A.4.3 Detailed results for the best simulation

The best simulation on the whole grid search (21000 simulations) described in Subsection A.4.1 is using parameters $|\mathcal{P}| = 5$, $\lambda = 0.75$ and $\mu = 0.1$, with 98.40% recall and 98.44% precision. In this section, we propose a detailed study of this method. Note that this particular method should only be seen as a good association (templates + λ + μ)



(a) Best step detection method ($|\mathcal{P}| = 5$, $\lambda = 0.75$ and $\mu = 0.1$)



(b) Best step detection method with one template ($|\mathcal{P}| = 1$, $\lambda = 0.6$ and $\mu = 0.15$)

Figure A.3: Differences between detected and annotated times (start, end and duration) for the best step detection method and the best step detection method with one template. Boxes correspond to quartiles and median, whiskers to 5 and 95 percentiles. Outliers are represented as +.

performing well, and does not constitute a golden standard (similar scores are obtained on several other simulations).

The detailed performances of this method on the whole database is presented on Table A.3: it is noticeable that scores are consistent on all groups of subjects. The best performances are obtained for healthy subjects, but there is no significant differences between the groups. This clearly shows that the method adapts well to different types of pathologies.

Out of the 40344 detected steps, 85% of them were detected with the Y-axis angular velocity, 2% with the vertical acceleration and 13% with Z-axis acceleration. This proportion can be due to the nature of the signals: medio-lateral angular velocity is actually known to be the direction in which there is the greatest quantity of movement during walking. This signal is often used in step detection (Salarian et al., 2004; Ben Mansour et al., 2015), and it is likely that this component captures a locomotion pattern that is the most reproducible among the subjects.

The good performances of this method are intuitively linked to the templates composing the library. It is remarkable that this method only requires a small number of templates, which tends to show that the algorithm do not need a large library to perform accurately. It probably rather needs a carefully selected set of templates, that are generic enough to fit the general shape of a step, but can also adapt to pathological steps. For instance, this library of 5 templates is composed as follows: 1 step belonging to an healthy subject, 3 steps corresponding to neurological diseases (2 with moderate troubles and 1 with severe troubles), and 1 step associated to orthopedic diseases (with moderate troubles). This covers all groups of subjects and the proportion of each group in the library is similar to the one of database. In particular, the neurological group is composed of many different diseases and it is likely that several patterns are necessary to accurately fit the whole range of step shapes.

In order to further investigate the accuracy of the method, some additional evaluation metrics are computed. For all correctly detected steps, we compute:

- the difference between the detected start time and the annotated start time
- the difference between the detected end time and the annotated end time

- the difference between the duration of the detected step and the duration of the annotated step

The repartition of these metrics on all 39677 correctly detected steps are presented on Figure A.3a. One interesting result is that our method does not introduce a bias: the median of the differences for all times (start, end, duration) is approximately equal to zero, and the quartiles are symmetric. This tend to prove that the library is able to accurately detect the step boundaries and to adapt to various step durations. For 90% of the steps (represented as whiskers on the figure), the errors for start, end and duration times are lower than 0.25 seconds (in absolute value), which corresponds to 25 samples. These results are satisfactory when compared to the annotations uncertainties of experts and specialists (which are around 20 samples - see Subsection A.3.3). Outliers are in fact due to two specificities of the database: tiny steps (under 50 samples) mainly located during U-turn (causing underestimation for start times and overestimation of end and duration times), and highly pathological steps for stroke subjects whose duration exceeds one second (causing upper outliers for start times and lower outliers on end and duration times). The method tested here is using five templates of durations 65, 76, 82, 86 and 105 samples and the detection is inevitably constrained by these step durations. While this phenomenon does not penalize the results on most steps, it is one limit of the algorithm especially with small libraries. Should these outliers become more frequent, one possible solution is to increase the number of templates and to add typical steps corresponding to these outliers within the library.

A.4.4 Comparison with the state-of-the-art

The reference procedure for step counting/detection is based on the Pan-Tomkins method (Pan & Tompkins, 1985). First intended for ECGs, it was later adapted to detect steps in the vertical accelerometer signal (Ying et al., 2007; Libby, 2012; Marschollek et al., 2008; Thüer & Verwimp, 2008). It is composed of several successive signal processing steps, which are designed to emphasize the structure of the step, making it easier to detect. These steps can be summarized as:

- Bandpass filtering (between f_{min} andf_{max}): removes the gravity component and the noise.
- Derivation: amplifies the slope changes in the filtered signal. Whenever the foot rises from the ground or the heel hits the ground, the acceleration slope changes significantly and it translates into a burst in the filtered signal.
- Squaring: makes all points positive and enhances the large values of the filtered signal.
- Integration: the signal is smoothed using a moving-window integrator of length N_{inte}.
- Peak search procedure: originally, Pan & Tompkins (1985) used a threshold to find the phenomena they were looking for in the heart rate signal (every time the filtered signal was above the threshold, it was considered as detected). When they adapted the algorithm to the step detection problem, Ying et al. (2007) relied on the fact that the filtered signal showed great regularity: a small peak was always followed by a bigger one (respectively matching the foot lift and the heel strike).

The time span of the second peak was defined as the peak-searching interval on the real acceleration signal. The maximum on that interval was considered a step.

Note that this step detection procedure only allows to detect steps but not to precisely know the start and end times of the step. Also, this method is not designed to perform properly during periods of no activity. We therefore added a post-processing step, which, once a step is detected, compares the standard deviation of a neighborhood around the detected peak to a noise level. The size of the neighborhood, as well as the noise level, are optimized by grid search so as to give the best performances.

In Ying et al. (2007), the parameters used are $f_{min} = 0$ Hz, $f_{max} = 20$ Hz, $N_{inte} = 0.1$ s. The peak search procedure is performed sequentially: they select one peak every other peak, starting with the second one. With these parameters, we obtain of our database a recall of 99.53% and a precision of 51.20%. In fact, the peak-search procedure is not adapted and tend to detect several peaks within a step except of only one. This phenomenon has already been described by Libby (2012) and Thüer & Verwimp (2008).

In order to objectively compare our method to the Pan-Tomkins, we therefore tested several values for f_{min}, f_{max} and N_{inte}, as well as a more relevant peak-search procedure, which only selects the local maxima among the detected peaks, thus preventing multiple detections. In total, 5 parameters need to be optimized by grid search (filter bandpass $\times$ 2, integration window, neighborhood size and noise level). When optimized on the whole database so as to maximize the F-measure, the algorithm gives a 97.82% recall and a 95.72% precision. Detailed results are presented on Table A.3 : while these scores are comparable with our method on healthy subjects, it is noticeable that Pan-Tomkins method has difficulty to deal with neurological and orthopedics subjects. In particular, on these subjects, an overdetection occurs, thus decreasing the precision. One possible explanation is that signals associated to pathological subjects tend to have smaller amplitudes and to be noisier that those belonging to healthy subjects. Thus, if the parameters of the filtering are inadapted, the preprocessing tends to increase the level of noise and to create artefacts that as misdetected as steps. This may be one limit of step detection methods based on signal processing: if the signals to be studied have different properties (noise, frequential content, amplitudes), it is tricky to find one unique processing adapted to all signals. This problem is overcome in template-based methods which inherently consider several models.

A.5 Discussion and perspectives

The main idea behind the algorithm is that there is not one typical step but rather several typical steps. This assumption is confirmed by the results obtained with state-of-the-art methods, which inherently define only one model and obtain degraded performances when confronted to pathological data. To go further, it is interesting to degrade the algorithm with only one template and look at the consequences on the results. A second grid search is conducted with the same parameters as in Subsection A.4.1, but considering libraries composed of one unique template.

The best results are displayed on Table A.3. The metrics used in Subsection A.4.3 are also evaluated for this simulation and presented on Figure A.3b. Surprisingly, the precision and recall are comparable with those obtained with five templates. The template used for detection in this method belongs to an orthopaedic subject with moderate troubles and lasts 82 samples (which is close to the median step duration on the data-

base which is equal to 77 samples). It seems that the task of step counting can be performed with only one template. However, it can be seen on Figure A.3b that using only one template creates a bias and a systematic error on the estimation of end and duration times. Due to the large duration of the template used for detection, an overestimation of the duration often occurs.

We believe this simulation shows that the use of a single template is adapted for step counting on most subjects. The use of templates appears to give better performances than thresholding methods for step detection. However, if additional information are desired (such as the start and end times of the steps), it is crucial to take into account the variability of the subjects and of their locomotion, which can be done by adding several templates that reflect the different step durations and shapes.

Intuitively, the composition of the library is a fundamental feature of the algorithm. The choice of the templates to be used is an interesting question that can be answered in many different ways. In a medical context, templates can for example be introduced according to the characteristics and pathologies of the subjects to be studied: a neurologist may benefit from a library of templates composed of a selection of different neurological pathologies. They can also be specified by experts such as biomechanists who can extract typical steps covering the whole range of types of locomotion. Unsupervised machine learning techniques (such as dictionary learning) can also be used to automatically extract typical steps that are found on several exercises. It is also relevant to test semi-supervised techniques that could automatically choose the best library according to the input signal. All these leads are to be studied soon in collaboration with medical doctors and experts, and on more pathologies.

A.6 Conclusion

We have described in this article a template-based method for step detection. This method, based on a greedy algorithm and a library of annotated step templates, achieves good and robust performances even with a small number of templates. When used on a large database composed of healthy and pathological subjects walking at different speeds, the method obtains a 98% recall and 98% precision. Moreover, the algorithm allows to detect the start and end times of each step with a very good precision even on pathological subjects.

Thanks to its robustness and low computational cost, this method could be extended to process signals acquired in free-living conditions. Indeed, the actual protocol is composed of a no activity period and a U-turn, and there is no obstacles for testing the algorithm on unconstrained walking. The algorithm may also be adapted to a lighter protocol using only waist accelerometer signals and based on the same principle.

Another topic of interest is the choice of the templates to be used in the library (as presented in Section A.5). Several selection processes could be implemented in order to automatically adapt to any type of pathology and to optimize the performances of the algorithm.

ACKNOWLEDGMENTs

The authors would like to thank N. Vayatis, D. Ricard, A. Yelnik, C. De Waele and T. Grégory for the thorough discussions, the design of the experiment, the data acquisition and clinical annotation. This work was supported by SATT Ile-de-France Innov.

Bibliography

Auvinet, B., Berrut, G., Touzard, C., Moutel, L., Collet, N., Chaleil, D., and Barrey, E. Reference data for normal subjects obtained with an accelerometric device. *Gait & posture*, 16(2):124–134, 2002

Ayachi, F., Nguyen, H., Goubault, E., Boissy, P., and Duval, C. The Use of Empirical Mode Decomposition-Based Algorithm and Inertial Measurement Units to Auto-Detect Daily Living Activities of Healthy Adults. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(10):1060–1070, 2016

Ben Mansour, K., Rezzoug, N., and Gorce, P. Comparison between several locations of gyroscope for gait events detection. *Computer methods in biomechanics and biomedical engineering*, pp. 1–2, 2015

Brajdic, A. and Harle, R. Walk detection and step counting on unconstrained smartphones. In *Proceedings of the ACM international joint conference on Pervasive and ubiquitous computing*, pp. 225–234, Zurich, Switzerland, 2013. ACM

Dijkstra, B., Zijlstra, W., Scherder, E., and Kamsma, Y. Detection of walking periods and number of steps in older adults and patients with Parkinson's disease: accuracy of a pedometer and an accelerometry-based method. *Age and ageing*, 37(4):436–441, 2008

Fortune, E., Lugade, V., Morrow, M., and Kaufman, K. Step counts using a tri-axial accelerometer during activity. In *Proceedings of the American Society of Biomechanics Annual Meeting (ASB)*, Gainesville, FL, USA, 2012

Han, J., Jeon, H. S., Jeon, B. S., and Park, K. S. Gait detection from three dimensional acceleration signals of ankles for the patients with Parkinson's disease. In *Proceedings of the International Special Topic Conference on Information Technology in Biomedicine (ITAB)*, Ioannina, Greece, 2006

Kim, J., Jang, H., Hwang, D.-H., and Park, C. A step, stride and heading determination for the pedestrian navigation system. *Journal of Global Positioning Systems*, 3 (1-2):273–289, 2004

Libby, R. A Simple Method for Reliable Footstep Detection in Embedded Sensor Platforms. Research report, 2012

Mariani, B. *Assessment of Foot Signature Using Wearable Sensors for Clinical Gait Analysis and Real-Time Activity Recognition.* PhD thesis, EPFL, 2012

Marschollek, M., Goevercin, M., Wolf, K.-H., Song, B., Gietzelt, M., Haux, R., and Steinhagen-Thiessen, E. A performance comparison of accelerometry-based step detection algorithms on a large, non-laboratory sample of healthy and mobility-impaired persons. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, pp. 1319–1322, Vancouver, Canada, 2008

Mladenov, M. and Mock, M. A step counter service for Java-enabled devices using a built-in accelerometer. In *Proceedings of the International Workshop on Context-Aware Middleware and Services (COMSWARE)*, pp. 1–5, Dublin, Ireland, 2009. ACM

Naqvi, N. Z., Kumar, A., Chauhan, A., and Sahni, K. Step Counting Using Smartphone-Based Accelerometer. *International Journal on Computer Science and Engineering*

(IJCSE), 4(5):675–682, 2012

Oner, M., Pulcifer-Stump, J., Seeling, P., and Kaya, T. Towards the run and walk activity classification through Step detection-An Android application. In *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1980–1983, San Diego, CA, USA, 2012

Pan, J. and Tompkins, W. J. A real-time QRS detection algorithm. *IEEE Transactions on Biomedical Engineering*, 32(3):230–236, 1985

Renaudin, V., Susi, M., and Lachapelle, G. Step length estimation using handheld inertial sensors. *Sensors*, 12(7):8507–8525, 2012

Salarian, A., Russmann, H., Vingerhoets, F., Dehollain, C., Blanc, Y., Burkhard, P., and Aminian, K. Gait assessment in Parkinson's disease: toward an ambulatory system for long-term monitoring. *IEEE Transactions on Biomedical Engineering*, 51(8):1434–1443, 2004

Thüer, G. and Verwimp, T. Step detection algorithms for accelerometers. Master's thesis, Artesis University College of Antwerp, Belgium, 2008

Tran, K., Le, T., and Dinh, T. A high-accuracy step counting algorithm for iPhones using accelerometer. In *Proceedings of the International Symposium on Signal Processing and Information Technology (ISSPIT)*, pp. 213–217, Ho Chi Minh City, Vietnam, 2012. IEEE

Willemsen, A., Bloemhof, F., and Boom, H. Automatic stance-swing phase detection from accelerometer data for peroneal nerve stimulation. *IEEE Transactions on Biomedical Engineering*, 37(12):1201–1208, 1990

Williamson, R. and Andrews, B. Gait event detection for FES using accelerometers and supervised machine learning. *IEEE Transactions on Rehabilitation Engineering*, 8(3): 312–319, 2000

Ying, H., Silex, C., Schnitzer, A., Leonhardt, S., and Schiek, M. Automatic step detection in the accelerometer signal. In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 80–85, Aachen, Germany, 2007

C

Optic pathway gliomas-associated nystagmus

Contents

Authors: Matthieu P. Robert, Jacques Grill, Thomas Moreau, David Grevent, Olivia Zambrowsky, Pascale Varlet, Emile Contal, Gilles Martin, Dominique Brémond-Gignac, Isabelle Ingster-Moati, Christelle Dufour, Laurence Brugières, Nicolas Vayatis, Nathalie Boddaert, Christian Sainte-Rose, Thomas Blauwblomme, Stéphanie Puget, Pierre-Paul Vidal

C.1 Introduction

While acquired nystagmus can be the sign of a brain tumour, this is almost never the case for early-onset nystagmus. "Early-onset" nystagmus were previously designated as "congenital", but this term has been abandoned, as they appear usually between 4 and 12 weeks of life, and by convention before 6 months of age. Optic pathways glioma (OPG)-associated nystagmus can be misleading in many ways: it is considered an "acquired" form of nystagmus, but it can appear so early that the age of onset might not be discriminant (Toledano et al., 2015). It is often described as a *spasmus-nutans*-type nystagmus, and can also disappear secondarily –like idiopathic *spasmus-nutans*–, while the tumour actually remains stable (Brodsky & Keating, 2014). These rare nystagmus are the source of multiple challenges. Clinically, if misdiagnosed as infantile nystagmus syndrome (INS), the underlying OPG diagnostic may be delayed; although far less frequent than INS or other varieties of nystagmus, they have led some clinicians to systematically prescribe brain imaging to any infant presenting with a nystagmus, while this may not be necessary in the great majority of cases. More generally, the pathophysiology of disconjugate and dissociated nystagmus resulting from an optic pathway anomaly is so far unexplained. There is to date no series describing systematically the clinical and oculographic characteristics of OPG-associated nystagmus and the literature remains poor and contradictory on that topic. The Classification of Eye Movement Abnormalities and Strabismus (CEMAS) does not even mention them (Group, 2001).

Spasmus-nutans-type nystagmus definitions are numerous and are a source of confusion. Historically, Raudnitz reported in 1897 a triad of nystagmus, head turn and head oscillations occurring in infants living in a dark quarter of Prague (Raudnitz, 1897).

This nystagmus was reported to appear between 6 and 12 months of age, to always disappear after a few months or years of evolution and to represent a benign entity. Since 1967, however, many cases, most often isolated case reports, have stressed on the possible association with suprasellar tumours, mainly OPG (Donin, 1967; Farmer & Hoyt, 1984; Kelly, 1970; Lavery et al., 1984). Since then, the term *spasmus-nutans* has been either designating the clinical syndrome whatever its cause, or the specific, benign entity. The CEMAS chose the latter and proposed criteria for *spasmus-nutans* syndrome: "Infantile onset, variable conjugacy, small frequency, low amplitude oscillation, abnormal head posture and head oscillation, improves ("disappears") during childhood, normal MRI/CT scan of visual pathways. Ocular motility recordings –high frequency (>10 Hz), asymmetric, variable conjugacy, pendular oscillations", with common associated findings: "Disconjugate, asymmetric, multiplanar, family history of strabismus, may be greater in one (abducting) eye, constant, head posture/oscillation (horizontal or vertical), usually no associated sensory system deficits, may have associated strabismus and amblyopia, may increase with convergence, head bobbing, head posture may be compensatory" (Group, 2001). Very few studies including oculomotor recordings have addressed the matter. Weissman et al. (1987) studied seven patients with *spasmus-nutans* and showed that in all cases, both eyes exhibited phase differences, most often of 180 degree, which varied over time. This was ascertained by the authors as being the hallmark of *spasmus-nutans*. Formal eye movements recordings are here most helpful, as the high frequency and the low amplitude often preclude any such observation through the naked eye. Arnoldi & Tychsen (1995) reviewed 67 consecutive children diagnosed with *spasmus-nutans* among whom none had evidence of a glioma. On the other hand, Toledano et al. (2015) found monocular nystagmus at diagnosis in 41% of 22 children with measurable OPG. The authors are not aware of any study about OPG-associated nystagmus including formal oculomotor recordings.

Infantile nystagmus syndrome may be observed in infants with low vision resulting from anterior visual pathway dysfunction. It usually then secondarily evolves to a vision loss nystagmus, which is a jerk, conjugate nystagmus with horizontal, vertical and torsional components, often of large amplitude (Group, 2001). Visual loss nystagmus is present in old children with OPG with the worst functional evolution, but both INS and vision loss nystagmus seem distinct from most descriptions of OPG-associated nystagmus. Monocular visual loss due to optic nerve involvement can be associated either with a slow, pendular, purely vertical nystagmus, often referred to as Heimann-Bielschowsky phenomenon, or sometimes with a low amplitude horizontal nystagmus (Good et al., 1993). This would be consistent with some cases of reported OPG-associated nystagmus, although it has been suggested that in the case of monocular OPG-associated nystagmus, the eye with the nystagmus could be the one with the best vision (Farmer & Hoyt, 1984). Lesions at the optic chiasm and parasellar lesions can also rarely be associated with pendular seesaw nystagmus. Seesaw nystagmus has been attributed either to the effects of an associated midbrain compression –which can be seen in some cases of OPG– or to the effects of the resulting temporal visual loss, inactivating the calibration mechanism for eye movements normally compensating for head rotations in roll (Leigh & Zee, 2015). Again, OPG-associated nystagmus descriptions are not compatible with seesaw nystagmus.

In that context, many questions related to important clinical issues remain unsolved. Are OPG-associated nystagmus always of *spasmus-nutans*-type? Can they sometimes fit thecriteria of INS? Do they ever exhibit the complete triad of *spasmus-nutans*, including

head oscillations, or, as suggested by some authors (Hertle & Dell'Osso, 2013), does the presence of associated head-oscillations associated with a *spasmus-nutans*-type nystagmus preclude it being associated with a tumour? Do OPG associated with a nystagmus at diagnosis have specific characteristics? What is the mechanism of OPG-associated nystagmus?

C.2 Material and methods

The way patients were selected is summarised in Figure C.2. Children (0-18 year-old) having presented with optic pathway gliomas at the paediatric neuro-ophthalmology clinic in *Hôpital Universitaire Necker-Enfants malades* between October 2009 and October 2014 (so as to allow for a follow-up of at least one year, until October 2015) were prospectively recorded in the database for rare diseases CEMARA. Those having been revealed by a nystagmus were identified. Children having been referred to the clinic over the same period of time for a nystagmus, which onset had been before two years of age, were similarly registered. Nystagmus that had appeared secondarily, as a consequence of an already known diagnosis, were not included. Nystagmus clinically classified as *spasmus-nutans*-type were identified, as well as nystagmus having led to the diagnosis of a brain tumour. *Spasmus-nutans*-type nystagmus was defined as pendular, medium or high-frequency, low-amplitude, often multidirectional, with a dissociation between the movements of the right vs the left eye. Cases of children presenting with a nystagmus with a later onset were not included, since these cases are not all seen in the paediatric neuro-ophthalmology clinic, nor therefore systematically registered in databases. Only the children who also benefitted from reproducible nystagmus recordings at diagnosis (video recording and/or infrared photoreflectometry with the Eyefant*, Ober consulting, Poland, an eyetracker specifically designed for young children and allowing for binocular recording at 1000 Hz) were included.

All had benefited from systematic ophthalmological examination, initially and during follow- up. All children diagnosed with OPG were followed-up by an interdisciplinary team comprising paediatric oncologists, neurosurgeons, paediatric endocrinologists, pathologists and ophthalmologists. Decisions were taken during neuro-oncology staff meetings. In the absence of clinical diagnostic criteria for neurofibromatosis type 1, the children benefitted from initial biopsy and histology, except for cases of intrinsic limited forms in young infants after consensual discussion. All children benefitted from hormonal check-up, both at initial evaluation and during follow-up. Visual evoked potentials (VEP) were performed in five cases, in order to better quantify the infants' visual function. Reversal of pattern, ON-OFF and flash VEP were performed with the 48-cm-diagonal screen at a 100 cm distance from the child, using the LED stimulator Vision Monitor MonPack by Metrovision system (Pérenchies, France) according to the International Society for Clinical Electrophysiology of Vision (ISCEV) protocol with five tinned, copper-cup occipital active electrodes (Odom et al., 2010).

The clinical and oculographic characteristics of the nystagmus; the clinical, imaging and histological characteristics of the OPG, were studied. The treatments' modalities and the visual outcomes were studied. When the Eyefant* was used, the nystagmus was recorded in the primary position of gaze, under binocular, right and left monocular viewing conditions; for movements requiring stimuli, a 19" screen was used with stimuli especially designed for infants. The infant was placed on his parent lap, 50cm from the screen. Calibration was made a posteriori, using the movements of the eyes doing

saccades from a central cue to four to eight eccentric locations. Before analysing the characteristics of the signals, singular spectrum analysis (SSA) was used, in order to isolate the nystagmus waveforms from other eye movements (Golyandina et al., 2001). This method allows for an adaptive extraction of the trend of a signal. In analysing all the sub-series of the original signal for a given scale, common behaviours can be found. The trend component was then considered to be the component explaining the most the global variation of the signal (Figure C.1). One drawback of this method is that the trend can be scattered on several different computed components. In order to automatize the trend extraction step, a grouping step is needed. For the robustness of the trend extraction, the grouping step used a k-means algorithm to regroup similar components (Moreau et al., 2015b). The signals for the nystagmus movement were then analysed using traditional signal processing tools. For each eye, the main frequency of the movement was estimated, using the maximum of the correlogram. The same principle was used to compute the phase delay of the movements of the eyes, based on the maximum of the cross correlation function between the two movement signals.

C.3 Results

The patients were selected in three ways (Figure C.2). First, over the studied period of time, 43 children presented at the neuro-ophthalmology clinic with an OPG and were consequently followed up. Out of these, 11 children were diagnosed through a nystagmus. Second, over the same period of time, 181 infants were referred to the neuro-ophthalmology clinic for a nystagmus that had appeared before the age of two years. Thirty-seven of them were clinically classified as spasmus nutans-type and did not exhibit obvious signs of retinal dysfunction or dystrophy on the first evaluation –these including a family history of retinal dystrophy or stationary retinal dysfunction, obvious photophobia, oculo-digital sign, or abnormal fundus in favour of a retinal dystrophy. They all benefited from a brain imaging. In ten cases an OPG was diagnosed. Third, the same ten cases were also found when selecting the association "nystagmus with onset before two years" and "brain tumour". The difference between the three ways to identify these patients (11 with the first way vs 10 with the second and third way) can be explained by the fact that one of the children from the OPG group exhibited nystagmus at the age of three. This correspondence between these ways of identifying the patients also means that in all cases, nystagmus having led to the diagnosis of a cerebral tumour had been clinically classified as spasmus nutans-type (never as infantile nystagmus syndrome (INS) or fusion maldevelopment nystagmus syndrome (FMNS); Group 2001), and that in all cases the cerebral tumour was an OPG. In three cases, no interpretable recording was available –one of these three cases was also the one who exhibited a nystagmus at the age of three.

Eight children were therefore included. The characteristics of their nystagmus and tumours are summarised in Table C.1 to C.4. Age at nystagmus onset was 2.5-10 months (mean=5.8, median=5.5, SD=2.4). Delay between nystagmus onset and glioma diagnosis was 0-13 months (mean=1.9, median=0.5, SD=4.5). In three cases, initial clinical examination showed early signs of Russel diencephalic syndrome –associating variable degrees of weight loss, emaciation, hydrocephalus and euphoria– in association with the nystagmus. In two cases, frank optic atrophy was clinically obvious; in one case a papilledema was noticed; in five cases, however, the fundus was considered either strictly normal or possibly within normal limits. Although no clear crossed asymetry pattern was ever detected, it was not possible to get reproducible enough VEP to allow

for robust description and analysis of VEP topography. In one case, the classic triad of spasmus nutans syndrome was complete, with head tilt and head oscillations associated to the specific nystagmus; in two cases, the nystagmus was associated with head oscillations alone; in two other cases, with head tilt alone. In all cases, the nystagmus consisted in medium to high frequency (2.7-5 Hz, mean=3.7 Hz, median=3.6, SD=0.8), low amplitude, multidirectional, disconjugated (i.e. not in phase) and dissociated (i.e. not of the same amplitude) movements of the eyes. There was no difference in the waveforms between binocular and monocular viewing conditions or according to the gaze direction. The dissociation was clinically obvious to the human eye in the cases with the lowest frequency and/or in cases where the nystagmus was very asymetrical, at least at some stage of the evolution (four cases), to the point of being apparently unilateral, also only at some stage (two cases). In two cases, the dissociation was unsuspectable clinically, even on videos watched at real speed by oculomotor experts (MR, OZ and PPV). In all cases, however, disconjugacy was obvious on recordings, with most of the time a 180 degrees interocular phase difference between the horizontal components of the two eyes, which horizontally beat out of phase, while the vertical components beat in phase, resulting in a characteristic "convection-like" oscillatory movement of the eyes, with a frank dissociation between the two eyes, highly variable over time (Figure C.1). In four cases, phase variations were observed over the length of the recordings, occurring at irregular intervals and usually lasting for less than a second to a few seconds, without any apparent trigger. Whenever identifiable, the waveforms were always truly sinusoidal. In six cases, the nystagmus resolved between age 6.5 and age 24 months. In two cases, the nystagmus was still present, though rarely, at age 3 and 5 years. Concerning case six, after an initial period of three months, the nystagmus disappeared for a month, while visual function and general health decreased, before reappearing with visual function improvement.

In all cases but one, the tumour volume was big (>28x28x20mm, up to 51x30x47mm), always involving the chiasm (Table C.2 and Figure C.3). In six cases, post-gadolinium enhancement was seen in the tumour periphery, while in the centre, the anatomical shape of a thickened chiasm could be seen, without enhancement; in three cases among these six, the global tumour shape respected the form of the chiasm, leading to a four-leaf clover appearance. In two cases, there was global enhancement of the tumour and the chiasm was undistinguishable within the tumour. In two cases, histology was not available. In five out of six cases, the tumours were pilocytic astrocytomas, while in the sixth case it consisted in a pilo-mixoid astrocytoma grade II.

In all cases, a chemotherapy was initiated, according to the SIOP-LGG 2004 protocol (EU trial-20555) comparing classical induction with vincristine-carboplatine for 10 weeks with reinforced induction with vincristine-carboplatine-etoposide for 10 weeks, followed in allcases by vincristine and carboplatine for a total treatment duration of 18 months. In selected cases a debulking surgery was also realised, initially in three cases and secondarily in one case. Visual outcome was highly variable and could not be correlated with either the delay between nystagmus onset and OPG diagnostic, the characteristics of the tumour or the treatments administered: in two cases bilateral profound visual impairment was noticed over the two months following diagnosis, in three cases, profound visual impairment developed in one eye, while the fellow eye exhibited moderate to no visual impairment; in three cases, only little to moderate symmetrical visual impairment was noticed at last examination.

Children	Age at nystagmus onset (months)	Age at glioma diagnosis (months)	Associated general signs at diagnosis	Visual function	Optic discs at fundoscopy	VEP (+ made, - not made)
1	5.5	5.5	Head oscillations long before nystagmus onset	Normal (Teller cards)	Normal	+
2	2.5	3	Head tilt Collier sign	Normal behaviour	Doubtful minimal optic atrophy	-
3	10	23	-	Normal behaviour	Bilateral severe optic atrophy	+
4	4	4.5	Head oscillations, Head tilt, Minimal hydrocephalus, Collier sign, Corticotrope deficiency	Intermittent fixation	Papilledema	+
5	4	4	Hydrocephalus Bilateral VI nerve paresis	Normal (Teller cards)	Bilateral severe optic atrophy	+
6	7.5	8	Weight loss, Bulging fontanel, Corticotrope deficiency	Normal (Teller cards)	Doubtful minimal optic atrophy	+
7	7	8	Head tilt	Normal behaviour Teller cards: inferior to age norms	Normal	-
8	5.5	5.5	Weight loss, Head oscillations before nystagmus onset, Hydrocephalus, Collier sign, Panhypopituitarism	Normal behaviour	Doubtful minimal optic atrophy	-

Table C.1: Clinical characteristics at diagnosis

Children	Location	Gadolinium enhancement	Tumour size at diagnosis transverse x antero-post x sagittal (volume)	Metastasis	Cyst	Histologic type	*BRAF V600e*	*KIAA1549-BRAF*
1	Optic nerves, chiasm, optic tract	+Peripheral	Largest diameter of the optic nerve = 10mm	-	-	-	-	-
2	Optic nerves, chiasm, optic tract with infra-tentorial infiltration	+Peripheral	51x40x37 mm supra and infra tent	+	+	Pilocytic astrocytoma	*0*	*WT*
3	Optic nerves, chiasm, optic tract	+Peripheral	33x30x21 mm (10.4 ml)	-	-	Pilocytic astrocytoma	*NF*	*NF*
4	Optic nerves, chiasm, optic tract	+Peripheral	28x28x24 mm	-	-	Pilocytic astrocytoma	*NF*	*NF*
5	Optic nerves, chiasm	+Peripheral	36x29x26 mm	-	+ frontal (big)	-	-	-
6	Optic nerves, chiasm, optic tract	+Global, uniform	42x38x36 mm and cyst 60x35 mm	-	+	Pilocytic astrocytoma	-	-
7	Chiasm, optic tract	+Peripheral	45x30mm	+ bulb, T1 and T6	+	Pilocytic astrocytoma	-	*Biopsy made on metastasis*
8	Optic nerves, chiasm, optic tract	+ Global, uniform	51x34x44 mm	-	+ (small)	Pilo-myxoid astrocytoma Grade II	-	-

Table C.2: Imaging and histologic characteristics

Children	Treatments: first line	Treatments: second line	Treatments: third line and following	Visual outcome (age at last examination)
1	SIOP LGG 2004 Stable then MRI progression	Velbe 1/week	-	OD: >0,2 R2; OS: severe amblyopia (4 y-o)
2	SIOP LGG 2004	VCR CPM/VCR Cisplatine	Debulking, Cystic derivation, Vinblastine, Avastin-Irinotecan	Profound visual impairment within the two months following diagnosis; NLP; severe bilateral optic atrophy (2 y-o)
3	SIOP LGG 2004	-	-	1.1/10 R12 // 1.5/10 R3 (5 y-o)
4	SIOP LGG 2004	TPCV	-	Profound visual impairment within the two months following diagnosis; NLP; major photophobia; severe bilateral optic atrophy (2.5 y-o)
5	Debulking SIOP LGG 2004	-	-	> 1.6/10 R1/3 // 1.6/10 R1/3 (4 y-o)
6	Debulking SIOP LGG 2004	-	-	Profound visual impairment OD within the two months following diagnosis; OD: NLP; OG: normal behaviour (> R10) (2.5 y-o)
7	SIOP LGG 2004 (1 year only, parents decision)	Re-evolution age 2.5; no treatment; spontaneous regression	-	Moderate visual impairment in both eyes, no amblyopia (4 y-o)
8	Debulking SIOP LGG 2004 (10 weeks) TPCV	TPCV	-	Light perception OD / Follows small targets OS; photophobia; severe bilateral optic atrophy, OD>OS (2 y-o)

Table C.3: Treatments characteristics and visual outcome (*TPCV*: Thioguanine, Procarbazine, CCNU or Lomustine, Vincristine *NLP*: no light perception)

Children	Nystagmus mean frequency (Hz)	Multi-directional	EOM/ Phase difference/ Phase shift	Laterality	Age at nystagmus resolution	Oculomotor anomalies after initial nystagmus resolution
1	3	+	-/+/+	Bilateral, initially RE>LE then LE>RE	Still present age 3 years	-
2	5	+	-/+/?	ODS then OD>OS	7 months	Stability, then searching nystagmus and III nerve paresis with Xt
3	3.6	+	+ NOC monoc NI /+/+	Right eye alone first, then bilateral RE>LE	Still rarely present age 5 years	-
4	3.6	+ pseudo-R	+ NOC bof mais voit-il le stim ?/+/+	OD>OG sur EOM 180° OP	6.5 months	Stability, then searching nystagmus and Xt
5	4.5	+	+ Ininterpretable +/?	-	24 months	-
6	3	+	+ NOC binoc NI/+/?	RE>LE	18 months	Et
7	4	+	-/+/?	-	15 months	-
8	2.7	+ pseudo-R	-/+/+	initially RE then RE>LE	12 months	Variable Xt Crossed fixation with LE

Table C.4: Nystagmus characteristics (*Xt:* exotropia; *Et:* esotropia)

C.4 Discussion

We describe here a series of patients diagnosed with OPG through nystagmus. All of them were between 2.5 and 10-month-old when the nystagmus was first noted. In all cases, the nystagmus characteristics were similar: multidirectional, medium to high frequency, low amplitude, highly variable over time, disconjugate and dissociated. Such nystagmus could not clinically be confused with an INS or a FMNS. Both eyes were rarely oscillating with independent frequency; most of the time they exhibited a "convection-like" oscillatory movement, with a 180 degrees horizontal phase shift and no vertical phase shift. In all cases, the OPG were Dodge grade 2 or 3, centred by the chiasm, with always peripheral or global gadolinium enhancement, which constitutes a specific subpopulation of OPG. Two cases out of eight were metastatic, which is also unusual. Eleven cases of tumours were diagnosed among the 37 cases of nystagmus clinically classified as spasmus nutans-type. All were OPG. By contrast, no case of brain tumour was diagnosed within the 143 children with other types of early-onset nystagmus. The high incidence of spasmus nutans (and hence of tumours) in this series likely reflects the selection bias of a tertiary referral paediatric hospital.

This study also allows for three practical considerations regarding nystagmus and OPG. First, age cannot be discriminant in differentiating secondary nystagmus, since in at least one case the nystagmus was present before 12 weeks of age, and in at least five cases, the nystagmus was present before six months of age –and could therefore be considered "early onset". Not only can a nystagmus present before 12 months be associated with an OPG, but such early onset is the rule for these nystagmus. Second, the presence of associated head oscillations is not specific of idiopathic spasmus nutans-type nystagmus –unlike what was recently suggested (Hertle & Dell'Osso, 2013)–, since it was noticed in three cases in this series. Third, in all cases but one, visual behaviour was normal at onset, and in the majority of cases (five out eight patients), despite the large volume of the glioma, the fundus was initially considered within normal limits. Therefore, considering that in all cases treatment was indicated from diagnosis, and considering the relative rarity of such nystagmus as opposed to INS and FMNS, the controversy as to whether imaging should be performed in cases of spasmus nutans-type nystagmus even if the visual function and the fundus are considered normal (Arnoldi & Tychsen, 1995; Lee, 1996; Newman et al., 1990) can be solved: we recommend urgent imaging to be also performed in all cases of spasmus nutans-type nystagmus, unless signs of retinal dystrophy or dysfunction are present.

Farmer & Hoyt (1984) reported that in asymmetrical OPG-associated nystagmus, the most oscillating eye was not necessarily the one with the lowest vision. A quick analysis of the present series would reach similar conclusions, as case one, who eventually developed severe amblyopia in the left eye, first exhibited an asymmetrical nystagmus with larger amplitude in the right eye. However, at that stage the vision was evaluated as being normal, while when a frank left amblyopia developed, the nystagmus asymmetry switched, with a larger nystagmus amplitude in the left eye. In the two other cases where unilateral amblyopia developed, the nystagmus also predominated in the amblyopic eye.

How can the peculiar nystagmus associated with these eight cases of OPG be described and classified? In all cases, the nystagmus characteristics fitted the usual definition of spasmus nutans-type nystagmus: pendular, medium (3-4 Hz) or high frequency (5Hz), low amplitude, multidirectional, with a dissociation between both eyes' movements.

Here we do not consider the frequency limit criterion proposed by the CEMAS (Group, 2001), which almost never fits with cases of published spasmus nutans cases –in Weissman study, no case had a frequency >10Hz (Weissman et al., 1987); in Gottlob study, only two cases out of ten had a frequency >10Hz (Gottlob et al., 1995). Although the frequencies observed here are in the high range of nystagmus frequencies observed in INS at similar ages –and in the low range of spasmus nutans-type nystagmus– they are clearly not discriminant. However, the absence of any jerk component in lateral gaze –the nystagmus always consisting of pendular oscillations whatever the position of gaze–, allows to easily rule out clinically the hypothesis of both an INS or an FMNS. So do the often predominant vertical component and also the dissociation of both eye movements, which was clinically obvious in three cases with medium frequency: one unilateral, one very asymetrical –both observed during initial examination–, plus one case where the nystagmus was reported to have been initially unilateral. In other cases, the high frequency and low amplitude characteristics of the nystagmus precluded any clinical analysis of the interocular phase relationship. Hence, based on clinical characteristics alone, the nystagmus was always classified within the category of spasmus nutans-type nystagmus. Notwithstanding the debate around what "spasmus nutans" should designate –the idiopathic form of a given syndrome, or the syndrome itself whatever its cause–, the definition of spasmus nutans-type nystagmus itself is challenging: for most authors, it is a clinical diagnostic as defined above, while for others, what defines spasmus nutans is the interocular phase difference (that is, the dissociation between the right and the left eye oscillations) and the variability of this difference over time (Weissman et al., 1987), which most often requires one or several formal oculomotor recordings. An interocular phase difference was assessed in all cases reported here, most of the time a 180 degrees phase shift; in four cases a variability of this phase difference over time could also be assessed. It is likely that with repeated oculomotor recordings, similar phase shifts would have been found in the four cases where they were not identified. Based on these results, no difference therefore allowed to distinguish between OPG gliomas and spasmus nutans-type nystagmus, even when applying recording-based definitions. The "convection" pattern of the OPG-associated nystagmus does not only fit with the characteristics of spasmus nutans-type nystagmus; it has also been reported in "convergent-divergent acquired pendular nystagmus", which has been described in a few adult patients (Averbuch-Heller et al., 1995; Galvez-Ruiz et al., 2011; Gresty et al., 1982; Mossman et al., 1990; Schwartz et al., 1986; Sharpe et al., 1975; Yang et al., 2006). What mechanism can give rise to such a nystagmus? Although in the present series, when an amblyopia was present, the amblyopic eye exhibited larger nystagmus amplitudes, the hypothesis of low vision as the cause for the nystagmus is unlikely: in all cases but one, there was still no sign of low vision at nystagmus onset and in many cases even for months after onset. Instead, here the nystagmus preceded the variable visual loss, while vision-loss nystagmus follows severe vision loss after a variable delay. Additionally, such pattern has not been described in other instances of vision loss nystagmus: early monocular vision loss can give rise to binocular, jerk nystagmus beating away from the amblyopic eye, which characteristics are similar to a hemi-FMNS (Kushnner, 1995) – the syndrome of monocular infantile blindness with bilateral nystagmus–, but also, more rarely, to monocular, pendular, high-frequency and low-amplitude, horizontal, monocular nystagmus (Good et al., 1993); late monocular vision loss can give rise to monocular, low-frequency and low-amplitude, vertical, monocular nystagmus – the so-called Heimann-Bielschowsky phenomenon– (Smith et al., 1982); binocular vision loss can lead to continuous jerk nystagmus, with horizontal,

vertical and torsional components – also called "searching nystagmus" – but also, more rarely, to medium-frequency and medium-amplitude, horizontal, binocular, symmetrical nystagmus (Good et al., 1997), or even more rarely, to seesaw nystagmus, mainly but not always, in the case of lesions at the optic chiasm (May & Truxal, 1997). None of these four categories of nystagmus could be mistaken for the nystagmus described here.

Another hypothesis would rely on the early crossing alterations induced by a chiasmal lesion. According to a recent hypothesis, disruption in the negative feedback function of the physiological optokinetic nystagmus system with inversion of the retinal slip, might be the cause of several varieties of early onset nystagmus (Huang et al., 2011). In almost all cases, such disruption results from constitutional misrouting of the ganglion cells, such as in oculo- cutaneous albinism or achiasma. The OPG from the present series represent the most early set of OPG, with an acquired, early disturbance of the normal repartition between crossing and uncrossing ganglion cells at the level of the chiasm. Like chiasmatic compression, chiasmatic infiltration –be it tumoural or inflammatory– can cause bitemporal hemianopia, by affecting more selectively the crossed axonal fibres. The most consistent hypothesis for this susceptibility of the nasal fibres to compression is based on structural collapse theories as applied to crossing vs non-crossing fibers (McIlwaine et al., 2005). The nystagmus associated with OPG could be a model of acquired, early disruption of the optokinetic system, interfering with the calibration of the visual system during the sensitive period of visual development. Such imperfectly calibrated oculomotor system could precisely produce a pendular nystagmus –one achiasmatic zebrafish belladonna mutant was shown to exhibit a pendular-waveform nystagmus (Huang et al., 2011). However, this hypothesis is unlikely for two reasons. First, we failed to show any consistent crossed asymmetry in the five children we recorded with VEPs. This, however, may be due to a lack of robustness in the technique used, since the stimuli were displayed on a small screen, while bigger ones allows for better reproducibility in VEPs in the infant age set (Thompson & Liasis, 2012). Second, this hypothesis does not account for the 180 degree phase difference between both eyes horizontal components: both practically and theoretically, such a failure in the calibration of the oculomotor system would give rise to conjugated nystagmus.

As synthesised by Averbuch-Heller et al. (1995), the two theoretical possibilities to give rise to the pattern of a "convergent-divergent acquired pendular nystagmus" are either a dysfunction in the normal yoking mechanisms of the version system with a 180 degree phase shift – as also proposed by some authors as an explanation for spasmus nutans (Weissman et al., 1987)–, or oscillations in the vergence system (Schwartz et al., 1986; Sharpe et al., 1975). Here, the OPG-associated nystagmus is also dissociated, which also points towards vergence movements. Although variations in the phase interactions between the two eyes were recorded, those were always brief (often lasting for less than a second), compared to the large majority of the recordings where the horizontal phase shift was 180 degree. The strict conjugacy of the vertical component also makes the first possibility unlikely. The response of the vergence system is traditionally believed to be slow. Most reported cases of convergent-divergent acquired pendular nystagmus had low velocities and frequencies (around 1Hz) and therefore fitted with this possibility (Schwartz et al., 1986; Sharpe et al., 1975). Experiments have shown, however, that the slow vergence system could oscillate at frequencies up to 2.5 Hz (Hung et al., 1986), while in two cases from Averbuch-Heller et al. (1995), the nystagmus frequency was as high as 6 Hz. The slow vergence system is a negative feedback system and may oscillate either through an increase in gain or delay within the system internal feedback

loops, or through external oscillation imposing upon the system (Averbuch-Heller et al., 1995). The high frequencies recorded in most of our patients suggest either pathological changes increasing the gain or decreasing the delay within this system, or external oscillations. While most pathological lesions –such as decrease in the myeline conduction speed– would increase the delay, a loss of inhibitory connections could increase it. However, such oscillations would then be electively provoked when the vergence system is operating, that is during near vision, which is not what was observed in our patients, unless one postulates that the lack of maturity of the vergence system at a young age could account for such frequent instability and result in a highly-variable-in-time pattern of oscillations. The other possibility would imply an external, linear, neuronal oscillator –for instance within the cerebellum– projecting to the slow vergence system (Averbuch-Heller et al., 1995). Another hypothesis would implicate the recently studied fast vergence system, which is responsible for the fast vergence movements occurring during rapid eye movement sleep (Cullen & Van Horn, 2011; Escudero & Vidal, 1996); however, its neural bases are still a matter of debate; these mouvements are disconjugate, which could possibly account for the observed variations in the phase shift over time.

How can oscillations in the vergence system result from the presence of an OPG? Within the slow vergence system, chiasmal gliomas strongly affect all the visual afferences of the nucleus reticularis tegmenti pontis (NRTP), through the superior colliculi and through the frontal eye fields (Figure C.4). The NRTP projects to the fastigial nucleus, the dentate nucleus and the nucleus of the raphe interpositus (RIP). These are reciprocate, mainly inhibitory connections. The RIP is a central element within the vergence system, with efferences to the medial recti via the supra-oculomotor area. Instability in the feedback loops between the NRTP and its cerebellar efferences, mainly the RIP, could lead to sinusoidal oscillations in the vergence system. This was experimentally shown in monkeys with selective lesions of the NRTP, who exhibited convergent-divergent oscillations (Gamlin & Mitchell, 1993). Furthermore, it was showed that the RIP could produce theta oscillations under experimental conditions in rabbits (Hoffmann & Berry, 2009). The frequency of theta oscillations in infants is known to range from 3.6 to 5.6 Hz, which is also close to the frequencies of OPG- associated nystagmus (Orekhova et al., 2006). The next question is: if this specific oscillation pattern results from such a common process, why would it only be noticed in infants with early chiasmal gliomas? A theoretically possible mechanism to consider –either in isolation or in conjunction with the one following– could be the contribution of some metabolic changes occurring in these children through pituitary compression and possibly acting on membrane proteins at the level of the previously discussed brainstem or cerebellar nuclei involved in the control of vergence, although no such common change can be identified. Anatomic considerations, however, suffice in providing a plausible pathophysiological mechanism: early chiasmal gliomas actually represent a unique pathological situation, where an early tumour grows and alters the visual pathways at the very age of a critical period of visual development, between three and nine months of age; hence probably the homogenous time of onset of this nystagmus. Earlier or later, similar processes do not give rise to the same nystagmus. Early chiasmal gliomas are one of the very few conditions –if not the only one– and hence the model of a process, which progressively and partly affects vision at the beginning of the critical period of visual development. The understanding of its clinical expressions is therefore of considerable interest for visual neuroscience and should benefit from ongoing larger studies in the field.

C.5 Figures

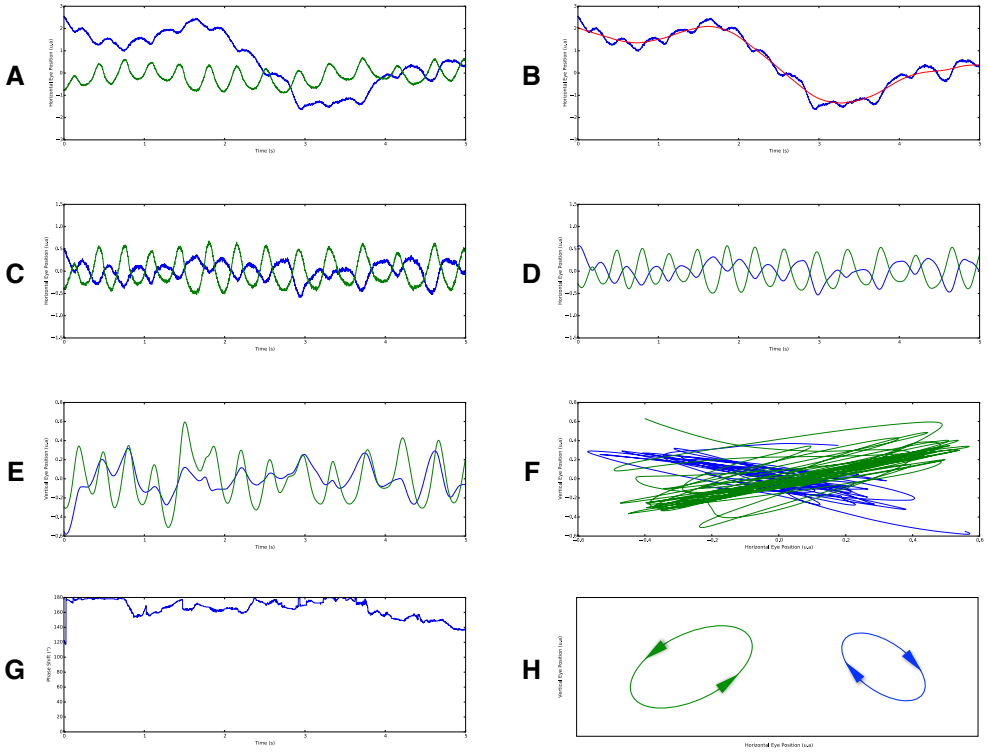


Figure C.1: Oculographic characteristics of typical optic pathway glioma-associated nystagmus. (A) Raw signal of the horizontal component of the nystagmus recorded with the Eyefant*: right eye in green, left eye in blue. (B) Extraction of the signal trend (in red), using singular spectrum analysis. (C) Horizontal nystagmus signal. (D) Smoothened horizontal nystagmus signal using a second order butterworth lowpass filter with cutting frequency 35Hz; both eyes are oscillating in the horizontal plane with a 180degree phase difference. (E) Smoothened vertical nystagmus signal; both eyes are oscillating in phase in the vertical plane. (F) Raw signal of both horizontal and vertical components of the nystagmus superimposed during the five considered seconds; the nystagmus direction appears to be mainly oblique. (G) Phase shift of the horizontal component of the nystagmus. (H) Schematic representation of the nystagmus represented as the position of both eyes in the coronal plane: the movement of both eyes can be compared with the convection movement of water molecules, going up and towards the center, then down and away from the center; right eye amplitude is larger than left eye amplitudes.

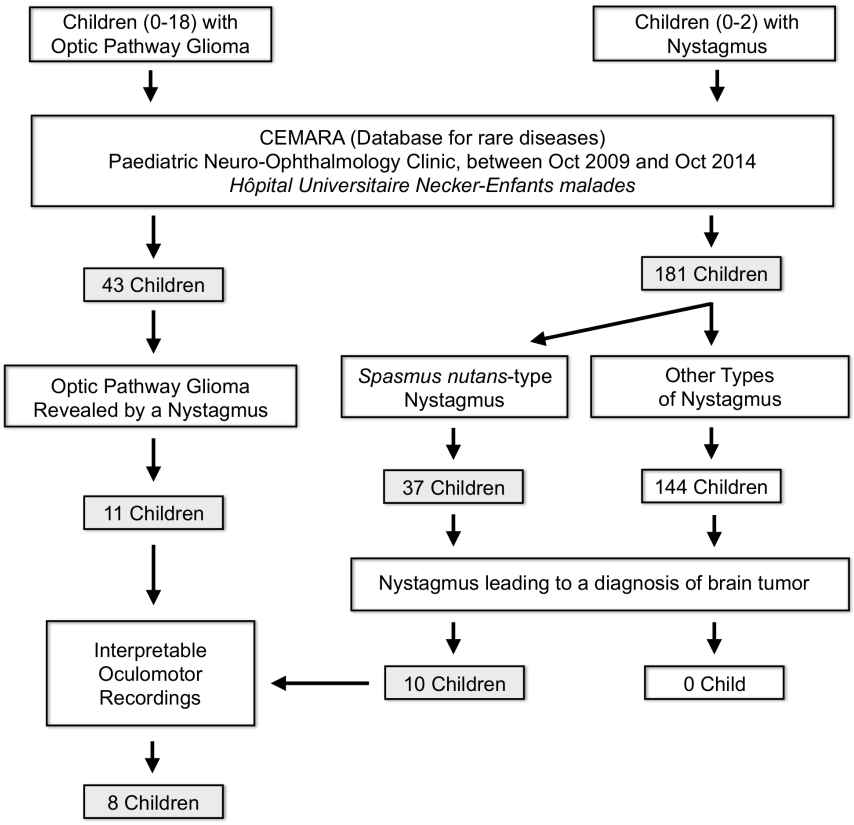


Figure C.2: Schematic representation of the patients' inclusion process

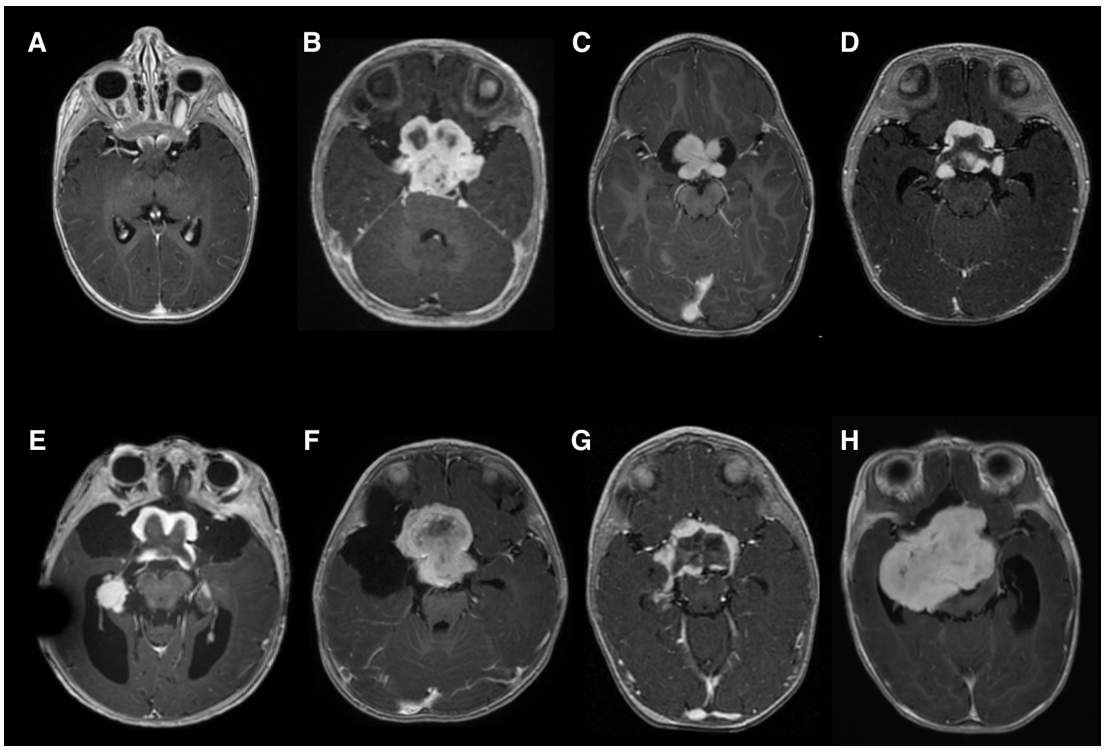


Figure C.3: Gadolinium-enhanced T1 axial imaging of the optic pathway gliomas. (A) Patient 1. (B) Patient 2. (C) Patient 3. (D) Patient 4. (E) Patient 5. (F) Patient 6. (G) Patient 7. (H) Patient 8.

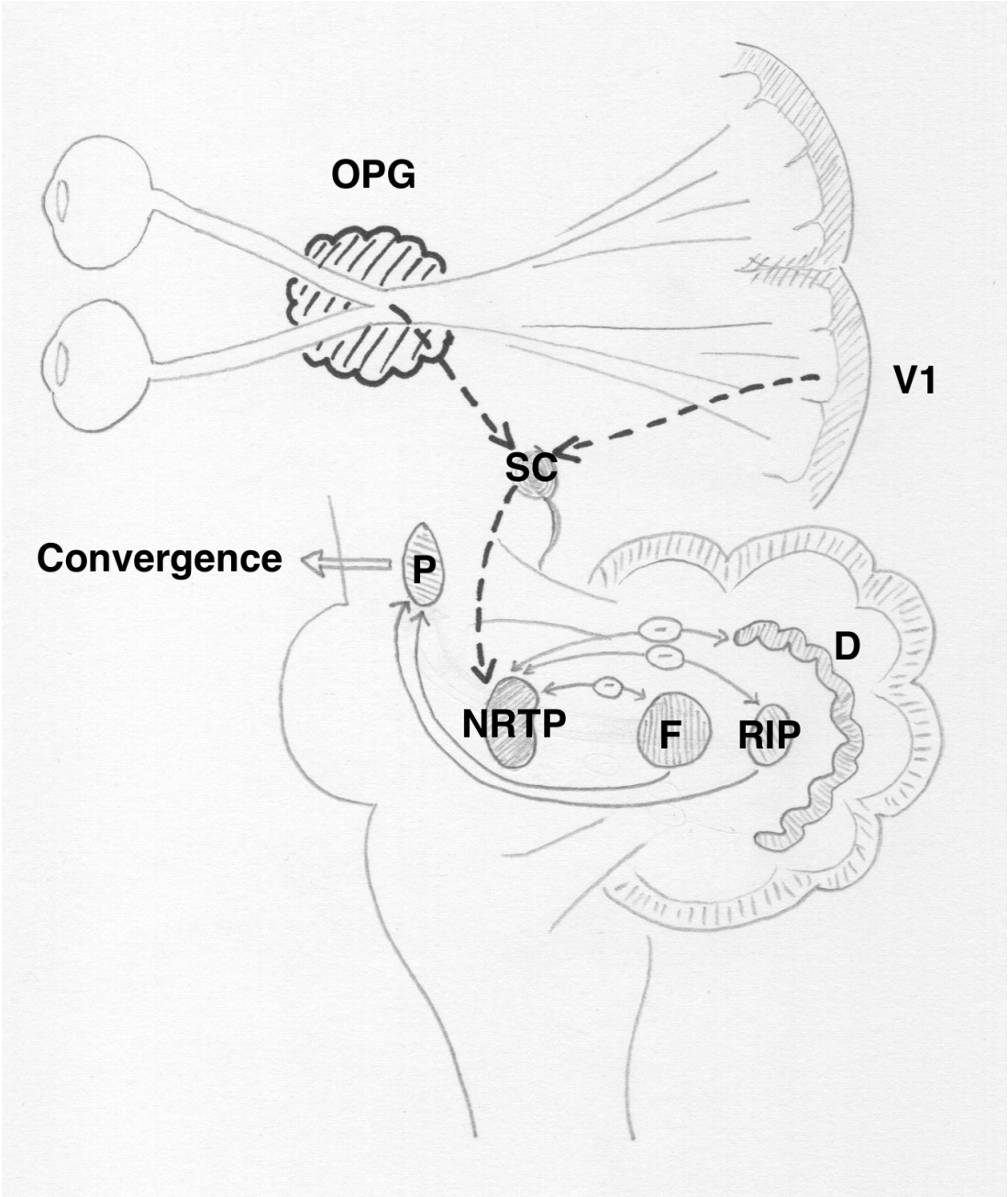


Figure C.4: Schematic representation of the disruption in the vergence system circuitry possibly leading to the specific OPG-associated nystagmus. Afferences to the nucleus reticularis tegmenti pontis (NRTP) are affected by big chiasmal gliomas at the age of the sensitive period of visual development; this can result in instability in the feedback loop between the NRTP and its cerebellar efferences and lead to sinusoidal oscillations in the vergence system.

Bibliography

Arnoldi, K. and Tychsen, L. Prevalence of intracranial lesions in children initially diagnosed with disconjugate nystagmus (spasmus nutans). *Journal of Pediatric Ophthalmology and Strabismus*, Sep-Oct(32):296–301, 1995

Averbuch-Heller, L., Zivotofsky, A., Remler, B., Das, V., Dell'Osso, L. F., and Leigh, R. Convergent- divergent pendular nystagmus: possible role of the vergence system. *Neurology*, Mar(45):509–515, 1995

Brodsky, M. and Keating, G. Chiasmal glioma in spasmus nutans: a cautionary note. *Neuro-Opthalmology*, Sep(34):274–275, 2014

Cullen, K. E. and Van Horn, M. R. The neural control of fast vs. slow vergence eye movements. *European Journal of Neuroscience*, Jun(33):2147–2154, 2011

Donin, J. Acquired monocular nystagmus in children. *Canadian Journal of Ophthalmology*, Jul(2):212–215, 1967

Escudero, M. and Vidal, P.-P. A quantitative study of electroencephalography, eye movements and neck electromyography characterizing the sleep-wake cycle of the guinea-pig. *European Journal of Neuroscience*, Mar(8):572–580, 1996

Farmer, J. and Hoyt, C. Monocular nystagmus in infancy and early childhood. *American Journal of Ophthalmology*, Oct(98):504–509, 1984

Galvez-Ruiz, A., Roig, C., Muñoz, S., and Arruga, J. Convergent-divergent nystagmus as a manifestation of oculopalatal tremor. *Neuro-Opthalmology*, 35:276–279, 2011

Gamlin, P. and Mitchell, K. Reversible lesions of nucleus reticularis tegmenti pontis affect convergence and ocular accommodation. *Society of Neuroscience Abstr.*, pp. 346, 1993

Golyandina, N., Nekrutkin, V., and Zhigljavsky, A. A. *Analysis of Time Series Structure: SSA and Related Techniques*. CRC Press, 2001

Good, W., Jan, J., Hoyt, C., Billson, F., Schoettker, P., and Klaeger, K. Monocular vision loss can cause bilateral nystagmus in young children. *Developmental Medicine and Child Neurology*, Jun(39):421–424, 1997

Good, W., Koch, T., and Jan, J. Monocular nystagmus caused by unilateral anterior visual- pathway disease. *Developmental Medicine and Child Neurology*, Dec(35):1106–1110, 1993

Gottlob, I., Wizov, S., and Reinecke, R. Spasmus nutans. A long-term follow-up. *Investigative Ophthalmology and Visual Science*, Dec(36):2768–2771, 1995

Gresty, M., Ell, J., and Findley, L. Acquired pendular nystagmus: its characteristics, localising value and pathophysiology. *Journal of Neurology, Neurosurgery and Psychiatry*, May(45):431–439, 1982

Group, C. A national eye institute sponsored workshop and publication on the classification of eye movement abnormalities and strabismus (CEMAS). *The National Eye Institute Publications*, 2001

Hertle, R. W. and Dell'Osso, L. F. *Nystagmus in infancy and childhood: current concepts in mechanisms, diagnoses, and management*. Oxford University Press, 2013

Hoffmann, L. and Berry, S. Cerebellar theta oscillations are synchronized during hippocampal theta-contingent trace conditioning. In *Proceedings of the National Academy of Sciences of the United States of America*, pp. 21371–21376, 2009

Huang, M., CC, C., Huber-Reggi, S., Neuhauss, S., and Straumann, D. Comparison of infantile nystagmus syndrome in achiasmatic zebrafish and humans. *Annals of the New-York Academy of Science*, Sep(1233):285–291, 2011

Hung, G., Semmlow, J., and Ciuffreda, K. A dual-mode dynamic model of the vergence eye movement system. *IEEE Transaction on Biomedical Engineering*, Nov(33):1021–1028, 1986

Kelly, T. Optic glioma presenting as spasmus nutans. *Pediatrics*, Feb(45):295–296, 1970

Kushnner, B. Infantile uniocular blindness with bilateral nystagmus. A syndrome. *Archive Ophthalmology*, Oct(113):1298–1300, 1995

Lavery, M., O'Neill, J., Chu, F., and Martyn, L. Acquired nystagmus in early childhood: a presenting sign of intracranial tumor. *Ophthalmology*, May(91):425–453, 1984

Lee, A. Neuroimaging in all cases of spasmus nutans. *Journal of Pediatric Ophthalmology and Strabismus*, Jan-Feb(33):68–69, 1996

Leigh, R. J. and Zee, D. S. *The neurology of eye movements*. Oxford University Press, USA, 2015

May, E. and Truxal, A. Loss of vision alone may result in seesaw nystagmus. *Journal of Neuro-Ophthalmology*, Jun(17):84–85, 1997

McIlwaine, G., Carrim, Z., Lueck, C., and Chrisp, T. A mechanical theory to account for bitemporal hemianopia from chiasmal compression. *Journal of Neuro-Ophthalmology*, Mar(25):40–43, 2005

Moreau, T., Oudre, L., and Vayatis, N. Groupement automatique pour l ' analyse du spectre singulier. In *Proceedings of the Groupe de Recherche et d'Etudes en Traitement du Signal et des Images (GRETSI)*, 2015b

Mossman, S., Bronstein, A. M., Gresty, M., Kendall, B., and Rudge, P. Convergence nystagmus associated with Arnold-Chiari malformation. *Archive Ophthalmology*, MAr (47):357–359, 1990

Newman, S., Hedges, T., Wall, M., and Sedwick, L. Spasmus nutans– or is it? *Survey of Ophthalmology*, May-Jun(34):453–456, 1990

Odom, J., Bach, M., Brigell, M., Holder, G., McCulloch, D., Tormene, A., and Et, A. ISCEV standard for clinical visual evoked potentials. *Documenta Ophthalmologica*, Feb (120):111–9, 2010

Orekhova, E., Stroganova, T., Posikera, I., and Elam, M. EEG theta rhythm in infants and preschool children. *Clinical Neurophysiology*, MAy(17):1047–62, 2006

Raudnitz, R. W. Zur Lehre von Spasmus Nutans. *Jb Kinderheilkd*, 45:145, 1897

Schwartz, M., Selhorst, J., Ochs, A., Beck, R., Campbell, W., HArris, J., and Et, A. Oculomasticatory myorhythmia: a unique movement disorder occurring in Whipple's disease. *Annals of Neurology*, Dec(20):677–83, 1986

Sharpe, J., Hoyt, W., and Rosenberg, M. Convergence-evoked nystagmus. Congenital and acquired forms. *Archive of Neurology*, Mar(32):191–4, 1975

Smith, J., Flynn, J., and Spiro, H. Monocular vertical oscillations of amblyopia: The Heimann- Bielschowsky phenomenon. *Journal of Clinical Neuro-Ophthalmology*, Jun (2):85–91, 1982

Thompson, D. and Liasis, A. Visual electrophysiology: how it can help you and your patient. *Journal of Pediatric Ophthalmology and Strabismus*, 4:55–62, 2012

Toledano, H., Muhsinoglu, O., Luckman, J., Goldenberg-Cohen, N., and Michowiz, S. Acquired nystagmus as the initial presenting sign of chiasmal glioma in young children. *European Journal of Pediatric Neurology*, Nov(19):694–700, 2015

Weissman, B., Dell'Osso, L., Abel, L., and Leigh, R. Spasmus nutans. A quantitative prospective study. *Archive of Ophthalmology*, Apr(105):525–8, 1987

Yang, S., Jeong, J., Kim, J., and Yoon, Y. Progressive venous stasis retinopathy and open-angle glaucoma associated with primary pulmonary hypertension. *Ophthalmic surgery Lasers Imaging*, May-Jun(37):230–3, 2006

Bibliography

Abalov, N. V. and Gubarev, V. V. Automated grouping of decomposition components of time series for singular spectrum analysis. In *Proceedings of the International Forum on Strategic Technology (IFOST)*, Cox's Bazar, Bangladesh, 2014.

Adler, A., Elad, M., Hel-Or, Y., and Rivlin, E. Sparse Coding with Anomaly Detection. In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 22 – 25, Southampton, United Kingdom, 2013.

Afsari, B., Chaudhry, R., Ravichandran, A., and Vidal, R. Group action induced distances for averaging and clustering Linear Dynamical Systems with applications to the analysis of dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2208–2215, Providence, RI, USA, 2012.

Agarwal, A. *Computational Trade-offs in Statistical Learning*. PhD thesis, University of California, Berkeley, 2012.

Agarwal, A., Anandkumar, A., and Netrapalli, P. A Clustering Approach to Learn Sparsely-Used Overcomplete Dictionaries. *arXiv preprint*, arXiv:1309(1952), 2013.

Agarwal, A., Anandkumar, A., and Jain, P. Learning sparsely used overcomplete dictionaries via alternating minimization. *Journal of Machine Learning Research (JMLR)*, 35:1–15, 2014.

Aharon, M. and Elad, M. Sparse and Redundant Modeling of Image Content Using an Image-Signature-Dictionary. *SIAM Journal on Imaging Sciences*, 1(3):228–247, 2008.

Aharon, M., Elad, M., and Bruckstein, A. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transaction on Signal Processing*, 54(11):4311–4322, 2006.

Ahmed, N., Natarajan, T., and Rao, K. Discrete Cosine Transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974.

Alaoui, A. and Mahoney, M. W. Fast randomized kernel ridge regression with statistical guarantees. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 775–783, 2015.

Alexandrov, T. and Golyandina, N. Automatic extraction and forecast of time series cyclic components within the framework of SSA. In *Proceedings of the Workshop on Simulation*, pp. 45–50, St. Petersburg, Russia, 2005.

Álvarez-Meza, A. M., Acosta-Medina, C. D., and Castellanos-Domínguez, G. Automatic Singular Spectrum Analysis for Time-Series Decomposition. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pp. 131–136, Bruges, Belgium, 2013.

Andén, J. and Mallat, S. Deep Scattering Spectrum. *IEEE Transaction on Signal Processing*, 62(16):4114–4128, 2014.

Andoni, A., Panigrahy, R., Valiant, G., and Zhang, L. Learning Polynomials with Neural Networks. *Journal of Machine Learning Research (JMLR)*, 32(2):1908–1916, 2014.

Arnoldi, K. and Tychsen, L. Prevalence of intracranial lesions in children initially diagnosed with disconjugate nystagmus (spasmus nutans). *Journal of Pediatric Ophthalmology and Strabismus*, Sep-Oct(32):296–301, 1995.

Arora, S., Ge, R., and Moitra, A. New Algorithms for Learning Incoherent and Overcomplete Dictionaries. *Journal of Machine Learning Research (JMLR)*, 35:1–20, 2013.

Auvinet, B., Berrut, G., Touzard, C., Moutel, L., Collet, N., Chaleil, D., and Barrey, E. Reference data for normal subjects obtained with an accelerometric device. *Gait & posture*, 16(2):124–134, 2002.

Averbuch-Heller, L., Zivotofsky, A., Remler, B., Das, V., Dell'Osso, L. F., and Leigh, R. Convergent- divergent pendular nystagmus: possible role of the vergence system. *Neurology*, Mar(45):509–515, 1995.

Ayachi, F., Nguyen, H., Goubault, E., Boissy, P., and Duval, C. The Use of Empirical Mode Decomposition-Based Algorithm and Inertial Measurement Units to Auto-Detect Daily Living Activities of Healthy Adults. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(10):1060–1070, 2016.

Bahl, L. and Jelinek, F. Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition. *IEEE Transactions on Information Theory*, 21(4):404–411, 1975.

Baldi, P. and Hornik, K. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989.

Barrois, R., Gregory, T., Oudre, L., Moreau, T., Truong, C., Pulini, A. A., Vienne, A., Labourdette, C., Vayatis, N., Buffat, S., Yelnik, A., De Waele, C., Laporte, S., Vidal, P. P., and Ricard, D. An automated recording method in clinical consultation to rate the limp in lower limb osteoarthritis. *PLoS ONE*, 11(10):e0164975, 2016.

Barrois, R., Oudre, L., Moreau, T., Truong, C., Vayatis, N., Buffat, S., Yelnik, A., de Waele, C., Gregory, T., Laporte, S., and Others. Quantify osteoarthritis gait at the doctor's office: a simple pelvis accelerometer based method independent from footwear and aging. *Computer methods in biomechanics and biomedical engineering*, 18(Sup1):1880–1881, 2015.

Barron, A. Universal Approximation Bounds for Superpositions of a Sigmoidal Function. *IEEE Transaction on Information Theory*, 39(3):930–945, 1993.

Bartlett, P. L. and Maass, W. Vapnik-Chervonenkis Dimension of Neural Nets. *The handbook of brain theory and neural networks*, pp. 1188–1192, 2003.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. Speeded-Up Robust Features. *Computer Vision and Image Understanding (CVIU)*, 110(September):346–359, 2008.

Beck, A. and Teboulle, M. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Bellman, R. On the Theory of Dynamic Programming. *Proceedings of the National Academy of Sciences of the United States of America*, 38(8):716–719, 1952.

Ben Mansour, K., Rezzoug, N., and Gorce, P. Comparison between several locations of gyroscope for gait events detection. *Computer methods in biomechanics and biomedical engineering*, pp. 1–2, 2015.

Bengio, Y. and LeCun, Y. Scaling learning algorithms towards AI. *Large-scale kernel machines*, 34(5):1–41, 2007.

Bottou, L. and Bousquet, O. Learning using large datasets. *Mining Massive DataSets for Security*, 3, 2008.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

Bradley, J. K., Kyrola, A., Bickson, D., and Guestrin, C. Parallel Coordinate Descent for ℓ_1-Regularized Loss Minimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 321–328, Bellevue, WA, USA, 2011.

Brady, M. L., Raghavan, R., and Slawny, J. Back Propagation Fails to Separate Where Perceptrons Succeed. *IEEE Transactions on Circuits and Systems*, 36(5):665–674, 1989.

Brajdic, A. and Harle, R. Walk detection and step counting on unconstrained smartphones. In *Proceedings of the ACM international joint conference on Pervasive and ubiquitous computing*, pp. 225–234, Zurich, Switzerland, 2013. ACM.

Bristow, H., Eriksson, A., and Lucey, S. Fast convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 391–398, Portland, OR, USA, 2013.

Brodsky, M. and Keating, G. Chiasmal glioma in spasmus nutans: a cautionary note. *Neuro-Opthalmology*, Sep(34):274–275, 2014.

Bruna, J. and Mallat, S. Invariant Scattering Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1872–1886, 2013.

Bubeck, S. Convex Optimization: Algorithms and Complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

Buchstaber, V. M. Time series analysis and grassmannians. *Translations of the American Mathematical Society-Series 2*, 162:1–18, 1994.

Cadieu, C. F. and Olshausen, B. A. Learning intermediate-level representations of form and motion from natural movies. *Neural computation*, 24(4):827–866, 2012.

Candes, E. J., Li, X., Ma, Y., and Wright, J. Robust Principal Component Analysis? *Journal of the Association for Computing Machinery (JACM)*, 58(3):11, 2011.

Caruana, R., Lawrence, S., and Giles, L. Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 402–408, Vancouver, Canada, 2001.

Chalasani, R., Principe, J. C., and Ramakrishnan, N. A fast proximal method for convolutional sparse coding. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–5, Dallas, TX, USA, 2013.

Chan, A. B. and Vasconcelos, N. Probabilistic kernels for the classification of auto-regressive visual processes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 846–851, San Diego, CA, USA, 2005.

Chandrasekaran, V. and Jordan, M. I. Computational and statistical tradeoffs via convex relaxation. *Proceedings of the National Academy of Sciences*, 110(13):E1181—-E1190, 2013.

Chaudhari, P. and Soatto, S. On the energy landscape of deep networks. *arXiv preprint*, arXiv:1511(06485), 2015.

Chaudhari, P., Choromanska, A., Soatto, S., Lecun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-SGD: Biasing Gradient Descent into Wide Valleys. In *Proceedings of the International Conference on Learning Representation (ICLR)*, Toulon, France, 2017a.

Chaudhari, P., Oberman, A., Osher, S., Soatto, S., and Carlier, G. Deep relaxation: partial differential equations for optimizing deep neural networks. *arXiv preprint*, arXiv:1704(04932), 2017b.

Cho, K., Courville, A., and Bengio, Y. Describing Multimedia Content Using Attention-Based Encoder-Decoder Networks. *IEEE Transactions on Multimedia*, 17(11):1875–1886, 2015.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The Loss Surfaces of Multilayer Networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 192—-204, San Diego, CA, USA, 2015.

Coates, A. and Ng, A. Y. The importance of encoding versus training with sparse coding and vector quantization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 921–928, Bellevue, WA, USA, 2011.

Cohen, N., Sharir, O., Levine, Y., Tamari, R., Yakira, D., and Shashua, A. Analysis and Design of Convolutional Networks via Hierarchical Tensor Decompositions. *arXiv preprint*, arXiv:1705(02302), 2017.

Combettes, P. L. and Bauschke, H. H. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2011.

Cullen, K. E. and Van Horn, M. R. The neural control of fast vs. slow vergence eye movements. *European Journal of Neuroscience*, Jun(33):2147–2154, 2011.

Cuturi, M. and Blondel, M. Soft-DTW: a Differentiable Loss Function for Time-Series. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 894–903, Sidney, Australia, 2017.

Cybenko, G. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.

Dahl, G. E., Sainath, T. N., and Hinton, G. E. Improving deep neural networks for LVCSR using rectified linear units and dropout. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8609–8613, Vancouver, Canada, 2013. IEEE.

Dalal, N. and Triggs, B. Histograms of Oriented Gradients for Human Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 886–893, San Diego, CA, USA, 2005.

Daubechies, I., Defrise, M., and De Mol, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

Dauphin, Y., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 2933–2941, Montreal, Canada, 2014.

De Cock, K. and De Moor, B. Subspace angles and distances between ARMA models. *System and Control Letter*, 46(4):265–270, 2002.

Dijkstra, B., Zijlstra, W., Scherder, E., and Kamsma, Y. Detection of walking periods and number of steps in older adults and patients with Parkinson's disease: accuracy of a pedometer and an accelerometry-based method. *Age and ageing*, 37(4):436–441, 2008.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp Minima Can Generalize For Deep Nets. *arXiv preprint*, arXiv:1703(04933), 2017.

Donin, J. Acquired monocular nystagmus in children. *Canadian Journal of Ophthalmology*, Jul(2):212–215, 1967.

Donoho, D. L. and Elad, M. Maximal Sparsity Representation via l 1 Minimization. *submitted to IEEE Transactions on Information Theory*, pp. 1–28, 2002.

Duchi, J., Hazan, E., and Singer, Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research (JMLR)*, 12: 2121–2159, 2011.

Eigen, D., Rolfe, J., Fergus, R., and LeCun, Y. Understanding Deep Architectures using a Recursive Convolutional Network. In *Proceedings of the International Conference on Learning Representation (ICLR)*, pp. 1–8, 2014.

El Ghaoui, L., Viallon, V., and Rabbani, T. Safe feature elimination for the LASSO and sparse supervised learning problems. *Journal of Machine Learning Research (JMLR)*, 8(4):667–698, 2012.

Elhamifar, E., Sapiro, G., and Vidal, R. See all by looking at a few: Sparse modeling for finding representative objects. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1600–1607, 2012.

Engan, K., Aase, S. O., and Husøy, J. H. Method of Optimal Directions for Frame Design. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2443—-2446, Phoenix, AZ, USA, 1999.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research (JMLR)*, 11(Feb):625–660, 2010.

Escudero, M. and Vidal, P.-P. A quantitative study of electroencephalography, eye movements and neck electromyography characterizing the sleep-wake cycle of the guinea-pig. *European Journal of Neuroscience*, Mar(8):572–580, 1996.

Farmer, J. and Hoyt, C. Monocular nystagmus in infancy and early childhood. *American Journal of Ophthalmology*, Oct(98):504–509, 1984.

Fercoq, O., Gramfort, A., and Salmon, J. Mind the duality gap : safer rules for the Lasso. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 333–342, Lille, France, 2015.

Fortune, E., Lugade, V., Morrow, M., and Kaufman, K. Step counts using a tri-axial accelerometer during activity. In *Proceedings of the American Society of Biomechanics Annual Meeting (ASB)*, Gainesville, FL, USA, 2012.

Frasconi, P., Gori, M., and Tesi, A. Successes and failures of backpropagation: A theoretical investigation. *Progress in Neural Networks: Architecture*, 5(265):42 – 47, 1997.

Freeman, C. D. and Bruna, J. Topology and Geometry of Deep Rectified Network Optimization Landscapes. In *Proceedings of the International Conference on Learning Representation (ICLR)*, Toulon, France, 2017.

Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

Friedman, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, 32(200): 675–701, 1937.

Fuchs, J. J. On sparse representations in arbitrary redundant bases. *IEEE Transactions on Information Theory*, 50(6):1341–1344, 2004.

Gabay, D. and Mercier, B. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

Gabor, D. Theory of Communication. *Journal of the Institution of Electrical Engineers*, 93(26)(26):429–457, 1946.

Galvez-Ruiz, A., Roig, C., Muñoz, S., and Arruga, J. Convergent-divergent nystagmus as a manifestation of oculopalatal tremor. *Neuro-Opthalmology*, 35:276–279, 2011.

Gamlin, P. and Mitchell, K. Reversible lesions of nucleus reticularis tegmenti pontis affect convergence and ocular accommodation. *Society of Neuroscience Abstr.*, pp. 346, 1993.

Gilles, J. Empirical wavelet transform. *IEEE Transactions on Signal Processing*, 61 (16):3999–4010, 2013.

Gillis, N. *Nonnegative matrix factorization: Complexity, algorithms and applications*. PhD thesis, Université catholique de Louvain, Louvain-La-Neuve, Belgium, 2011.

Giryes, R., Eldar, Y. C., Bronstein, A. M., and Sapiro, G. Tradeoffs between Convergence Speed and Reconstruction Accuracy in Inverse Problems. *arXiv preprint*, arXiv:1605(09232), 2016a.

Giryes, R., Sapiro, G., Bronstein, A. M., and Carolina, N. Deep Neural Networks with Random Gaussian Weights: A Universal Classification Strategy? *IEEE Transaction on Signal Processing*, 64(13):3444–3457, 2016b.

Golyandina, N. and Korobeynikov, A. Basic Singular Spectrum Analysis and Forecasting with R. *Computational Statistics & Data Analysis*, 71:934—-954, 2014.

Golyandina, N., Nekrutkin, V., and Zhigljavsky, A. A. *Analysis of Time Series Structure: SSA and Related Techniques*. CRC Press, 2001.

Good, W., Koch, T., and Jan, J. Monocular nystagmus caused by unilateral anterior visual- pathway disease. *Developmental Medicine and Child Neurology*, Dec(35):1106–1110, 1993.

Good, W., Jan, J., Hoyt, C., Billson, F., Schoettker, P., and Klaeger, K. Monocular vision loss can cause bilateral nystagmus in young children. *Developmental Medicine and Child Neurology*, Jun(39):421–424, 1997.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

Gori, M. and Tesi, A. Backpropagation converges for multi-layered networks and linearly-separable patterns. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, volume 2, pp. 896, Seattle, WA, USA, 1991. IEEE.

Gorodnitsky, I. F. and Rao, B. D. Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing*, 45(3):600–616, 1997.

Goroshin, R. *Unsupervised Feature Learning in Computer Vision*. PhD thesis, New-York University, 2015.

Gottlob, I., Wizov, S., and Reinecke, R. Spasmus nutans. A long-term follow-up. *Investigative Ophthalmology and Visual Science*, Dec(36):2768–2771, 1995.

Gregor, K. and Lecun, Y. Learning Fast Approximations of Sparse Coding Karol. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 399–406, Haifa, Israel, 2010.

Gresty, M., Ell, J., and Findley, L. Acquired pendular nystagmus: its characteristics, localising value and pathophysiology. *Journal of Neurology, Neurosurgery and Psychiatry*, May(45):431–439, 1982.

Gribonval, R., Jenatton, R., Bach, F., Kleinsteuber, M., and Seibert, M. Sample complexity of dictionary learning and other matrix factorizations. *IEEE Transactions on Information Theory*, 61(6):3469–3486, 2015.

Grosse, R., Raina, R., Kwong, H., and Ng, A. Y. Shift-Invariant Sparse Coding for Audio Classification. *Cortex*, 8:9, 2007.

Group, C. A national eye institute sponsored workshop and publication on the classification of eye movement abnormalities and strabismus (CEMAS). *The National Eye Institute Publications*, 2001.

Hadsell, R., Erkan, A., Sermanet, P., Scoffier, M., Muller, U., and LeCun, Y. Deep belief net learning in a long-range vision system for autonomous off-road driving. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 628–633. IEEE, 2008.

Haeffele, B. D. and Vidal, R. Global Optimality in Tensor Factorization, Deep Learning, and Beyond. *arXiv preprint*, arXiv:1506(07540), 2015.

Haeffele, B. D. and Vidal, R. Global Optimality in Neural Network Training. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7331–7339, Honolulu, HI, USA, 2017.

Haeffele, B. D., Stahl, R., Vanmeerbeeck, G., and Vidal, R. Efficient Reconstruction of Holographic Lens-Free Images by Sparse Phase Recovery. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 109–117, Quebec, Canada, 2017.

Hahnloser, R. H., Sarpeshkar, R., Mahowald, M. a., Douglas, R. J., and Seung, H. S. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947–951, 2000.

Hahnloser, R. H. R., Seung, H. S., and Slotine, J.-J. Permitted and Forbidden Sets in Symmetric Threshold-Linear Networks. *Neural Computation*, 15(3):621–638, 2003.

Han, J., Jeon, H. S., Jeon, B. S., and Park, K. S. Gait detection from three dimensional acceleration signals of ankles for the patients with Parkinson's disease. In *Proceedings of the International Special Topic Conference on Information Technology in Biomedicine (ITAB)*, Ioannina, Greece, 2006.

Harris, Z. S. Distributional Structure. *WORD*, 10(2-3):146–162, 1954.

Hastie, T., Tibshirani, R., and Wainwright, M. J. *Statistical Learning with Sparsity*. CRC Press, 2015.

He, B. S., Yang, H., and Wang, S. L. Alternating Direction Method with Self-Adaptive Penalty Parameters for Monotone Variational Inequalities. *Journal of Optimization Theory and Applications*, 106(2):337–356, 2000.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016.

Hertle, R. W. and Dell'Osso, L. F. *Nystagmus in infancy and childhood: current concepts in mechanisms, diagnoses, and management*. Oxford University Press, 2013.

Hesterberg, T., Choi, N. H., Meier, L., and Fraley, C. Least angle and ℓ_1 penalized regression: A review. *Statistics Surveys*, 2:61–93, 2008.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

Hinton, G. E. and Salakhutdinov, R. R. Using deep belief nets to learn covariance kernels for Gaussian processes. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 1249–1256, Vancouver, Canada, 2008.

Hinton, G. E., Osindero, S., and Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Hinton, G. E., Srivastava, N., and Swersky, K. Lecture 6a- overview of mini-batch gradient descent. Slide for online class COURSERA: Neural Networks for Machine Learning, 2012.

Hiriart-Urruty, J. B. How to regularize a difference of convex functions. *Journal of Mathematical Analysis and Applications*, 162(1):196–209, 1991.

Hoffmann, L. and Berry, S. Cerebellar theta oscillations are synchronized during hippocampal theta-contingent trace conditioning. In *Proceedings of the National Academy of Sciences of the United States of America*, pp. 21371–21376, 2009.

Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

Hotelling, H. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

Hu, R., Barnard, M., and Collomosse, J. Gradient Field Descriptor for Sketch Based Retrieval and Localization. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 1025–1028, Hong Kong, China, 2010.

Huang, M., CC, C., Huber-Reggi, S., Neuhauss, S., and Straumann, D. Comparison of infantile nystagmus syndrome in achiasmatic zebrafish and humans. *Annals of the New-York Academy of Science*, Sep(1233):285–291, 2011.

Hung, G., Semmlow, J., and Ciuffreda, K. A dual-mode dynamic model of the vergence eye movement system. *IEEE Transaction on Biomedical Engineering*, Nov(33):1021–1028, 1986.

Janzamin, M., Sedghi, H., and Anandkumar, A. Beating the Perils of Non-Convexity: Guaranteed Training of Neural Networks using Tensor Methods. *arXiv preprint*, arXiv:1506(08473), 2015.

Jas, M., Dupré La Tour, T., Şimşekli, U., and Gramfort, A. Learning the Morphology of Brain Signals Using Alpha-Stable Convolutional Sparse Coding. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 1–15, Long Beach, CA, USA, 2017.

Jia Deng, Wei Dong, Socher, R., Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, Miami Beach, FL, USA, 2009.

Jiang, Z., Lin, Z., Davis, L. S., Incorporated, A. S., and Jose, S. Learning A Discriminative Dictionary for Sparse Coding via Label Consistent K-SVD. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1697–1704, Colorado Spring, CO, USA, 2011.

Johnson, T. and Guestrin, C. Blitz: A Principled Meta-Algorithm for Scaling Sparse Optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1171–1179, Lille, France, 2015.

Jutten, C. and Herault, J. Blind seperation of sources, part I: an adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10, 1991.

Kadri, H., DUFLOS, E., Preux, P., canu, S., Rakotomamonjy, A., and Audiffren, J. Operator-valued Kernels for Learning from Functional Response Data. *Journal of Machine Learning Research (JMLR)*, 2015.

Kavukcuoglu, K., Sermanet, P., Boureau, Y.-l., Gregor, K., and Lecun, Y. Learning Convolutional Feature Hierarchies for Visual Recognition. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 1090–1098, Vancouver, Canada, 2010.

Kelly, T. Optic glioma presenting as spasmus nutans. *Pediatrics*, Feb(45):295–296, 1970.

Keogh, E., Chu, S., Hart, D., and Pazzani, M. An online algorithm for segmenting time series. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pp. 289–296, San Jose, United States, 2001. IEEE.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. In *Proceedings of the International Conference on Learning Representation (ICLR)*, Toulon, France, 2017.

Kim, J., Jang, H., Hwang, D.-H., and Park, C. A step, stride and heading determination for the pedestrian navigation system. *Journal of Global Positioning Systems*, 3(1-2): 273–289, 2004.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representation (ICLR)*, pp. 1–10, San Diego, CA, USA, 2015.

Knopp, J., Prasad, M., Willems, G., Timofte, R., and Van Gool, L. Hough transform and 3D SURF for robust three dimensional classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 589–602, Heraklion, Crete, Greece, 2010.

Krishnaprasad, P. S. On families of systems and deformations. *International Journal of Control,*, 38(5):1055–1079, 1983.

Krizhevsky, A. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Advances in neural information processing systems (NIPS)*, pp. 1097–1105, South Lake Tahoe, United States, 2012.

Kushnner, B. Infantile uniocular blindness with bilateral nystagmus. A syndrome. *Archive Ophthalmology*, Oct(113):1298–1300, 1995.

Larochelle, H., Erhan, D., Courville, A., Bergstra, J., and Bengio, Y. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 473–480, Corvallis, United States, 2007. ACM.

Larochelle, H., Bengio, Y., Louradour, J., and Lamblin, P. Exploring strategies for training deep neural networks. *Journal of Machine Learning Research (JMLR)*, 10 (Jan):1–40, 2009.

Lavery, M., O'Neill, J., Chu, F., and Martyn, L. Acquired nystagmus in early childhood: a presenting sign of intracranial tumor. *Ophthalmology*, May(91):425–453, 1984.

Le, Q. V., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J., and Ng, A. Y. Building high-level features using large scale unsupervised learning. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8595–8598, Vancouver, Canada, 2013.

Lee, A. Neuroimaging in all cases of spasmus nutans. *Journal of Pediatric Ophthalmology and Strabismus*, Jan-Feb(33):68–69, 1996.

Lee, H., Battle, A., Raina, R., and Ng, A. Y. Efficient Sparse coding algorithms. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 801–808, Vancouver, Canada, 2007.

Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1–8, Montreal, Canada, 2009.

Lee, J. D., Simchowitz, M., Jordan, M. I., and Recht, B. Gradient Descent Converges to Minimizers. In *Proceedings of the Conference on Learning Theory (COLT)*, pp. 1246–1257, New-York, NY, USA, 2016.

Leigh, R. J. and Zee, D. S. *The neurology of eye movements*. Oxford University Press, USA, 2015.

Liang, T., Poggio, T., Rakhlin, A., and Stokes, J. Fisher-Rao Metric, Geometry, and Complexity of Neural Networks. *arXiv preprint*, arXiv:1711(01530), 2017.

Libby, R. A Simple Method for Reliable Footstep Detection in Embedded Sensor Platforms. Research report, 2012.

Liu, G., Lin, Z., and Yu, Y. Robust subspace segmentation via low-rank representation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 663–670, Haifa, Israel, 2010.

Liu, J., Wright, S. J., Ré, C., Bittorf, V., and Sridhar, S. An asynchronous parallel stochastic coordinate descent algorithm. *Journal of Machine Learning Research (JMLR)*, 16:285–322, 2015.

Liu, J., Garcia-Cardona, C., Wohlberg, B., and Yin, W. Online Convolutional Dictionary Learning. *arXiv preprint*, arXiv:1709(00106), 2017.

Lowe, D. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pp. 1150–1157, Corfu, Greece, 1999. IEEE.

Luo, Z. Q. and Tseng, P. Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46-47(1):157–178, 1993.

Macqueen, J. Some methods for classification and analysis of multivariate observations. *Proceedings of the 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1(233):281–297, 1967.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. Discriminative Learned Dictionaries for Local Image Analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, Anchorage, AK, USA, 2008.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. Online Learning for Matrix Factorization and Sparse Coding. *Journal of Machine Learning Research (JMLR)*, 11(1):19–60, 2010.

Mairal, J., Bach, F., and Ponce, J. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):791–804, 2012.

Mallat, S. *A Wavelet Tour of Signal Processing*. Academic press, 2008.

Mallat, S. Group Invariant Scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

Mallat, S. Understanding Deep Convolutional Networks. *Philosophical Transaction of the Royal Society A*, 374(2065), 2016.

Mallat, S. and Zhang, Z. Matching Pursuits With Time-Frequency Dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330, 1993.

Mariani, B. *Assessment of Foot Signature Using Wearable Sensors for Clinical Gait Analysis and Real-Time Activity Recognition*. PhD thesis, EPFL, 2012.

Marschollek, M., Goevercin, M., Wolf, K.-H., Song, B., Gietzelt, M., Haux, R., and Steinhagen-Thiessen, E. A performance comparison of accelerometry-based step detection algorithms on a large, non-laboratory sample of healthy and mobility-impaired persons. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, pp. 1319–1322, Vancouver, Canada, 2008.

Martin, R. J. Metric for ARMA processes. *IEEE Transactions on Signal Processing*, 48(4):1164–1170, 2000.

May, E. and Truxal, A. Loss of vision alone may result in seesaw nystagmus. *Journal of Neuro-Ophthalmology*, Jun(17):84–85, 1997.

McCulloch, W. S. and Pitts, W. A Logical Calculus of the Idea Immanent in Nervous Activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.

McIlwaine, G., Carrim, Z., Lueck, C., and Chrisp, T. A mechanical theory to account for bitemporal hemianopia from chiasmal compression. *Journal of Neuro-Ophthalmology*, Mar(25):40–43, 2005.

Mikolajczyk, K., Schmid, C., and A, C. S. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

Mladenov, M. and Mock, M. A step counter service for Java-enabled devices using a built-in accelerometer. In *Proceedings of the International Workshop on Context-Aware Middleware and Services (COMSWARE)*, pp. 1–5, Dublin, Ireland, 2009. ACM.

Montavon, G., Braun, M. L., and Müller, K.-R. Kernel analysis of deep networks. *Journal of Machine Learning Research (JMLR)*, 12(Sep):2563–2581, 2011.

Montavon, G., Samek, W., and Müller, K.-r. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

Montúfar, G. F., Pascanu, R., Cho, K., and Bengio, Y. On the Number of Linear Regions of Deep Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 2924–2932, Montreal, Canada, 2014.

Moreau, T. and Audiffren, J. Post Training in Deep Learning with Last Kernel. *arXiv preprint*, arXiv:1611(04499), 2016.

Moreau, T. and Bruna, J. Understanding Neural Sparse Coding with Matrix Factorization. In *Proceedings of the International Conference on Learning Representation (ICLR)*, 2017.

Moreau, T., Oudre, L., and Vayatis, N. Distributed Convolutional Sparse Coding via Message Passing Interface (MPI). In *Proceedings of the NIPS Workshop on Non-parametric Methods for Large Scale Representation Learning*, 2015a.

Moreau, T., Oudre, L., and Vayatis, N. Groupement automatique pour l ' analyse du spectre singulier. In *Proceedings of the Groupe de Recherche et d'Etudes en Traitement du Signal et des Images (GRETSI)*, 2015b.

Moreau, T., Oudre, L., and Vayatis, N. Distributed Convolutional Sparse Coding. *arXiv preprint*, arXiv:1705(10087), 2017.

Morgan, N. and Bourlard, H. *Generalization and parameter estimation in feedforward nets: Some experiments*. International Computer Science Institute, Denver, United States, 1990.

Mossman, S., Bronstein, A. M., Gresty, M., Kendall, B., and Rudge, P. Convergence nystagmus associated with Arnold-Chiari malformation. *Archive Ophthalmology*, MAr (47):357–359, 1990.

Naik, G. R. and Kumar, D. K. An Overview of Independent Component Analysis and Its Applications. *Informatica*, 35:63–81, 2011.

Naqvi, N. Z., Kumar, A., Chauhan, A., and Sahni, K. Step Counting Using Smartphone-Based Accelerometer. *International Journal on Computer Science and Engineering (IJCSE)*, 4(5):675–682, 2012.

Nesterov, Y. A method of solving a convex programming problem with convergence rate O (1/k2). *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

Nesterov, Y. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

Nesterov, Y. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Newman, S., Hedges, T., Wall, M., and Sedwick, L. Spasmus nutans– or is it? *Survey of Ophthalmology*, May-Jun(34):453–456, 1990.

Neyshabur, B. Implicit Regularization in Deep Learning. PhD Thesis, 2017.

Neyshabur, B., Salakhutdinov, R., and Srebro, N. Path-SGD: Path-Normalized Optimization in Deep Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 2422–2430, Montreal, Canada, 2015.

Nutini, J., Schmidt, M., Laradji, I. H., Friedlander, M. P., and Koepke, H. Coordinate Descent Converges Faster with the Gauss-Southwell Rule Than Random Selection. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1632–1641, Lille, France, 2015.

Odom, J., Bach, M., Brigell, M., Holder, G., McCulloch, D., Tormene, A., and Et, A. ISCEV standard for clinical visual evoked potentials. *Documenta Ophthalmologica*, Feb(120):111–9, 2010.

Olah, C., Mordvintsev, A., and Schubert, L. Feature Visualization. *Distill*, 2017.

Olshausen, B. a. and Field, D. J. Sparse coding with an incomplete basis set: a strategy employed by V1, 1997.

Oner, M., Pulcifer-Stump, J., Seeling, P., and Kaya, T. Towards the run and walk activity classification through Step detection-An Android application. In *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1980–1983, San Diego, CA, USA, 2012.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1717–1724, Colombus, OH, USA, 2014.

Orekhova, E., Stroganova, T., Posikera, I., and Elam, M. EEG theta rhythm in infants and preschool children. *Clinical Neurophysiology*, MAy(17):1047–62, 2006.

Osher, S. and Li, Y. Coordinate descent optimization for ℓ_1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Problems and Imaging*, 3 (3):487–503, 2009.

Oudre, L., Jakubowicz, J., Bianchi, P., and Simon, C. Classification of periodic activities using the Wasserstein distance. *IEEE Transactions on Biomedical Engineering*, 59 (6):1610–1619, 2012.

Oudre, L., Moreau, T., Truong, C., Barrois-Müller, R., Dadashi, R., and Grégory, T. Détection de pas à partir de données d'accélérométrie. In *Proceedings of the Groupe de Recherche et d'Etudes en Traitement du Signal et des Images (GRETSI)*, Lyon, France, 2015.

Oymak, S., Recht, B., and Soltanolkotabi, M. Sharp Time–Data Tradeoffs for Linear Inverse Problems. *arXiv preprint*, arXiv:1507(04793), 2015.

Paatero, P. and Tapper, U. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

Pan, J. and Tompkins, W. J. A real-time QRS detection algorithm. *IEEE Transactions on Biomedical Engineering*, 32(3):230–236, 1985.

Papyan, V., Sulam, J., and Elad, M. Working Locally Thinking Globally - Part II: Theoretical Guarantees for Convolutional Sparse Coding. *arXiv preprint*, arXiv:1607 (02009), 2016.

Papyan, V., Sulam, J., and Elad, M. Working Locally Thinking Globally: Theoretical Guarantees for Convolutional Sparse Coding. *IEEE Transactions on Signal Processing*, 65(21):5687–5701, 2017.

Pati, Y., Rezaiifar, R., and Krishnaprasad, P. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, pp. 40–44, Pacific Grove, CA, USA, 1993.

Poultney, C., Chopra, S., Cun, Y. L., and Others. Efficient learning of sparse representations with an energy-based model. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 1137–1144, Vancouver, Canada, 2006.

Pratt, W. K., Kane, J., and Andrews, H. C. Hadamard transform image coding. *Proceedings of the IEEE*, 57(1):58–68, 1969.

Qiu, G. Indexing chromatic and achromatic patterns for content-based colour image retrieval. *Pattern Recognition*, 35(8):1675–1686, 2002.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 1177–1184, Vancouver, Canada, 2007.

Rakotomamonjy, A. and Gasso, G. Histogram of gradients of time-frequency representations for audio scene classification. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(1):142–153, 2015.

Raudnitz, R. W. Zur Lehre von Spasmus Nutans. *Jb Kinderheilkd*, 45:145, 1897.

Razavian, A. S., Azizpour, H., Sullivan, J., and Carlsson, S. CNN Features off-the-Shelf: an Astounding Baseline for Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 806–813, Colombus, OH, USA, 2014.

Renaudin, V., Susi, M., and Lachapelle, G. Step length estimation using handheld inertial sensors. *Sensors*, 12(7):8507–8525, 2012.

Robert, M., Contal, E., Moreau, T., Vayatis, N., and Vidal, P.-P. The Why and How of Recording Eye Movement from Very Early Childhood. Oral Presentation, Gordon Research Conference on Eye Movement, 2015.

Robert, M. P., Grill, J., Moreau, T., Grevent, D., Zambrowsky, O., Varlet, P., Contal, E., Martin, G., Brémond-Gignac, D., Ingster-Moati, I., Dufour, C., Brugières, L., Vayatis, N., Boddaert, N., Sainte-Rose, C., Blauwblomme, T., Puget, S., and Vidal, P.-P. Optic pathway gliomas-associated nystagmus. *submitted to Brain*, 2016.

Rockafellar, R. T. Monotone Operators and the Proximal Point Algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

Rozell, C. J., Johnson, D. H., Baraniuk, R. G., and Olshausen, B. A. Sparse coding via thresholding and local competition in neural circuits. *Neural Computation*, 20(10): 2526–63, 2008.

Saigo, H., Vert, J. P., Ueda, N., and Akutsu, T. Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11):1682–1689, 2004.

Sakoe, H. and Chiba, S. A Dynamic Programming Approach to Continuous Speech Recognition. In *Proceedings of the International Congress on Acoustics*, volume 3, pp. 65–69, Budapest, Hungary, 1971. Akadémiai Kiadó.

Salakhutdinov, R. and Hinton, G. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978, 2009.

Salarian, A., Russmann, H., Vingerhoets, F., Dehollain, C., Blanc, Y., Burkhard, P., and Aminian, K. Gait assessment in Parkinson's disease: toward an ambulatory system for long-term monitoring. *IEEE Transactions on Biomedical Engineering*, 51 (8):1434–1443, 2004.

Scherrer, C., Halappanavar, M., Tewari, A., and Haglin, D. Scaling Up Coordinate Descent Algorithms for Large ℓ_1 Regularization Problems. Technical report, Pacific Northwest National Laboratory (PNNL), 2012a.

Scherrer, C., Tewari, A., Halappanavar, M., and Haglin, D. J. Feature Clustering for Accelerating Parallel Coordinate Descent. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 28–36, South Lake Tahoe, United States, 2012b.

Schölkopf, B., Herbrich, R., and Smola, A. J. A generalized representer theorem. In *Proceedings of the International Conference on Computational Learning Theory (COLT)*, pp. 416–426. Springer, 2001.

Schüldt, C., Laptev, I., and Caputo, B. Recognizing human actions: A local SVM approach. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pp. 32–36, Cambridge, UK, 2004.

Schwartz, M., Selhorst, J., Ochs, A., Beck, R., Campbell, W., HArris, J., and Et, A. Oculomasticatory myorhythmia: a unique movement disorder occurring in Whipple's disease. *Annals of Neurology*, Dec(20):677–83, 1986.

Schwartz-ziv, R. and Tishby, N. Opening the black box of Deep Neural Networks via Information. *arXiv preprint*, arXiv:1703(00810), 2017.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In *Proceedings of the International Conference on Learning Representation (ICLR)*, 2014.

Shalev-Shwartz, S. and Tewari, A. Stochastic Methods for ℓ_1-regularized Loss Minimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 929–936, Montreal, Canada, 2009.

Sharpe, J., Hoyt, W., and Rosenberg, M. Convergence-evoked nystagmus. Congenital and acquired forms. *Archive of Neurology*, Mar(32):191–4, 1975.

Sifre, L. Rigid-Motion Scattering For Image Classification. PhD Thesis, 2014.

Smith, J., Flynn, J., and Spiro, H. Monocular vertical oscillations of amblyopia: The Heimann- Bielschowsky phenomenon. *Journal of Clinical Neuro-Ophthalmology*, Jun (2):85–91, 1982.

Sokolic, J., Giryes, R., Sapiro, G., and Rodrigues, M. R. D. Robust Large Margin Deep Neural Networks. *arXiv preprint*, arXiv:1605(08254), 2016.

Sokolic, J., Giryes, R., Sapiro, G., and Rodrigues, M. R. D. Generalization Error of Invariant Classifiers. In *Proceedings of the Artificial Intelligence and Statistics (AISTAT)*, pp. 1094–1103, 2017.

Song, D. and Gupta, A. K. L_p-norm Uniform Distribution. *The American Mathematical Society*, 125(2):595–601, 1997.

Sprechmann, P., Bronstein, A., and Sapiro, G. Learning Efficient Structured Sparse Models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 615–622, Edinburgh, Great Britain, 2012.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research (JMLR)*, 15:1929–1958, 2014.

Su, W., Boyd, S., and Candes, E. J. A Differential Equation for Modeling Nesterov's Accelerated Gradient Method: Theory and Insights. *Journal of Machine Learning Research (JMLR)*, 17(153):1–43, 2016.

Telgarsky, M. Benefits of depth in neural networks. In *Proceedings of the Conference on Learning Theory (COLT)*, pp. 1–23, New-York, NY, USA, 2016.

Thompson, D. and Liasis, A. Visual electrophysiology: how it can help you and your patient. *Journal of Pediatric Ophthalmology and Strabismus*, 4:55–62, 2012.

Thüer, G. and Verwimp, T. Step detection algorithms for accelerometers. Master's thesis, Artesis University College of Antwerp, Belgium, 2008.

Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society: Series B (statistical methodology)*, 58(1):267—-288, 1996.

Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J., and Tibshirani, R. Strong rules for discarding predictors in Lasso- type problems. *Journal of the Royal Statistical Society: Series B (statistical methodology)*, 74(2):245–266, 2012.

Tishby, N. and Zaslavsky, N. Deep Learning and the Information Bottleneck Principle. In *Proceedings of the IEEE Information Theory Workshop (ITW)*, 2015.

Toledano, H., Muhsinoglu, O., Luckman, J., Goldenberg-Cohen, N., and Michowiz, S. Acquired nystagmus as the initial presenting sign of chiasmal glioma in young children. *European Journal of Pediatric Neurology*, Nov(19):694–700, 2015.

Tran, K., Le, T., and Dinh, T. A high-accuracy step counting algorithm for iPhones using accelerometer. In *Proceedings of the International Symposium on Signal Processing and Information Technology (ISSPIT)*, pp. 213–217, Ho Chi Minh City, Vietnam, 2012. IEEE.

Tsanas, A., Little, M. A., McSharry, P. E., and Ramig, L. O. Accurate telemonitoring of Parkinson's disease progression by noninvasive speech tests. *IEEE Transactions on Biomedical Engineering*, 57(4):884–893, 2010.

Tseng, P. Coordinate Ascent for Maximizing Nondifferentiable Concave Functions. Technical report, Laboratory for Information and Decision Systems (LIDS), MIT, 1988.

Van Overschee, P. and De Moor, B. N4SID: Subspace Algorithms for the Stochastic Systemst. *Automatica*, 30(1):75–93, 1994.

Vapnik, V. N. and Chervonenkis, A. Y. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. *Theory of Probability and its Applications*, XVI(2):264–280, 1971.

Vautard, R. and Ghil, M. Deterministic chaos, stochastic processes, and dimension. *Physica D: Nonlinear Phenomena*, 35(3):395–424, 1989.

Vishwanathan, S. V. N., Smola, A. J., and Vidal, R. Binet-Cauchy Kernels on Dynamical Systems and its Application to the Analysis of Dynamic Scenes. *International Journal of Computer Vision*, 73(1):95–119, 2007.

Wan, L., Zeiler, M., Zhang, S., Lecun, Y., and Fergus, R. Regularization of Neural Networks using DropConnect. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1058–1066, Atlanta, GA, USA, 2012.

Wang, H., Guodong, L., and Guohua, J. Robust Regression Shrinkage and Consistent Variable Selection Through the LAD-Lasso. *Journal of Business & Economic Statistics*, 25(3):347–355, 2007.

Weissman, B., Dell'Osso, L., Abel, L., and Leigh, R. Spasmus nutans. A quantitative prospective study. *Archive of Ophthalmology*, Apr(105):525–8, 1987.

Werbos, P. Applications of advances in nonlinear sensitivity analysis, 1982.

Wiatowski, T. and Bölcskei, H. A Mathematical Theory of Deep Convolutional Neural Networks for Feature Extraction. *IEEE Transactions on Information Theory*, to appear, 2018.

Willemsen, A., Bloemhof, F., and Boom, H. Automatic stance-swing phase detection from accelerometer data for peroneal nerve stimulation. *IEEE Transactions on Biomedical Engineering*, 37(12):1201–1208, 1990.

Williamson, R. and Andrews, B. Gait event detection for FES using accelerometers and supervised machine learning. *IEEE Transactions on Rehabilitation Engineering*, 8(3):312–319, 2000.

Wohlberg, B. Efficient Algorithms for Convolutional Sparse Representations. *IEEE Transactions on Image Processing*, 25(1), 2016.

Xin, B., Wang, Y., Gao, W., and Wipf, D. Maximal Sparsity with Deep Networks? *arXiv preprint*, arXiv:1605(01636), 2016.

Yang, S., Jeong, J., Kim, J., and Yoon, Y. Progressive venous stasis retinopathy and open-angle glaucoma associated with primary pulmonary hypertension. *Ophthalmic surgery Lasers Imaging*, May-Jun(37):230–3, 2006.

Yang, Y., Pilanci, M., and Wainwright, M. J. Randomized sketches for kernels: Fast and optimal non-parametric regression. *arXiv preprint*, arXiv:1501(06195), 2015.

Yellin, F., Haeffele, B. D., and Vidal, R. Blood cell detection and counting in holographic lens-free imaging by convolutional sparse dictionary learning and coding. In *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)*, Melbourne, Australia, 2017.

Ying, H., Silex, C., Schnitzer, A., Leonhardt, S., and Schiek, M. Automatic step detection in the accelerometer signal. In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 80–85, Aachen, Germany, 2007.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 3320–3328, Montreal, Canada, 2014.

You, Y., Lian, X., Liu, J., Yu, H.-F., Dhillon, I. S., Demmel, J., and Hsieh, C.-J. Asynchronous Parallel Greedy Coordinate Descent. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 4682–4690, Barcelona, Spain, 2016.

Yu, H. F., Hsieh, C. J., Si, S., and Dhillon, I. Scalable coordinate descent approaches to parallel matrix factorization for recommender systems. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pp. 765–774, Brussels, Belgium, 2012.

Yuan, M. and Lin, Y. Model selection and estimation in regression with. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Zaremba, W., Sutskever, I., and Vinyals, O. Recurrent neural network regularization. *arXiv preprint*, arXiv:1409(2329), 2014.

Zeiler, M. D., Krishnan, D., Taylor, G. W., and Fergus, R. Deconvolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2528—-2535, San Francisco, CA, USA, 2010. IEEE.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *Proceedings of the International Conference on Learning Representation (ICLR)*, Toulon, France, 2017.

Zhang, Q. and Li, B. Discriminative K-SVD for Dictionary Learning in Face Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, 2010.

Zhu, B., Li, W., Wang, Z., and Xue, X. A novel audio fingerprinting method robust to time scale modification and pitch shifting. In *Proceedings of the International Conference on Multimedia (MM)*, pp. 987, Singapore, Singapore, 2010.

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

Titre : Représentations Convolutives Parcimonieuses – application aux signaux physiologiques et interprétabilité de l'apprentissage profond

Mots Clefs : Codage parcimonieux, Apprentissage de dictionnaire convolutif, Signaux Physiologiques, Optimisation adaptive, Apprentissage Profond.

Résumé : Les représentations convolutives extraient des motifs récurrents qui aident à comprendre la structure locale dans un jeu de signaux. Elles sont adaptées pour l'analyse des signaux physiologiques, qui nécessite des visualisations mettant en avant les informations pertinentes. Ces représentations sont aussi liées aux modèles d'apprentissage profond. Dans ce manuscrit, nous décrivons des avancées algorithmiques et théoriques autour de ces modèles.
Notre contribution principale dans la première partie est un algorithme asynchrone pour accélérer le codage parcimonieux convolutif, nommé DICOD. Notre algorithme présente une accélération super-linéaire. Nous explorons aussi la relation entre l'Analyse du Spectre Singulier et les représentations convolutives, comme une étape d'initialisation de ces dernières.
Dans une seconde partie, nous analysons les liens entre représentations et réseaux de neurones. Le résultat principal est une étude des mécanismes qui rendent possible l'accélération du codage parcimonieux avec des réseaux de neurones. Nous montrons que cela est lié à une factorisation de la matrice de Gram du dictionnaire. D'autres aspects des représentations dans les réseaux neuronaux sont aussi étudiés à travers une étape d'apprentissage supplémentaire, appelée *post-entraînement*, qui améliore les performances du réseau entraîné.
Finalement, nous illustrons l'intérêt de l'utilisation des représentations convolutives pour les signaux physiologiques. L'apprentissage de dictionnaire convolutif est utilisé pour résumer des signaux de marche et le mouvement du regard est soustrait de signaux oculométriques avec l'Analyse du Spectre Singulier.

Title : Convolutional Sparse Representations – application to physiological signals and interpretability for Deep Learning

Keys words : Sparse Coding, Convolutional Dictionary Learning, Physiological Signals, Adaptive Optimization, Deep Learning.

Abstract : Convolutional representations extract recurrent patterns which lead to the discovery of local structures in a set of signals. They are well suited to analyze physiological signals which requires interpretable representations in order to understand the relevant information. Moreover, these representations can be linked to deep learning models, as a way to bring interpretability in their internal representations. In this dissertation, we describe recent advances on both computational and theoretical aspects of these models.
Our main contribution in the first part is an asynchronous algorithm, called DICOD, based on greedy coordinate descent, to solve convolutional sparse coding for long signals. Our algorithm has super-linear acceleration. We also explored the relationship of Singular Spectrum Analysis with convolutional representations, as an initialization step for convolutional dictionary learning.
In a second part, we focus on the link between representations and neural networks. Our main result is a study of the mechanisms which accelerate sparse coding algorithms with neural networks. We show that it is linked to a factorization of the Gram matrix of the dictionary. Other aspects of representations in neural networks are also investigated with an extra training step for deep learning, called post-training, to boost the performances of trained networks by improving their last layer's weights.
Finally, we illustrate the relevance of convolutional representations for physiological signals. Convolutional dictionary learning is used to summarize signals from human walking and Singular Spectrum Analysis is used to remove the gaze movement in young infant's oculometric recordings.