Understanding physiological signals via sparse representations

Thomas Moreau ENS Cachan - CMLA

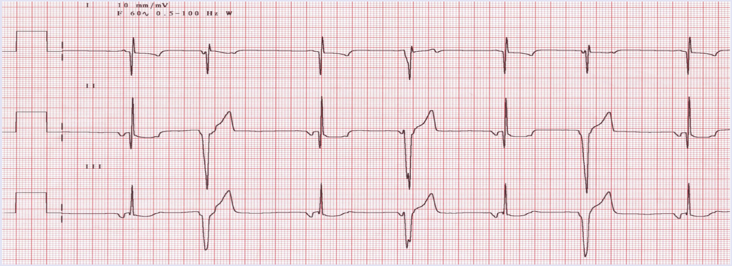
Work in collaboration with L. Oudre, N. Vayatis

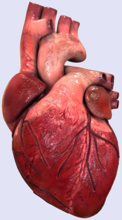
Plan

1. Physiological signals
2. Time Series Representations
3. Singular Spectrum Analysis (SSA)
4. Convolutional Dictionary Learning
5. Application to physiological signals

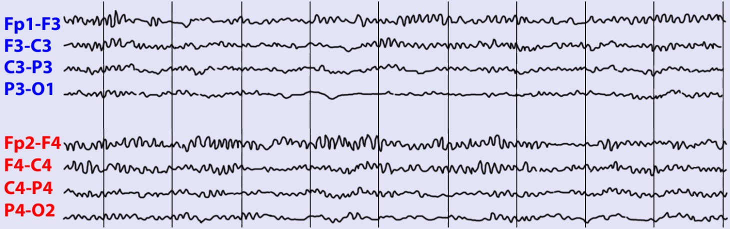
Plan

ECG

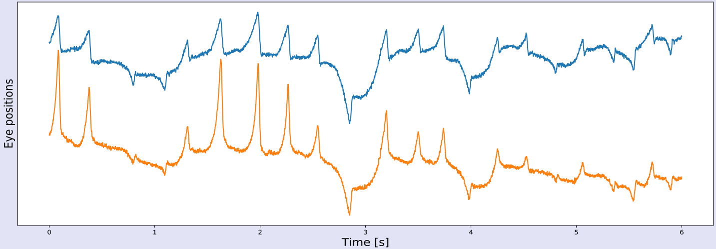


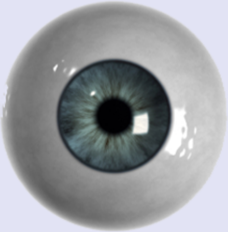


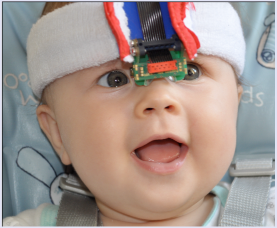
EEG



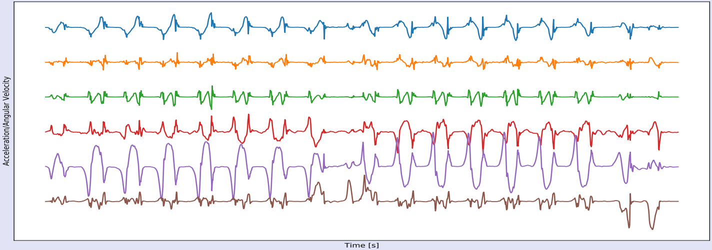
Oculometric signals

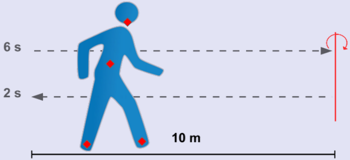




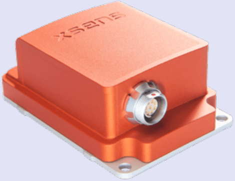


Accelerometers





4×



- Failure of the vectorial distances
  - Alignment issues, different lengths (can be solved with DTW)
  - "Curse of dimensionality"
- Different approaches which can be classified in 2 categories :
  - Model based methods (feature extraction + vectorial method, ...)
  - Data driven methods (End-to-end model, Neural networks, ...)

- Failure of the vectorial distances
  - Alignment issues, different lengths (can be solved with DTW)
  - "Curse of dimensionality"
- Different approaches which can be classified in 2 categories :
  - Model based methods (feature extraction + vectorial method, ...)
  - Data driven methods (End-to-end model, Neural networks, ...)

Plan

Finding a good representation is challenging :

- Samples with various lengths and scales,

  Invariant

- Heterogeneous sampling rates across channels,

  Nonparametric

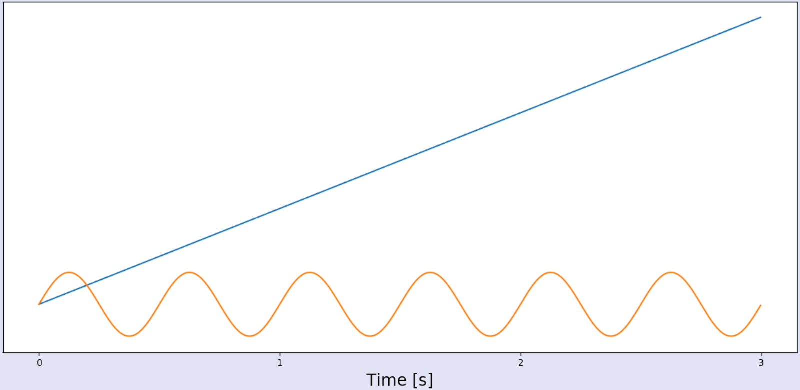
- Samples have high dimension,

  Scalable

- Non stationary signals.

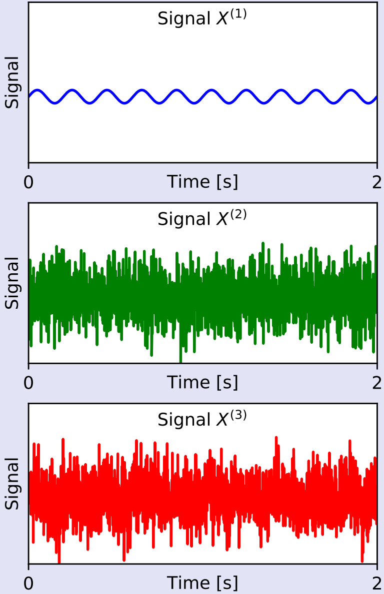
  Robust

Temporal representation



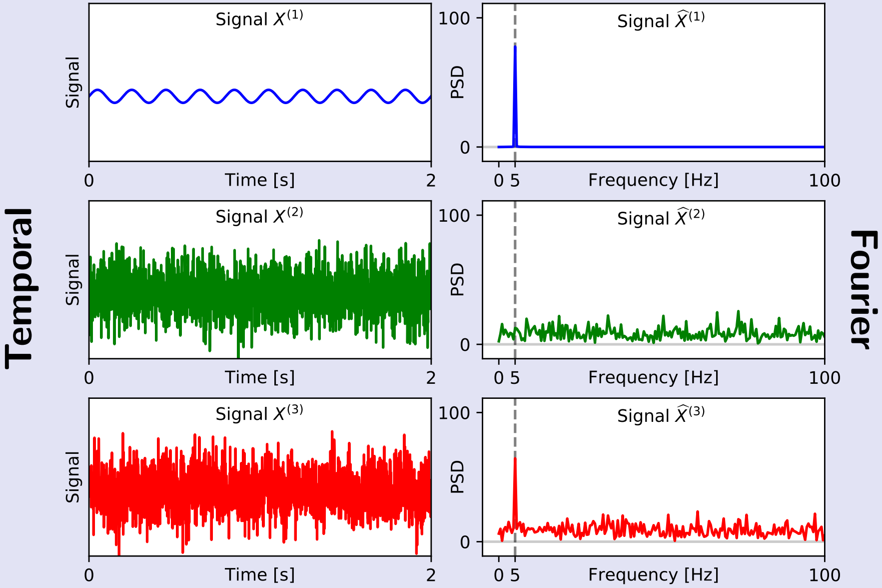
- Most common representation
- Can be efficient (cardiologist)
- Permits to detect patterns (linearity, periodicity, ...)

Temporal representation

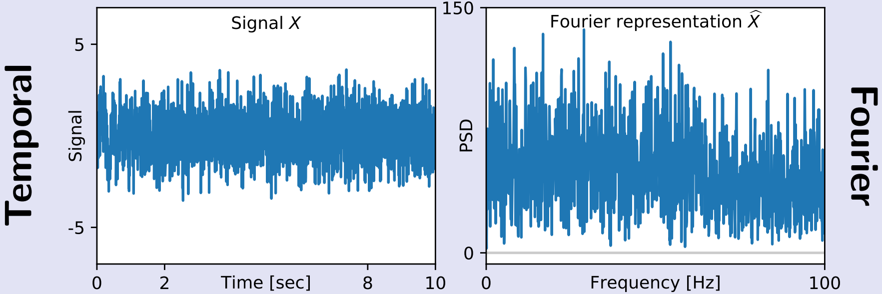


- Not robust to noise
- Limited interpretation

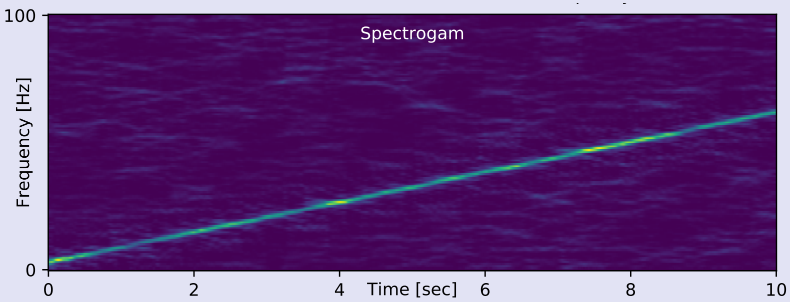
Fourier representation



Not so efficient for non stationary signals.



Need some time-frequency insights.



Main idea :

apply global representation to windowed signals

Main drawback :

We need to know the property we are looking for.

Data-driven representation

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

⇒ Extract components linked to trend and oscillations

In practice, this solves the following problem

Optimization problem

Solve a convolutional list square

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

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

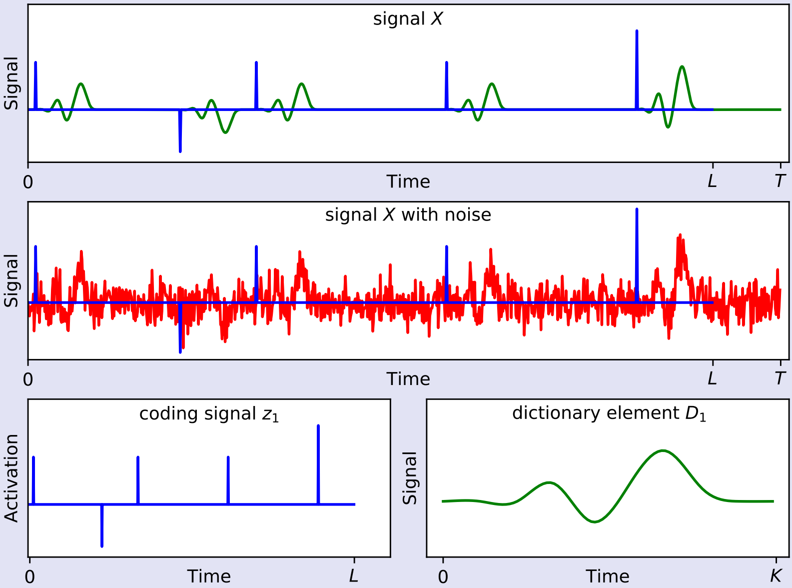
Same pattern present in different low rank components

Representation is "dense", no localization

Different representation for each signal

Plan

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 dictionary learning

- Shift invariant patterns
- Separation between the localization and the shapes of the patterns

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

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

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., 2013]

Coordinate Descent (CD)

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

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

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

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., 2013]

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

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 ?

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).

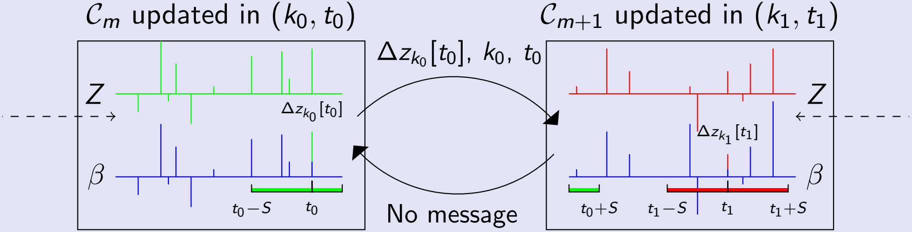
If we update simultaneously (k_0, t_0) and (k_1, t_1) coefficients, it can be shown that :

$$\Delta E_{0,1} = \underbrace{\Delta E_0 + \Delta E_1}_{\text{iterative steps}} - \underbrace{\mathcal{S}_{k_0,k_1}[t_1 - t_0]\Delta z_0 \Delta z_1}_{\text{interference}},$$

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.

Convergence Analysis

We denote :

$$C_{k_0 k_1}[t_0 - t_1] = \frac{\mathcal{S}_{k_0,k_1}[t_0 - t_1]}{\|\boldsymbol{D}_{k_0}\|_2 \|\boldsymbol{D}_{k_1}\|_2}.$$

Theorem

We consider the following assumptions :

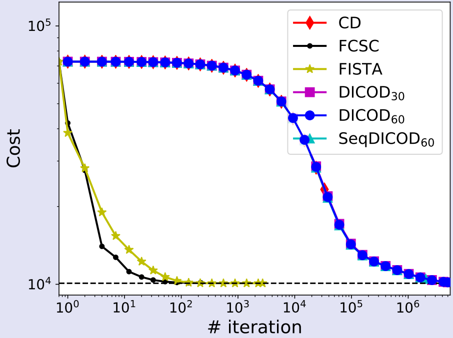
H1 : If for all k_0, k_1, t_0, t_1 such that $t_0 - t_1 \neq 0$, $|C_{k_0 k_1}[t_0 - t_1]| < 1$.

H2 : If there exist $A \in \mathbb{N}^*$ such that for all $m \in \{1, \dots M\}$ and $q \in \mathbb{N}$, $\mathcal{C}_m$ is updated at least once between iteration q and $q + A$ if the solution is not optimal for all coefficients assigned to $\mathcal{C}_m$.

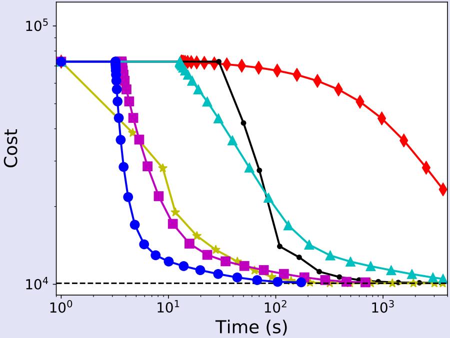
Under these assumptions, the DICOD algorithm converges to the optimal solution z^* of 2.

Artificial problems with $\boldsymbol{D}$ sinusoidal patterns of size $W = 200$ in $\mathbb{R}^7$, Z gaussian bernouilli of length $600W$ and ϵ a white noise such that :

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



Cost as a function of the iterations



Cost as a function of the runtime

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

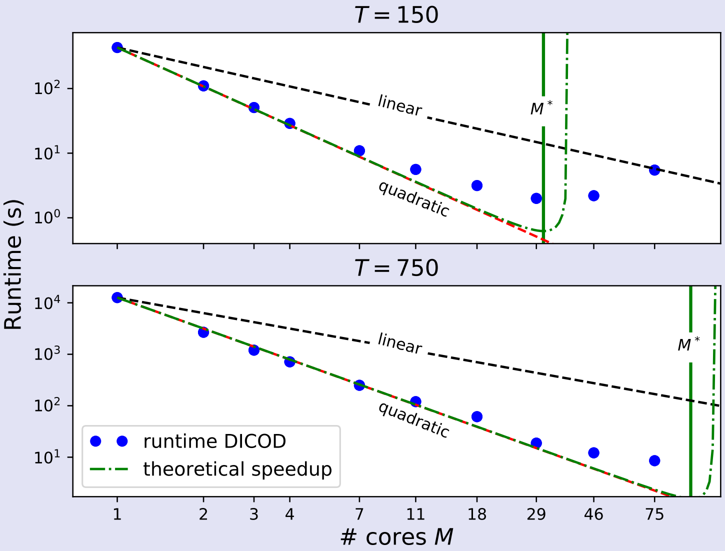
With an analysis of the interference probability, the convergence rate of DICOD with M cores can be bounded by :

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

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

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

Encoding a walk signal

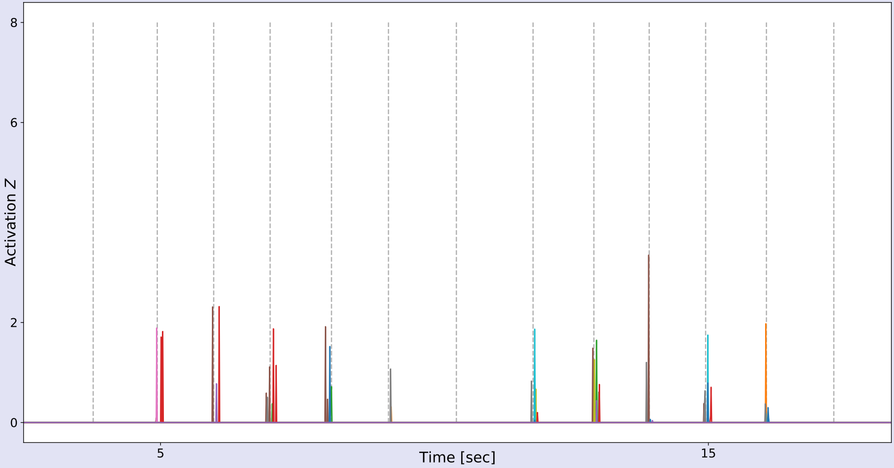
Protocol :

- Select 25 exercises and extract one step from each of them
- Encode other exercises using this pattern dictionary.

Details :

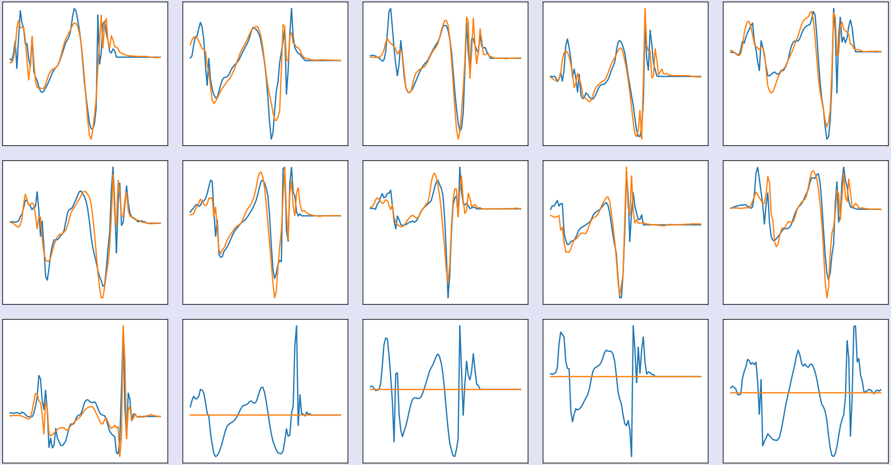
- Only used healthy patients in this study,
- Use the greedy CD to encode the signals and set $\lambda = 5$,

Encoding a walk signal



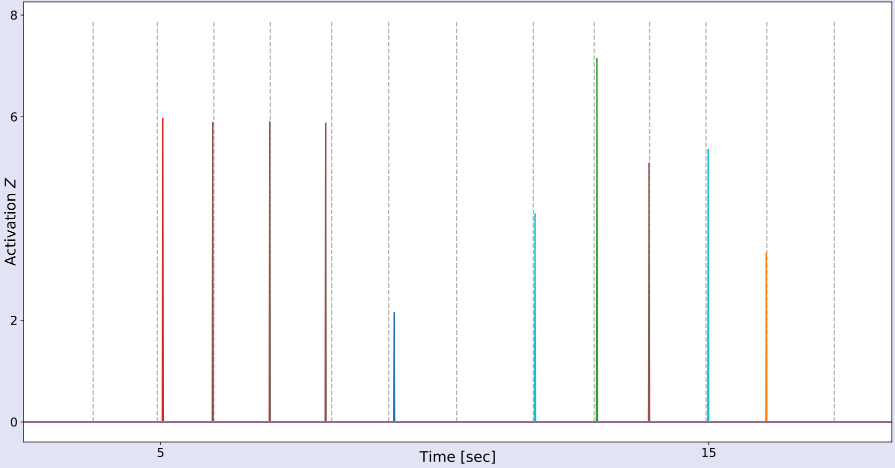
The activation are concentrated around the steps but there is some dispersion on multiple patterns.

Encoding a walk signal



Update of the dictionaries with 10 iterations of alternate minimization and FISTA updates for the dictionary.

Encoding a walk signal



The activation are more concentrated and only activate one pattern.

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.

References

Bradley, J. K., Kyrola, A., Bickson, D., and Guestrin, C. (2011). Parallel Coordinate Descent for ℓ_1-Regularized Loss Minimization. In *International Conference on Machine Learning (ICML)*, pages 321–328, Bellevue, WA, USA.

Bristow, H., Eriksson, A., and Lucey, S. (2013). Fast convolutional sparse coding. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 391–398, Portland, OR, USA.

Chalasani, R., Principe, J. C., and Ramakrishnan, N. (2013). A fast proximal method for convolutional sparse coding. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–5, Dallas, TX, USA.

Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2) :302–332.

Grosse, R., Raina, R., Kwong, H., and Ng, A. Y. (2007). Shift-Invariant Sparse Coding for Audio Classification. *Cortex*, 8 :9.

Kavukcuoglu, K., Sermanet, P., Boureau, Y.-l., Gregor, K., and Le Cun, Y. (2013). Learning Convolutional Feature Hierarchies for Visual Recognition. In *Advances in Neural Information Processing Systems (NIPS)*, number 1, pages 1–9.

Low, Y., Bickson, D., Gonzalez, J., Guestrin, C., Kyrola, A., and Hellerstein, J. M. (2012). Distributed GraphLab : a framework for machine learning and data mining in the cloud. *VLDB Endowment*, 5(8) :716–727.

Nesterov, Y. (2012). Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2) :341–362.

Nutini, J., Schmidt, M., Laradji, I. H., Friedlander, M. P., and Koepke, H. (2015). Coordinate Descent Converges Faster with the Gauss-Southwell Rule Than Random Selection. In *International Conference on Machine Learning (ICML)*, pages 1632–1641, Lille, France.

Osher, S. and Li, Y. (2009). Coordinate descent optimization for ℓ_1 minimization with application to compressed sensing ; a greedy algorithm. *Inverse Problems and Imaging*, 3(3) :487–503.

Scherrer, C., Tewari, A., Halappanavar, M., and Haglin, D. J. (2012). Feature Clustering for Accelerating Parallel Coordinate Descent. In *Advances in Neural Information Processing Systems (NIPS)*, pages 28–36, South Lake Tahoe, United States.

Vautard, R. and Ghil, M. (1989). Deterministic chaos, stochastic processes, and dimension. *Physica D : Nonlinear Phenomena*, 35(3) :395–424.

Yu, H.-F., Hsieh, C.-J., Si, S., and Dhillon, I. (2012). Scalable coordinate descent approaches to parallel matrix factorization