Convolutional Representation
application to physiological signals and interpretability of deep learning.

Thomas Moreau – École Normale Supérieure Paris-Saclay

Plan

Studying physiological signals

Adaptive Iterative Soft Thresholding

Numerical Experiments

Plan

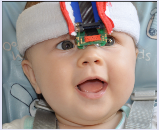
Studying physiological signals

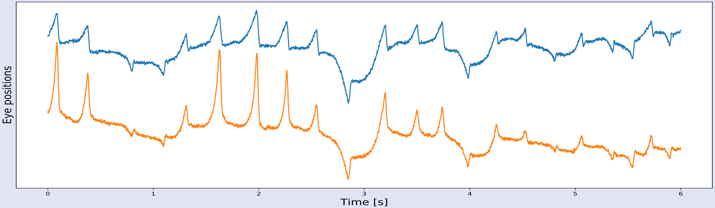
Adaptive Iterative Soft Thresholding

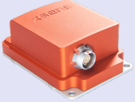
Numerical Experiments

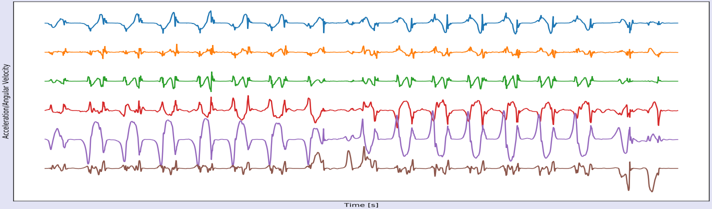
Physiological signals

Oculometric signals









Accelerometers

Physiological signals

Studies with many constraints:

- Non-stationary
- High-dimension
- High-variability
- Interpretability

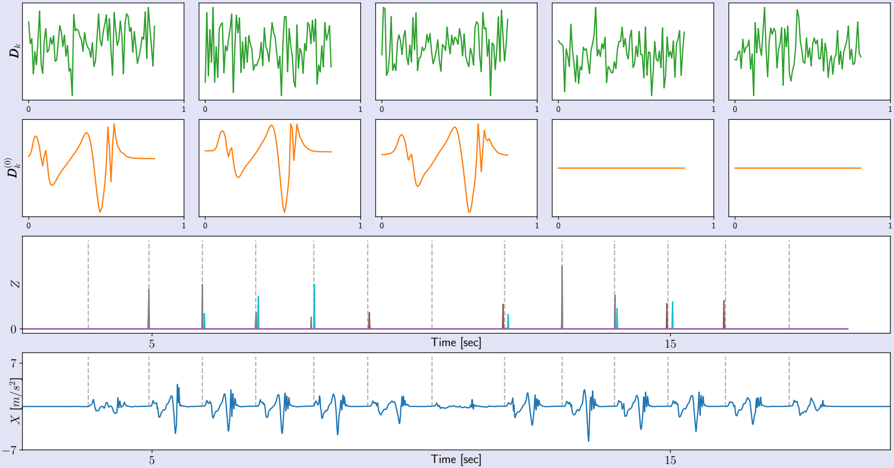
Adaptive methods

Scalable

Dimension reduction

Sparsity

Encoding a walk signal



The activation are concentrated around the steps but there is some dispersion on multiple patterns.

Sparse Representation

Notation:

- x a vector in $\mathbb{R}^P$
- $\mathcal{E}$ is a noise signal in $\mathbb{R}^P$
- $D \in \mathbb{R}^{P \times K}$ is a set of K patterns in $\mathbb{R}^P$
- Z is a coding vector in $\mathbb{R}^K$

Linear model:

$$x = Dz + \mathcal{E}$$

with z sparse. Few of its coefficients are non-zero.

Dictionary learning optimization problem

$$z^*, D^* = \underset{z,D}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} \underbrace{\|x^{[n]} - Dz^{[n]}\|_2^2}_{\text{data fit}} + \underbrace{\lambda \|z^{[n]}\|_1 + \mathbf{1}_\Omega(D)}_{\text{penalizations}}$$

with a constraing set Ω and a regularization parameter $\lambda > 0$.

This problem is non-convex and is generaly solved using an alternate minimization:

1. **Dictionary update:** z fixed, update D
2. **Sparse coding:** D fixed, update Z, independent for each $n \in [\![1, N]\!]$

Sparse coding algorithm

- ISTA [Daubechies et al., 2004]
- Fast ISTA [Beck and Teboulle, 2009]
- ADMM [Gabay and Mercier, 1976]
- Coordinate Descent [Friedman et al., 2007]
- Feature Sign-Search [Lee et al., 2007]

Learned Iterative Soft-Thresholding Algorithm (LISTA)

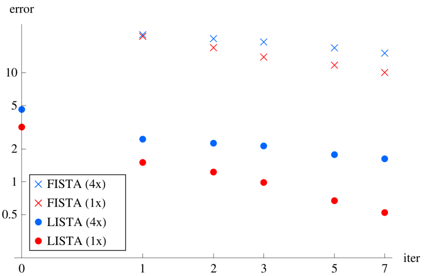
We have to solve N problems with a common structure D.

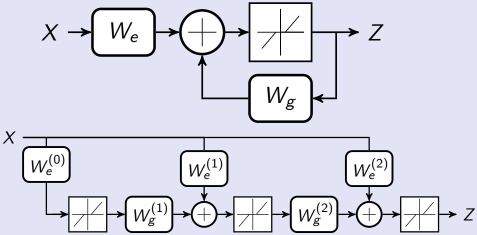
Can we use this structure?

Learned Iterative Soft-Thresholding Algorithm (LISTA)

We have to solve N problems with a common structure D.

Can we use this structure?





LISTA – Adapted from [Gregor and Lecun, 2010]

Why does it work?

Plan

Studying physiological signals

Adaptive Iterative Soft Thresholding

Numerical Experiments

LASSO

The LASSO or sparse coding problem searches for z^* such that

$$z^* = \operatorname*{argmin}_z F(z) := \underbrace{\frac{1}{2}\|x - Dz\|_2^2}_{E(z)} + \lambda\|z\|_1 \ , \qquad (1)$$

where $x \in \mathbb{R}^P$, $D \in \mathbb{R}^{P \times K}$ and $z \in \mathbb{R}^K$.

We denote $B = D^\mathsf{T} D$ is the Gram matrix of D.

The quadratic-form Q_S

Define $Q_S(u,v) = \frac{1}{2}(u-v)^\mathsf{T} S(u-v) + \lambda \|u\|_1$.

If S is diagonal, the following problem can efficiently be solved:

$$\operatorname*{argmin}_u Q_S(u,v)$$

The problem is separable on each coordinate:

$$\operatorname*{argmin}_{u_i} \frac{s_i}{2}(u_i - v_i)^2 + \lambda \|u_i\|$$

$\Rightarrow$ Scaled soft thresholding

$$u_i^* = \frac{\operatorname{sign}(v_i)}{s_i} \max(0, |v_i| - \lambda)$$

Toward an adaptive procedure

Given an estimate $z^{(q)}$ of z^* at iteration q, we can write:

$$\begin{aligned} F(z) &= E(z) + \lambda\|z\|_1 \\ &= E(z^{(q)}) + \left\langle \nabla E(z^{(q)}), z - z^{(q)} \right\rangle + Q_B(\quad z, \quad z^{(q)}) \ , \end{aligned}$$

Toward an adaptive procedure

Given an estimate $z^{(q)}$ of z^* at iteration q, we can write:

$$\begin{aligned} F(z) &= E(z) + \lambda \|z\|_1 \\ &= E(z^{(q)}) + \left\langle \nabla E(z^{(q)}), z - z^{(q)} \right\rangle + Q_B(\quad z, \quad z^{(q)}) \ , \end{aligned}$$

ISTA: Replace B by diagonal matrix $S = L\boldsymbol{I}_K$

$$F_q(z) \quad = \quad E(z^{(q)}) + \left\langle \nabla E(z^{(q)}), z - z^{(q)} \right\rangle + Q_{\mathbf{S_q}}(\quad z, \quad z^{(q)}) \ ,$$

$$\min_z F_q(z) \quad \Leftrightarrow \quad \min_z Q_{S_q}\left(\quad z, \quad z^{(q)} - \ S_q^{-1} \ \nabla E(z^{(q)})\right)$$

Toward an adaptive procedure

Given an estimate $z^{(q)}$ of z^* at iteration q, we can write:

$$\begin{aligned} F(z) &= E(z) + \lambda\|z\|_1 \\ &= E(z^{(q)}) + \left\langle \nabla E(z^{(q)}), z - z^{(q)} \right\rangle + Q_B(\quad z, \quad z^{(q)}) \ , \end{aligned}$$

ISTA: Replace B by diagonal matrix $S = L\mathbf{I}_K$

FacNet: Replace B by $A^\mathsf{T} SA$ (S diagonal, A unitary)

$$\begin{aligned} \widetilde{F}_q(z) &= E(z^{(q)}) + \left\langle \nabla E(z^{(q)}), z - z^{(q)} \right\rangle + Q_{S_q}(\mathbf{A_q} z, \mathbf{A_q} z^{(q)}) \ , \\ \min_z \widetilde{F}_q(z) &\Leftrightarrow \min_z Q_{S_q}\left(A_q z, A_q z^{(q)} - S_q^{-1} A_q \nabla E(z^{(q)}) \right) \end{aligned}$$

Toward an adaptive procedure

Given an estimate $z^{(q)}$ of z^* at iteration q, we can write:

$$
\begin{aligned}
F(z) &= E(z) + \lambda \|z\|_1 \\
&= E(z^{(q)}) + \left\langle \nabla E(z^{(q)}), z - z^{(q)} \right\rangle + Q_B(\quad z, \quad z^{(q)}) \ ,
\end{aligned}
$$

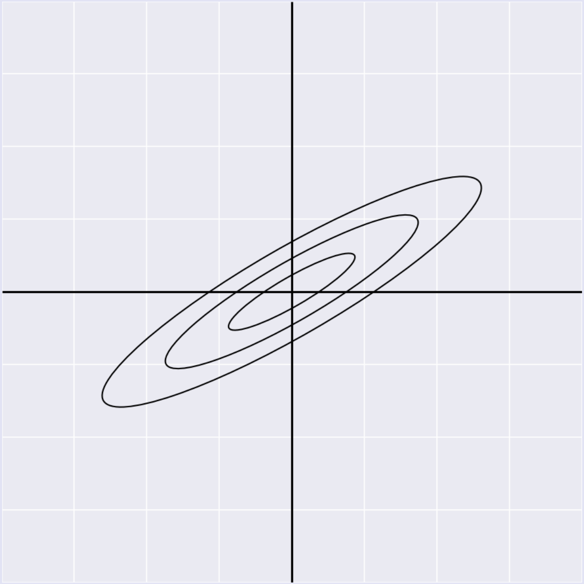
ISTA: Replace B by diagonal matrix $S = L\boldsymbol{I}_K$

FacNet: Replace B by $A^\mathsf{T} S A$ (S diagonal, A unitary)

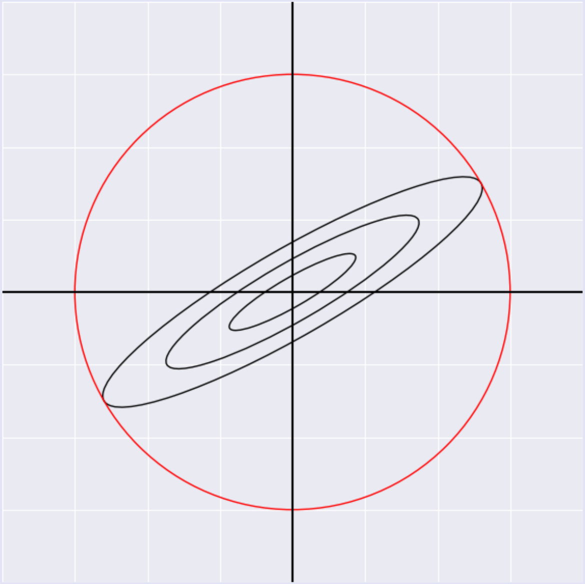
$$
\begin{aligned}
\widetilde{F}_q(z) &= E(z^{(q)}) + \left\langle \nabla E(z^{(q)}), z - z^{(q)} \right\rangle + Q_{S_q}(\mathbf{A_q} z, \mathbf{A_q} z^{(q)}) \ , \\
\min_z \widetilde{F}_q(z) &\Leftrightarrow \min_z Q_{S_q}\left(A_q z, A_q z^{(q)} - S_q^{-1} A_q \nabla E(z^{(q)})\right)
\end{aligned}
$$

Can we choose A_q, S_q to accelerate the optimization compared to ISTA?

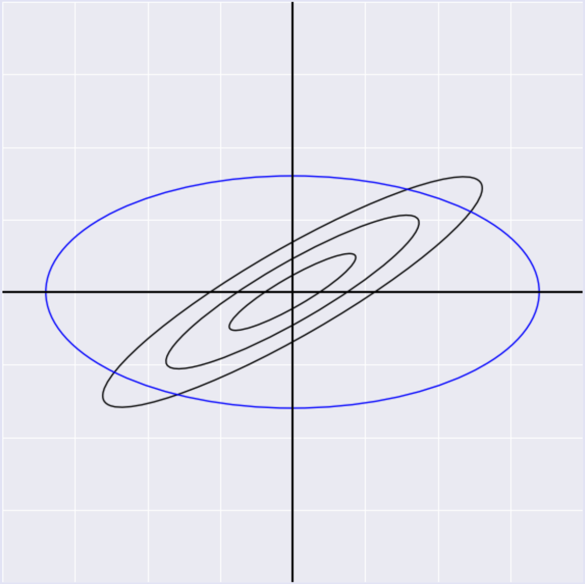
Quadratic form



Quadratic form



Quadratic form



Iterative procedures

ISTA:

$$
\begin{aligned}
z^{(q+1)} &= \operatorname*{argmin}_z F_q(z) \\
&= \operatorname*{prox}_S\left(z^{(q)} - \frac{1}{\|B\|_2}\nabla E(z^{(q)})\right) ,
\end{aligned}
$$

FacNet:

$$
\begin{aligned}
z^{(q+1)} &= \operatorname*{argmin}_z \widetilde{F}_q(z) \\
&= A_q^{\mathsf{T}} \operatorname*{prox}_S\left(A_q z^{(q)} - S_q^{-1} A_q \nabla E(z^{(q)})\right) ,
\end{aligned}
$$

Toward an adaptive procedure

Similar iterative procedure with steps adapted to the problem topology.

$$\widetilde{F_q}(z) = F(z) + (z - z^{(q)})^{\mathsf{T}} R (z - z^{(q)}) + \delta_A(z)$$

Tradeoff between:

- Rotation to align the norm $\|\cdot\|_B$ and the norm $\|\cdot\|_1$, Computation

$$R = A^{\mathsf{T}} S A - B$$

- Deformation of the ℓ_1-norm with the rotation A . Accuracy

$$\delta_A(z) = \lambda \left(\|Az\|_1 - \|z\|_1 \right)$$

One step improvement

Proposition

Suppose that $R_q = A_q^\mathsf{T} S_q A_q - B \succ 0$ is positive definite, and define

$$z^{(q+1)} = \arg\min_z \widetilde{F_q}(z) \ ,$$

Then

$$F(z^{(q+1)}) - F(z^*) \leq \frac{1}{2}(z^{(q)} - z^*)^\mathsf{T} R_q (z^{(q)} - z^*) + \delta_{A_q}(z^*) - \delta_{A_q}(z^{(q+1)}) \ .$$

We are interested in factorization (A_q, S_q) for which $\|R_q\|_2$ and δ_{A_q} are small.

Adaptive Iterative Soft thresholding - Convergence rate [Moreau and Bruna, 2017]

Theorem

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

$$F(z^{(q)}) - F(z^*) \leq \frac{(z^* - z^{(0)})^\mathsf{T} R_0 (z^* - z^{(0)}) + 2L_{A_0}(z^{(1)}) \|(z^* - z^{(1)})\|_2}{2q} + \frac{\alpha_q - \beta_q}{2q} ,$$

$$\alpha_q = \sum_{i=1}^{q-1} \left(2L_{A_i}(z^{(i+1)}) \|(z^* - z^{(i+1)})\| + (z^* - z^{(i)})^\mathsf{T} (R_{i-1} - R_i)(z^* - z^{(i)}) \right) ,$$

$$\beta_q = \sum_{i=0}^{q-1} (i+1) \left((z^{(i+1)} - z^{(i)})^\mathsf{T} R_i (z^{(i+1)} - z^{(i)}) + 2\delta_{A_i}(z^{(i+1)}) - 2\delta_{A_i}(z^{(i)}) \right) ,$$

where $L_A(z)$ denote the local Lipschitz constant of δ_A at z.

Interpretation

- For $A_q = \boldsymbol{I}_K$ and $S_q = \|B\|_2 \boldsymbol{I}_K$, the procedure is equivalent to ISTA, with the same rate of convergence.

- If $\|R_0\|_2 + 2\frac{L_{A_0}(z_1)}{\|z^* - z_0\|_2} \leq \frac{\|B\|_2}{2}$ and $A_q = \boldsymbol{I}_K$ and $S_q = \|B\|_2 \boldsymbol{I}_K$ for $q > 0$, then the procedure get a head start compare to ISTA

- **Phase transition :**
  The upper bound is improved when $\|R_q\|_2 + 2\frac{L_{A_q}(z^{(q+1)})}{\|z^* - z^{(q)}\|_2} \leq \frac{\|B\|_2}{2}$,
  it is thus harder to gain as $\|z^{(q)} - z^*\|_2 \to 0$

Generic Dictionaries

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 $\mathcal{S}^{p-1}$.

Theorem (Acceleration conditions)

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 – when

$$\lambda \mathbb{E}_z\left[\|z^{(q+1)}\|_1 + \|z^*\|_1\right] \le \sqrt{\frac{K(K-1)}{p}}\underbrace{\mathbb{E}_z\left[\|z^{(q)} - z^*\|_2^2\right]}_{\substack{\text{expected resolution}\\ \text{at iteration } q}}$$

Generic Dictionaries

Corollary (Acceleration conditions)

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.

FacNet can improve the performances compared to ISTA when this is verified.

Related works - explaining LISTA

- [Giryes et al., 2016]: Explanation based on the input distribution.
  Propose the inexact projected gradient descent and conjecture that LISTA accelerate the LASSO resolution by learning the sparsity pattern of the input distribution.

- [Xin et al., 2016]: Study assymptotic properties of z^* estimators.
  Study the Hard-thresholding Algorithm and its capacity to recover the support of a sparse vector.
  The paper relax the RIP conditions for the dictionary.

Plan

Studying physiological signals

Adaptive Iterative Soft Thresholding

Numerical Experiments

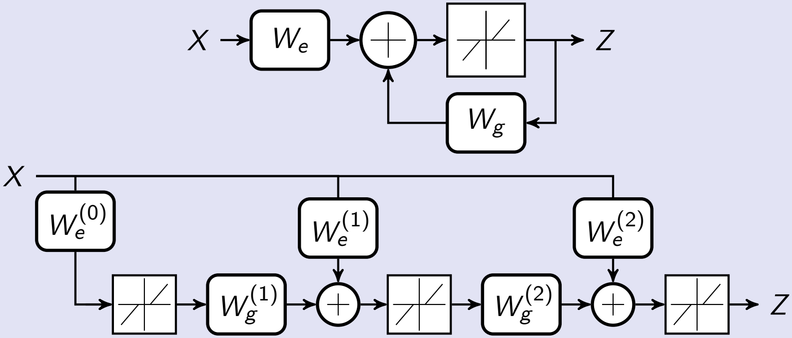


Figure: Network architecture for ISTA/LISTA. LISTA is the unfolded version of the RNN of ISTA, trainable with back-propagation.

If $W_e = \frac{D^\top}{L}$ and $W_g = I - \frac{B}{L}$, this network is exactly 2 iterations of ISTA.

FacNet

Specialization of LISTA

$$z^{(q+1)} = A^{\mathsf{T}} \operatorname*{prox}_{S}(Az^{(q)} - S^{-1}AB(z^{(q)} - y)) \ ,$$

with A unitary and S diagonal.
Same architecture with more constraints on the parameter space:

$$\begin{cases} W_e & = S^{-1}AD^{\mathsf{T}} \\ W_g & = A^{\mathsf{T}} - S^{-1}ABA^{\mathsf{T}} \end{cases}$$

$\Rightarrow$ LISTA can be at least as good as this model.

Artificial simulation

Generating Model:

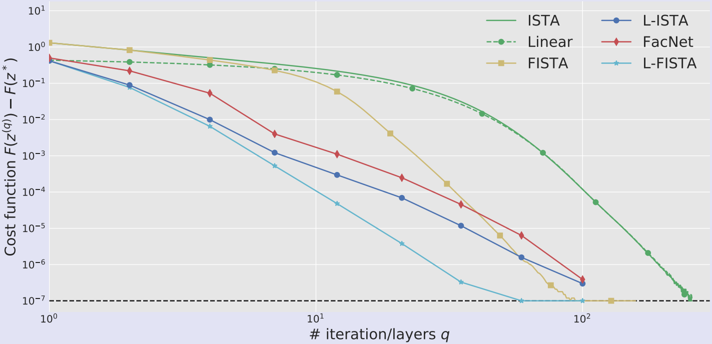
- $D = \left(\frac{d_1}{\|d_1\|_2}, \dots \frac{d_K}{\|d_K\|_2} \right)$ with $d_k \sim \mathcal{N}(0, \boldsymbol{I}_P)$ for all $k \in [\![1, K]\!]$,
- $z = (z_1, \dots z_K)$ are constructed following a bernouilli gaussian:

$$z_k = b_k a_k, \qquad b_k \sim \mathcal{B}(\rho) \text{ and } a \sim \mathcal{N}(0, \sigma \boldsymbol{I}_K)$$

with: $K = 100$, $P = 64$, for the dimension, $\sigma = 10$ and $\lambda = 0.01$

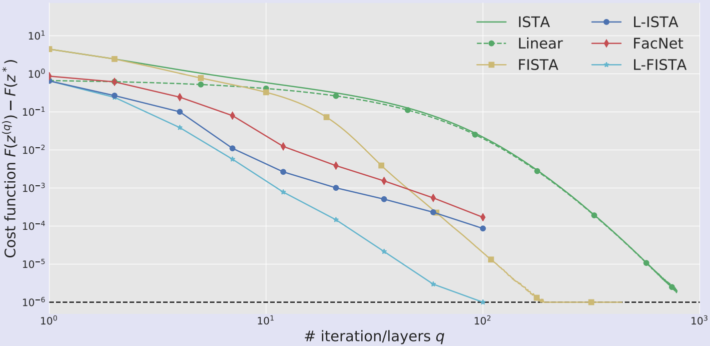
$\Rightarrow$ The sparsity patterns are uniformely distributed.

Artificial simulation



Evolution of the cost function $F(z^{(q)}) - F(z^*)$ with the number of layers/iterations q with a sparse model $\rho = {}^1/_{20}$.

Artificial simulation

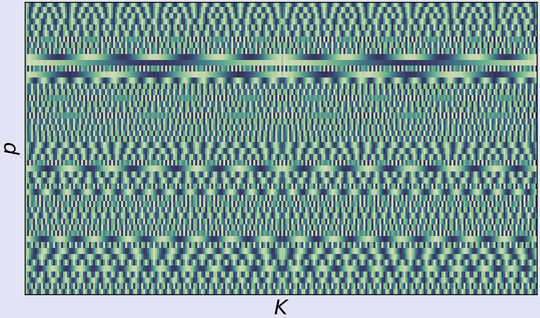


Evolution of the cost function $F(z^{(q)}) - F(z^*)$ with the number of layers/iterations q with a denser model $\rho = {}^1/_4$.

Adversarial dictionary:
$D = \left[d_1 \dots d_K\right] \in \mathbb{R}^{K \times p}$,
with

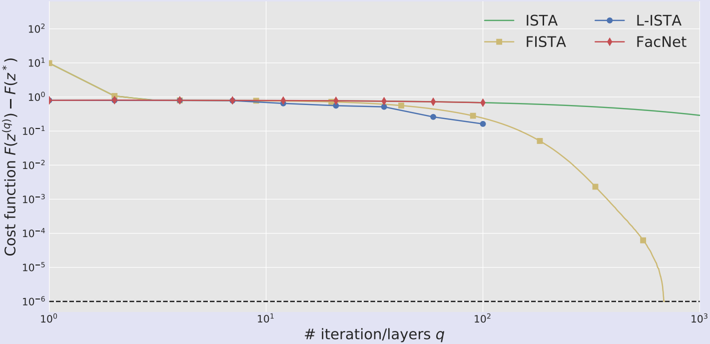
$$d_j = e^{-i\frac{2\pi j \zeta_q}{K}}$$

for a random subset of frequencies $\left\{\zeta_i\right\}_{i \leq m}$



$\Rightarrow$ Eigenvectors of D are far from canonical basis.

Adversarial dictionary



Evolution of the cost function $F(z^{(q)}) - F(z^*)$ with the number of layers/iterations k with n adversarial dictionary.

Contribution

Contributions

- Non asymptotic acceleration of ISTA is possible based on the structure of $\boldsymbol{D}$
- Sufficient analysis to explain LISTA acceleration,
- The dictionary structure seems necessary.

Future work:

- Improve the factorization formulation for direct optimization,
- Second order analysis for generic dictionary,
- Link to Sparse PCA.

Thanks!

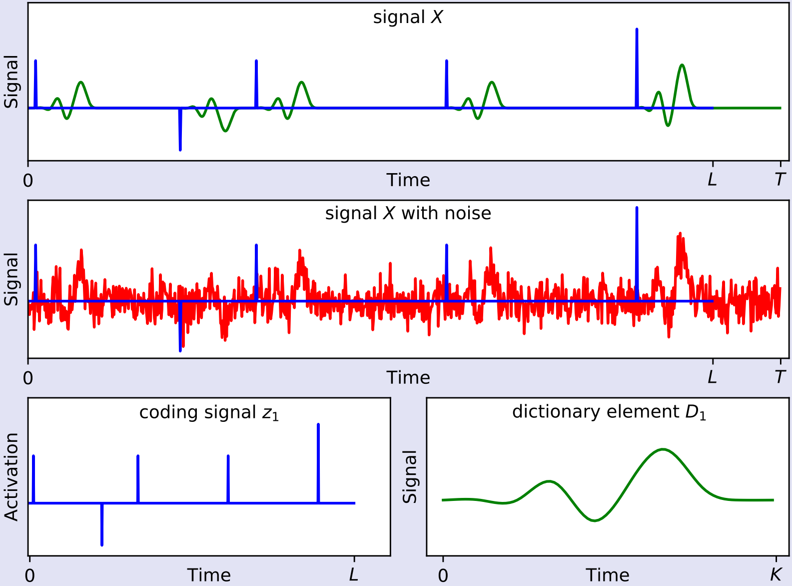
Code: tomMoral

Papers: tommoral.github.io

Part II:
Accelerating Convolutional Sparse Coding

Plan

Motivation


signal X
Signal
0
Time
L
T
signal X with noise
Signal
0
Time
L
T
coding signal z_1
Activation
0
Time
L
dictionary element D_1
Signal
0
Time
K

Convolutional Sparse Coding

For a signal X, find the coding signal Z given a set of K patterns $\boldsymbol{D}$.

Optimization problem

Solve a ℓ_1-regularized minimization problem

$$Z^* = \arg\min_Z E(Z) = \frac{1}{2}\|X - \sum_{k=1}^{K} Z_k * \boldsymbol{D}_k\|_2^2 + \lambda\|Z\|_1, \qquad (2)$$

Existing algorithms do not scale well with the size of the signal X.

- Feature Sign Search (FSS) [Grosse et al., 2007]
- Fast Iterative Soft Thresholding (FISTA) [Chalasani et al., 2013]
- Fast Convolutional Sparse Coding (FCSC) [Bristow et al., 2013]
- Coordinate Descent (CD) [Kavukcuoglu et al., 2010]

Coordinate Descent (CD)

Update the problem for one coordinate at each iteration.
The problem in one coordinate is:

$$e_{k,t}(u) = \frac{\|\boldsymbol{D}_k\|_2^2}{2}\left(u - \beta_k[t]\right)^2 + \lambda|y|$$

with $\beta_k[t] = \left(\left(X - \Phi_{k,t}(Z) * \boldsymbol{D}^T\right) * \widetilde{\boldsymbol{D}}_k\right)[t]$.

Three algorithms based on this idea:

- Cyclic updates [Friedman et al., 2007]
- Random updates [Nesterov, 2012]
- Greedy updates [Osher and Li, 2009]

Recent work shows it is more efficient to use greedy updates.
[Nutini et al., 2015]

Convolutional Coordinate Descent

For convolutional CD, we can use greedy updates:

$$Z_k' = \frac{1}{\|\boldsymbol{D}_k\|_2^2}\mathrm{Sh}(\beta_k, \lambda),$$

with $\mathrm{Sh}(y, \lambda) = \mathrm{sign}(y)(|y| - \lambda)_+$.

This can be done efficiently for this problem by maintaining β, with $\mathcal{O}(KS)$ operations. [Kavukcuoglu et al., 2010]

$$\beta_k^{(q+1)}[t] = \beta_k^{(q)}[t] - \mathcal{S}_{k,k_0}[t - t_0](Z_{k_0}[t_0] - Z'_{k_0}[t_0]),$$

with $\mathcal{S}_{k,k_0}[t]=\sum_{\tau=0}^{S-1}\boldsymbol{D}_k[t+\tau]\boldsymbol{D}_{k_0}^T[\tau]$.

Improving Convolutional Coordinate Descent(1/2)

This is not so efficient to only change one coordinate as updates only affect a small range of coefficients.

We could update M coefficients that are in disjoint neighborhoods in **parallel**.

Issue: Choose disjoint coordinates
Split the signal in M continuous chunks and perform updates:

- Use a lock to avoid updates that are too close,
- Use a parameter server to reject multiple updates.

[Scherrer et al., 2012, Bradley et al., 2011]
[Yu et al., 2012, Low et al., 2012]

Is it necessary?

Improving Convolutional Coordinate Descent (2/2)

Consider the cost function $E(Z) = \frac{1}{2}\|X - \sum_{k=1}^{K} Z_k * \boldsymbol{D}_k\|_2^2 + \lambda\|Z\|_1$

We denote $\Delta E_0 = E(Z^{(q+1)}) - E(Z^{(q)})$ the update performed at step q for coefficient (k_0, t_0).

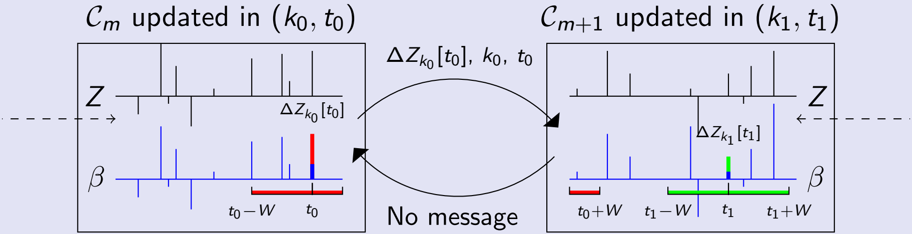
If we update simultaneously (k_0, t_0) and (k_1, t_1) coefficients, it can be shown that:

$$\Delta E_{0,1} = \underbrace{\Delta E_0 + \Delta E_1}_{\text{iterative steps}} - \underbrace{\mathcal{S}_{k_0,k_1}[t_1 - t_0]\Delta Z_0 \Delta Z_1}_{\text{interference}},$$

If interference are not too high, the updates can be asynchronous.

Distributed Convolutional Coordinate Descent (DICOD)

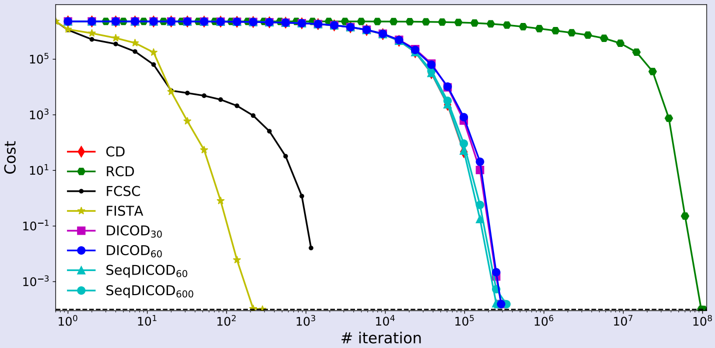
Each core is responsible for the updates of a chunk of coefficients.



Retrieve the notification when possible to update β.

Generated problems with $\boldsymbol{D}$ gaussian and Z Bernouilli-Gaussian

$$X = \sum_{k=1}^{K} Z_k * \boldsymbol{D}_k + \epsilon$$

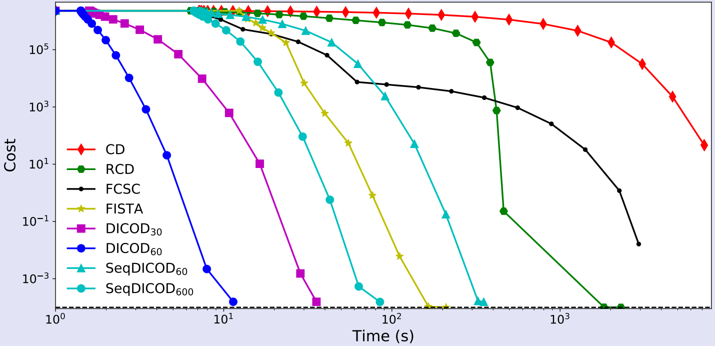


Cost as a function of the iterations

Numerical convergence

Generated problems with $\boldsymbol{D}$ gaussian and Z Bernouilli-Gaussian

$$X = \sum_{k=1}^{K} Z_k * \boldsymbol{D}_k + \epsilon$$



Cost as a function of the time

Complexity Analysis

Computational cost of one update for greedy CD is linear in $\mathcal{O}(T)$:

- Compute potential updates $Z'_k[t]$,
- Find $(k_0, t_0) = \arg\min_{k,t} |Z'_k[t] - Z_k[t]|$.

Computational cost for one update of DICOD is linear in $\mathcal{O}(\frac{T}{M})$:

- Same steps but with a signal of size $\frac{T}{M}$.

Speedup

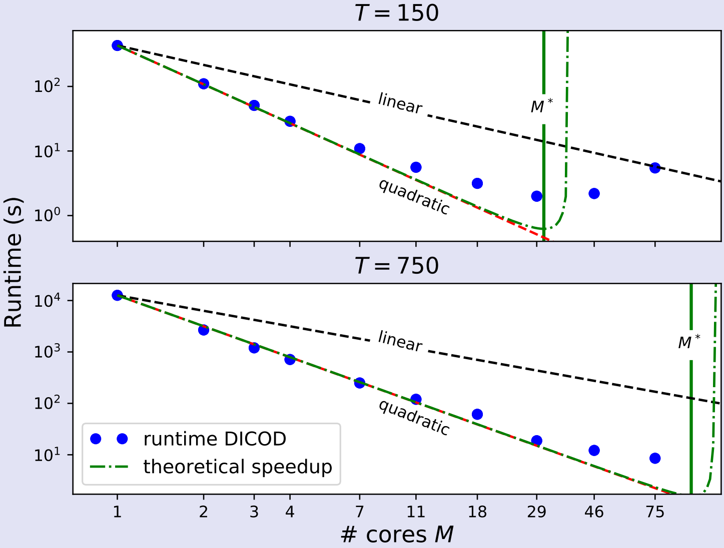
With an analysis of the interference probability, the convergence rate of DICOD with M cores can be bounded by:

$$
\begin{aligned}
\mathbb{E}[S_{dicod}] &\geq M^2\left(1 - 2\alpha^2 M^2 \left(1 + 2\alpha^2 M^2\right)^{\frac{M}{2}-1}\right) ,\\
&\underset{\alpha\to 0}{\gtrsim} M^2(1 - 2\alpha^2 M^2 + \mathcal{O}(\alpha^4 M^4)) .
\end{aligned}
\tag{3}
$$

with $\alpha^2 = \left(\frac{SM}{T}\right)^2$ the probability of interference.

- For α close to 0, the speedup is quadratic.
- There is a sharp transition as α grows that degrade the performance of the algorithm.

Numerical Speedup



Runtime as a function of the number of cores

Contributions

Contributions

- Distributed algorithm efficient to solve the CSC problem
- Guaranteed convergence to the optimal solution
- Super-linear speedup

Future work

- Extend this work to 2D case
- Handle local penalties

What next?

- Find a good way to solve the dictionary learning problem,
- Change the penalization? (group sparse),
- Use the learned dictionary to extract meaningful features.

Publications

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

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

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

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

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

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

Barrois, R., Oudre, L., Moreau, T., Truong, C., Vayatis, N., Buffat, S., Yelnik, A., de Waele, C., Gregory, T., Laporte, S., and Others (2015). 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

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. (2016). An automated recording method in clinical consultation to rate the limp in lower limb osteoarthritis. *PLoS ONE*, 11(10):e0164975

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

Plan

Convolutional Dictionary Learning

Annex FacNet

Annex DICOD

Singular Spectrum Analysis (SSA)

Post-training for Deep Learning

Related works

- [Giryes et al., 2016]: Propose the inexact projected gradient descent and conjecture that LISTA accelerate the LASSO resolution by learning the sparsity pattern of the input distribution.

- [Xin et al., 2016]: Study the Hard-thresholding Algorithm and its capacity to recover the support of a sparse vector.
  The paper relax the RIP conditions for the dictionary.

Learned FISTA

The same ideas can also be applied to FISTA to obtain similar procedures:

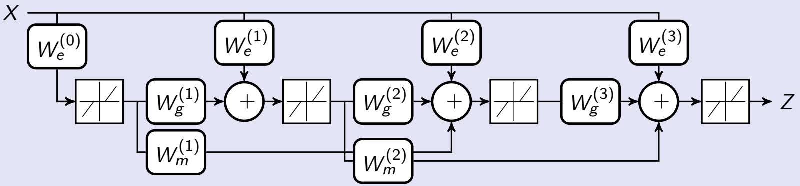


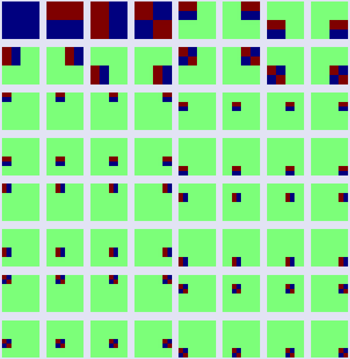
Figure: Network architecture for L-FISTA.

PASCAL 08

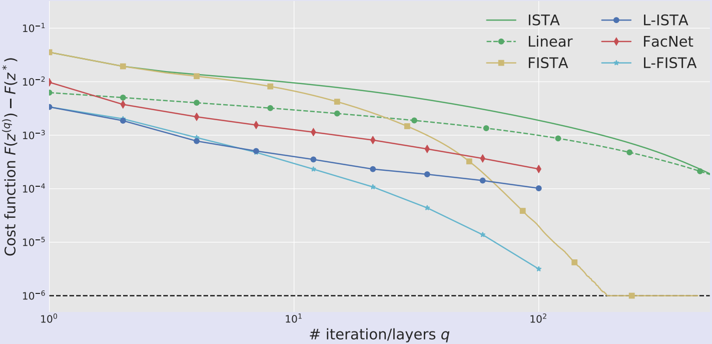
Sparse coding for the PASCAL 08 datasets over the Haar wavelets family.

The sparse coding is performed for patches of size 8×8.

Train over 500 images and test over 100 images.



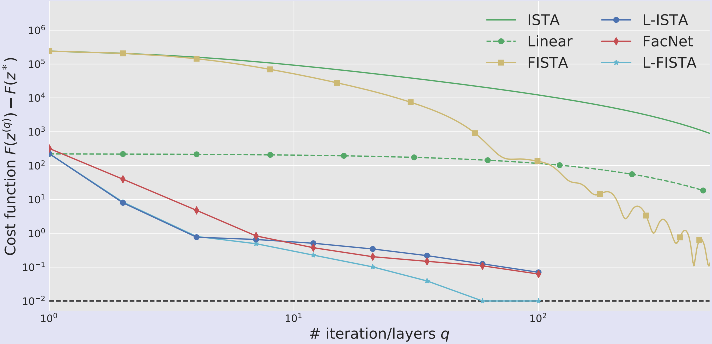
PASCAL 08



Evolution of the cost function $F(z^{(q)}) - F(z^*)$ with the number of layers or the number of iteration q for Pascal VOC 2008.

MNIST

Dictionary D with $K = 100$ atoms learned on 10 000 MNIST samples (17x17) with dictionary learning. LISTA trained with MNIST training set and tested on MNIST test set.



Evolution of the cost function $F(z^{(q)}) - F(z^*)$ with the number of layers or the number of iteration q for MNIST.

Plan

Finishing the process in a linear grid?

Non trivial point: **How to decide that the algorithm has converged?**

- Neighbors paused is not enough!
- Define a master 0 and send probes.
  Wait for M probes return.
- Uses the notion of message queue and network flow.
  Maybe we can have better way?

Plan

Idea

- Choose a window size K and extract sub series, → K-trajectory matrix $\boldsymbol{X}^{(K)}$
- Reconstruct a low rank estimate of all the K-length sub series, → Singular Value decomposition $X^{(K)} = \sum_{k=1}^{K} \lambda_k U_k V_k^T$
- Decomposition of the series as a sum of "low rank" components. → Average along anti-diagonals

$\Rightarrow$ Extract components linked to trend and oscillations

Singular Spectrum Analysis

We show in the thesis that this solves the following problem

Optimization problem

Solve a convolutional list square

$$Z^*, \boldsymbol{D}^* = \arg\min_{Z,\boldsymbol{D}} \frac{1}{2} \left\| X - \sum_{k=1}^{K} z_k * D_k \right\|_2^2, \qquad (4)$$

with constraints $\langle D_i, D_j \rangle = \delta_{i,j}$

- $\boldsymbol{D}$ is the dictionary with K patterns in $\mathbb{R}$ of length W
- Z is an activation signal, or coding signal in $\mathbb{R}^K$ of length $L = T - W + 1$

Issues

Same pattern present in different low rank components

Representation is "dense", no localization

Different representation for each signal

Plan

Paper with J. Audiffren: arxiv:1611.04499

Use the idea to split the representation learning and the task resolution:

- *Post-training* step: only train the last layer,
- Easy problem: this problem is often convex,
- Link with kernel: close form solution for optimal last layer.
- Experiments: consistent performance boost with multiple architecture.