Modeling Brain Waveforms with Convolutional Dictionary Learning and Point Processes

Thomas Moreau
INRIA Saclay - MIND Team

Joint work with Tom Dupré La Tour, Mainak Jas, Alexandre Gramfort, Cédric Allain, Lindsey Power, Tim Bardouille

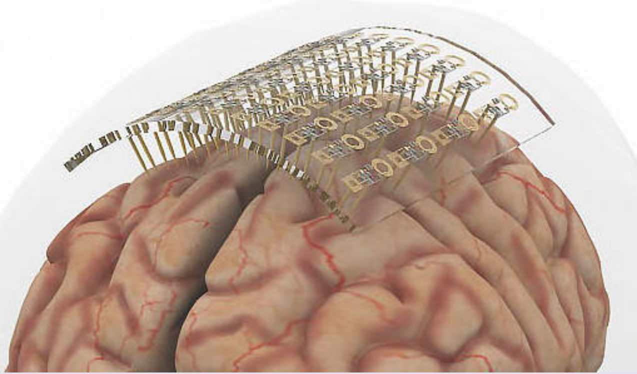
Goal: Study the brain mechanisms while it is functioning.

Outputs:

- **Functional Atlases:** Link areas of the brain to specific cognitive functions.
- **Functional Connectivity:** Highlight the information flow in the brain.
- **Healthcare:** Develop bio-markers for neurological disorders.

How to record living brains electrical activity: **Electrophysiology**

Direct measurement: intracranial EEG.



High Localization

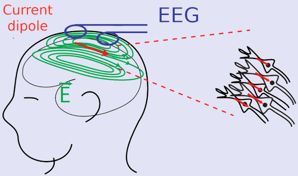
Low Resolution

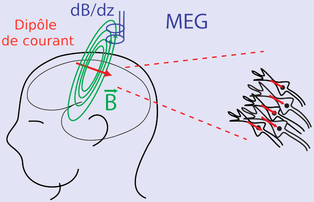
Invasive

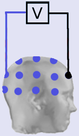
Context: functional Neuroimaging

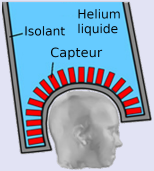
How to record living brains electrical activity: **Electrophysiology**

Remote measurement: M/EEG.



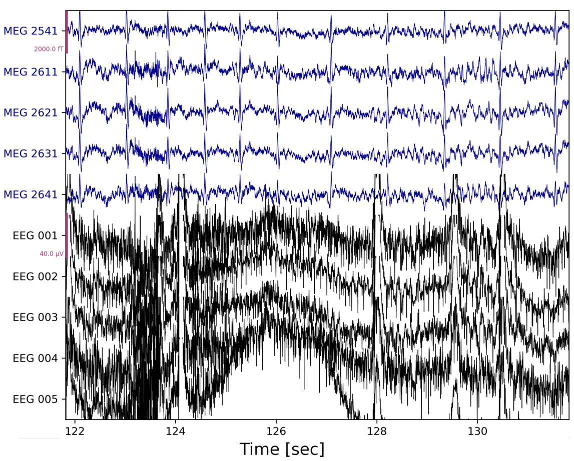






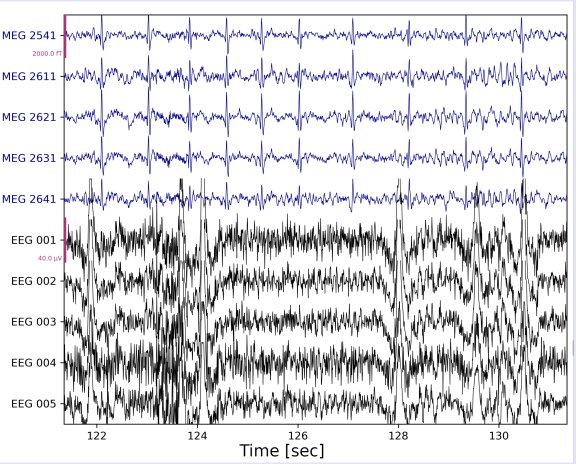
No Localization **Global** **Non Invasive**

Multivariate time-series X



Noisy **Many artifacts** **Complex**

Multivariate time-series X

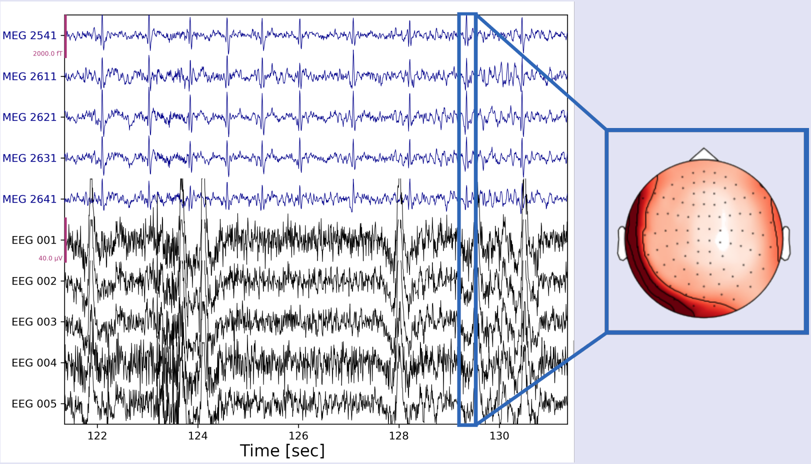


Noisy

Many artifacts

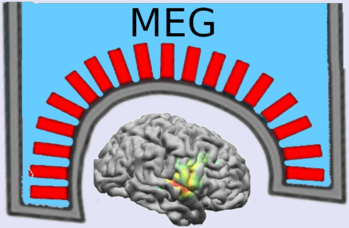
Complex

Multivariate time-series X

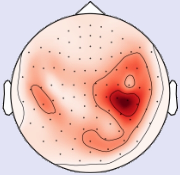


Noisy **Many artifacts** **Complex**

How to get back to electrical activity?



Maxwell's Equations



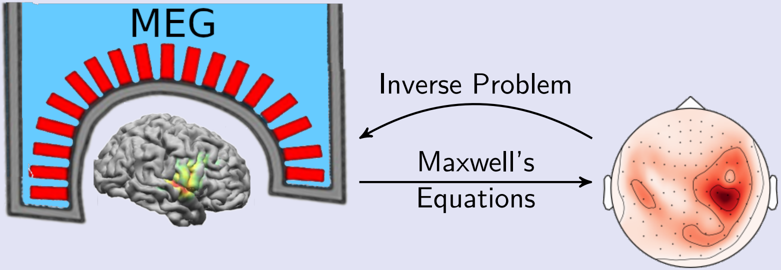
$\boxed{\varepsilon}$ **Electrical activity**

$\boxed{G}$

$\boxed{X}$ **Observed signal**

Forward model: $X = G\varepsilon$

How to get back to electrical activity?



$\boxed{\varepsilon}$ **Electrical activity**

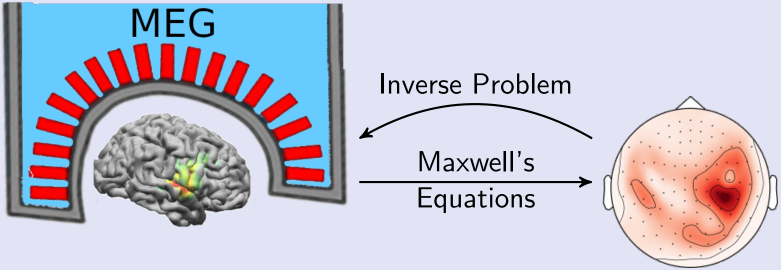
$\boxed{G}$

$\boxed{X}$ **Observed signal**

Forward model: $X = G\varepsilon$

Inverse problem: $\varepsilon = f(X)$ (ill-posed)

How to get back to electrical activity?



$\boxed{\varepsilon}$ **Electrical activity**

$\boxed{G}$

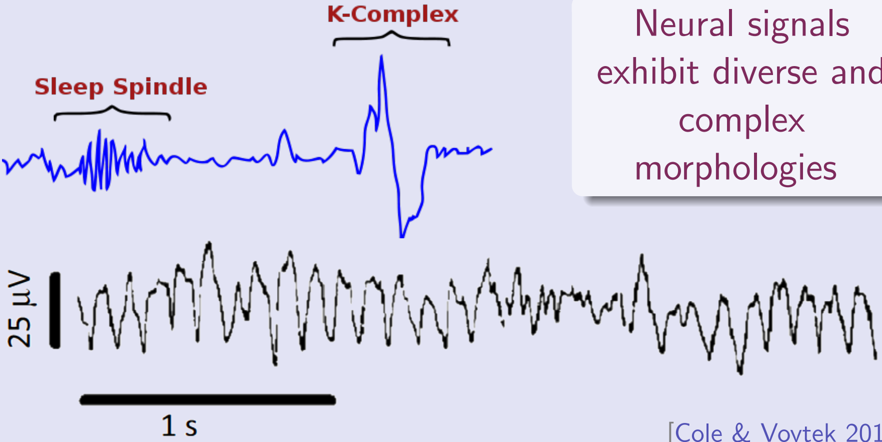
$\boxed{X}$ **Observed signal**

Forward model: $X = G\varepsilon$

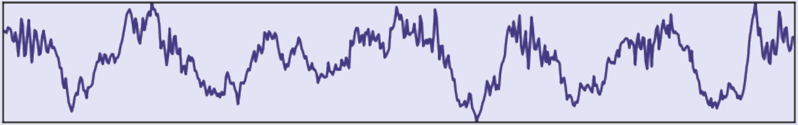
Inverse problem: $\varepsilon = f(X)$ (ill-posed)

- Dipole fit [Sarvas, 1987]
- Regularized optimization [Gramfort et al., 2012]
- Deep-learning [Hecker et al., 2021]

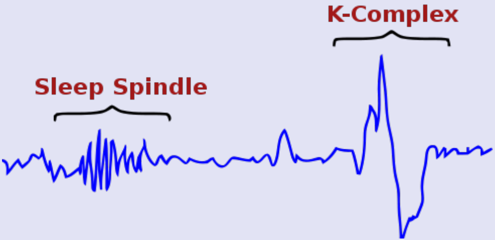
Neural signals exhibit diverse and complex morphologies



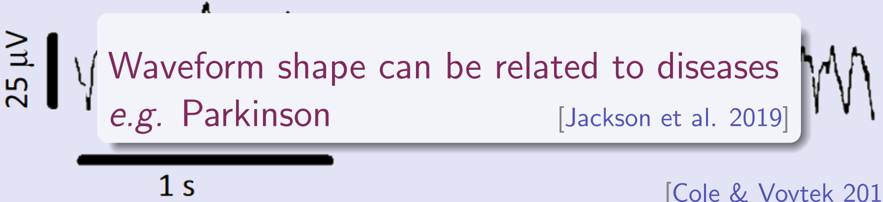
[Cole & Voytek 2017]



[Dupré la Tour et al. 2017]

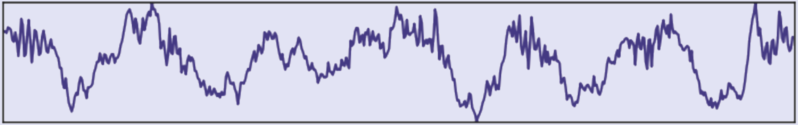


Neural signals exhibit diverse and complex morphologies



Waveform shape can be related to diseases
e.g. Parkinson [Jackson et al. 2019]

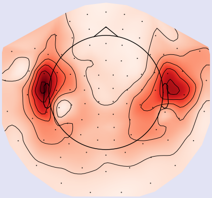
[Cole & Voytek 2017]

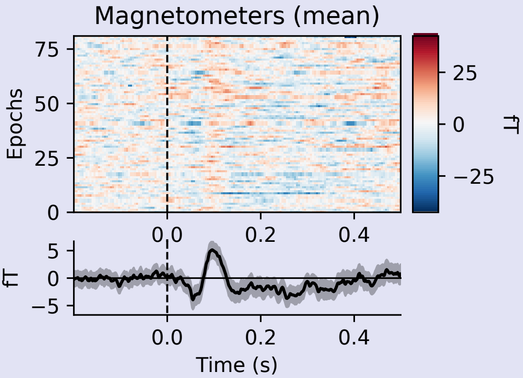


[Dupré la Tour et al. 2017]

- Subject is presented some stimuli – Audio, Visual, Motor, ...
- Record onset of the stimuli
- Average signal on window aligned around the stimulus

Evoked response to an auditory stimuli

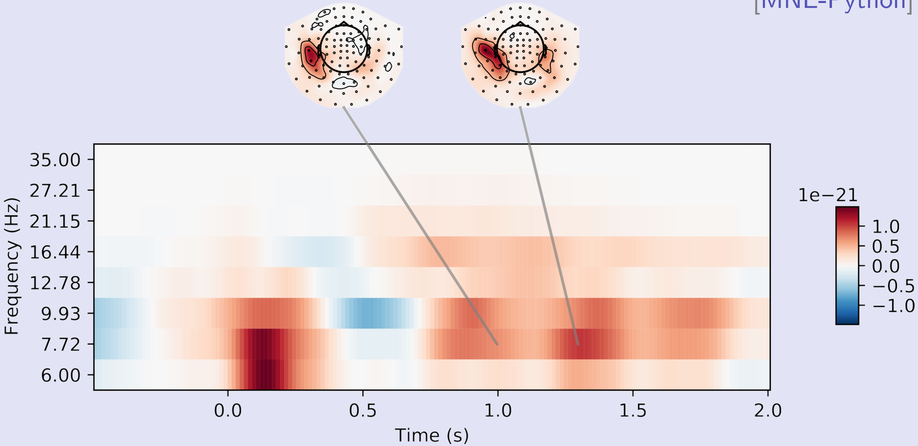




[MNE-Python]

- Subject is presented some stimuli – Audio, Visual, Motor, ...
- Average PSD on window aligned around the stimulus

[MNE-Python]



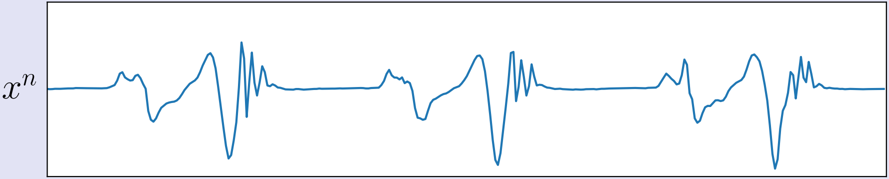
Evoked response to an somatosensory stimuli

Learning the waveform: Convolutional Dictionary Learning

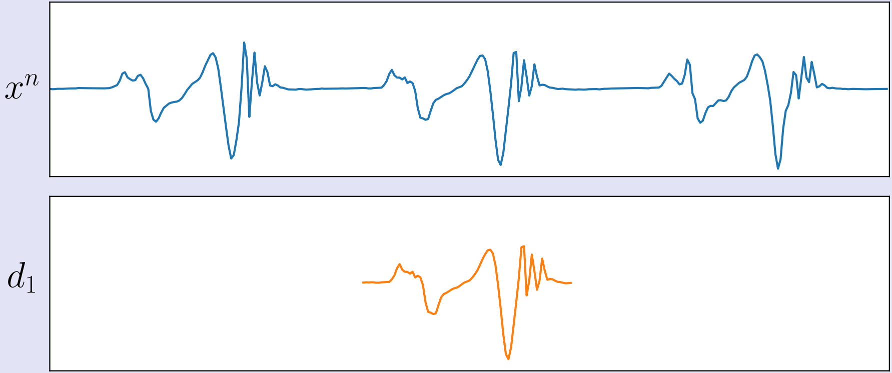
References

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

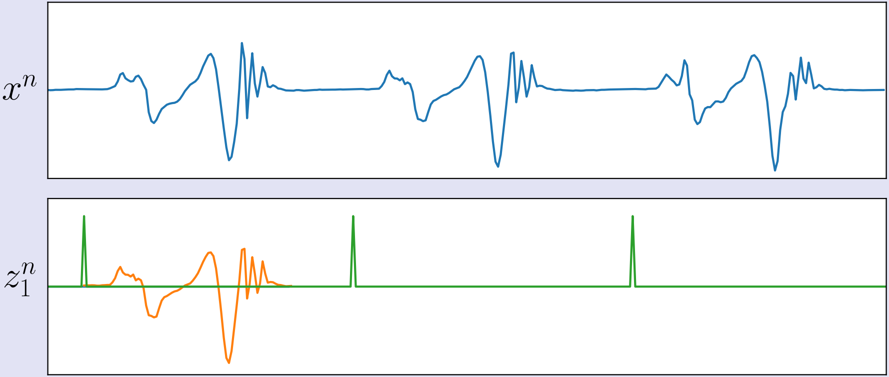
Local structure in signals



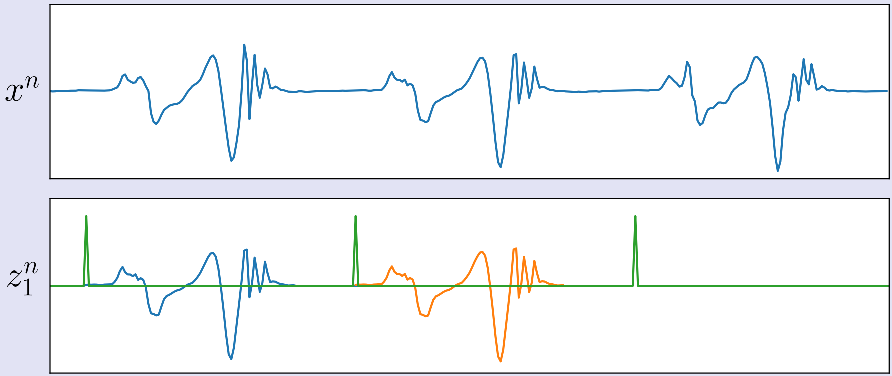
Local structure in signals



Local structure in signals

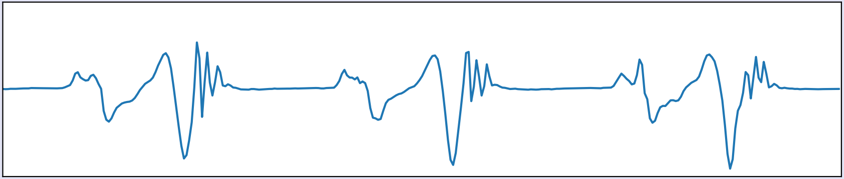


Local structure in signals

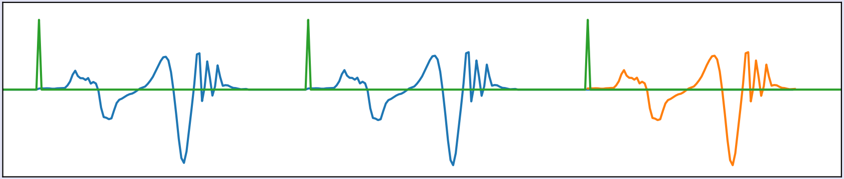


Key idea: decouple the localization of the patterns and their shape

x^n

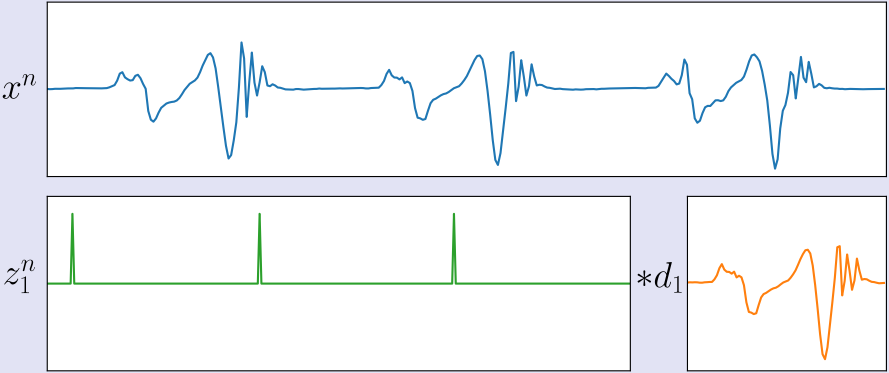


z_1^n



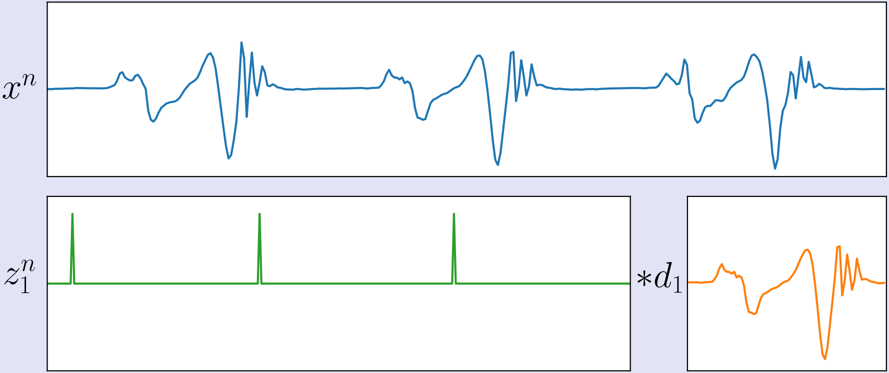
Local structure in signals

Key idea: decouple the localization of the patterns and their shape



Local structure in signals

Key idea: decouple the localization of the patterns and their shape



$$\boxed{x^n}[t] = \sum_{k=1}^{K} (\boxed{z_k^n} * \boxed{d_k})[t] + \varepsilon[t]$$

For a set of N univariate signals x^n, solve

$$\min_{d,z} \sum_{n=1}^{N} \frac{1}{2} \left\| \boxed{x^n} - \sum_{k=1}^{K} \boxed{z_k^n} * \boxed{d_k} \right\|_2^2 + \lambda \sum_{k=1}^{K} \|\boxed{z_k^n}\|_1,$$
$$\text{s.t.} \quad \|\boxed{d_k}\|_2^2 \leq 1$$

Hypothesis: patterns d_k are not present everywhere in the signal. They are localized in time.

$\Rightarrow$ Sparse activation signals z

Technical hypothesis: the patterns are in the ℓ_2-ball: $\|d_k\|_2^2 \leq 1$.

Optimization strategy

Bi-convex: The problem is not jointly convex in z_k^n, and d_k but it is convex in each block of coordinate.

Alternate minimization (*a.k.a.* Bloc Coordinate Descent):

- **Z-step:** given a fixed estimate of the atom, compute the activation signal z_k^n associated to each signal x^n.
- **D-step:** given a fixed estimate of the activation, update the atoms in the dictionary d_k.

Bi-convex: The problem is not jointly convex in z_k^n, and d_k but it is convex in each block of coordinate.

Alternate minimization (*a.k.a.* Bloc Coordinate Descent):

- **Z-step:** given a fixed estimate of the atom, compute the activation signal z_k^n associated to each signal x^n.
- **D-step:** given a fixed estimate of the activation, update the atoms in the dictionary d_k.

Unrolled optimization:

- **Z-step:** use an fixed differentiable procedure $f(x^n, D)$.
- **D-step:** learn D through back-propagation.

[Malezieux et al. 2022]

How to extend CSC to multivariate signals?

We can just use multivariate convolution,

$$\underbrace{X[t]}_{\in\mathbb{R}^P} = \sum_{k=1}^{K} \left(z_k * D_k\right)[t] = \sum_{k=1}^{K}\sum_{\tau=1}^{L} z_k[t-\tau]\underbrace{D_k[\tau]}_{\in\mathbb{R}^P}$$

with:

- X a multivariate signal of length T in $\mathbb{R}^P$
- D_k a multivariate signal of length L in $\mathbb{R}^P$
- z_k a univariate activation signal of length $\widetilde{T} = T - L + 1$

However, this model does not account for the physics of the problem.

Rank-1 constrained dictionary learning

References

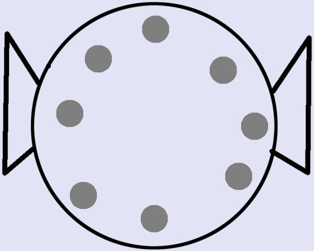
- Dupré la Tour, T., **Moreau, T.**, Jas, M., and Gramfort, A. (2018).

  Multivariate Convolutional Sparse Coding for Electromagnetic Brain Signals.

  In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3296–3306, Montreal, Canada

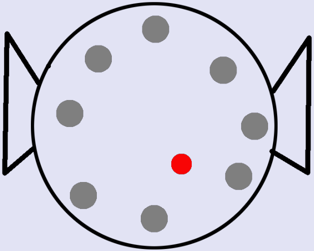
EM wave diffusion

- Recording here with 8 sensors



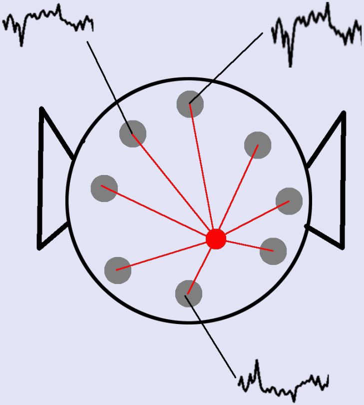
EM wave diffusion

- Recording here with 8 sensors
- EM activity in the brain



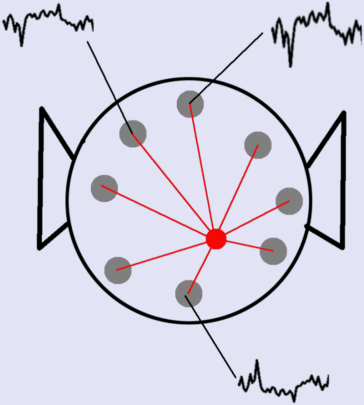
EM wave diffusion

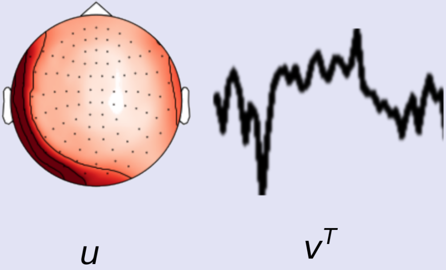
- Recording here with 8 sensors
- EM activity in the brain
- The electric field is spread **linearly** and **instantaneously** over all sensors (Maxwell equations)



EM wave diffusion

- Recording here with 8 sensors
- EM activity in the brain
- The electric field is spread **linearly** and **instantaneously** over all sensors (Maxwell equations)





Multivariate CSC with rank-1 constraint

Idea: Impose a rank-1 constraint on each dictionary atom D_k

To make the problem tractable, use u_k and v_k *s.t.* $D_k = u_k v_k^\top$.

$$\min_{u_k, v_k, z_k^n} \sum_{n=1}^{N} \frac{1}{2} \left\| X^n - \sum_{k=1}^{K} z_k^n * (u_k v_k^\top) \right\|_2^2 + \lambda \sum_{k=1}^{K} \left\| z_k^n \right\|_1 , \tag{1}$$
$$\text{s.t.} \quad \|u_k\|_2^2 \leq 1 \;, \; \|v_k\|_2^2 \leq 1 \text{ and } z_k^n \geq 0 \;.$$

Here,

- $u_k \in \mathbb{R}^P$ is a spatial pattern
- $v_k \in \mathbb{R}^L$ is a temporal pattern

$\Rightarrow$ This is a tri-convex problem

Optimization strategy

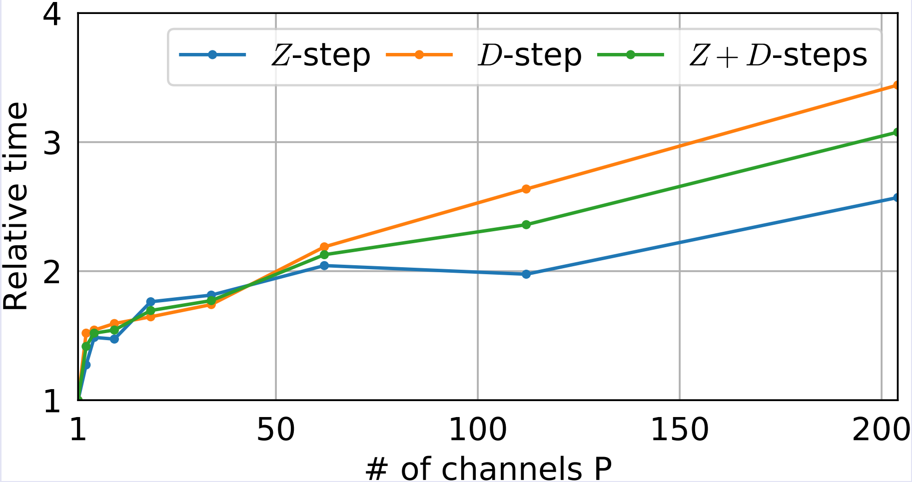
Tri-convex: The problem is not jointly convex in z_k^n, u_k and v_k but it is convex in each block of coordinate.

We can use a block coordinate descent, aka alternate minimization, to converge to a local minima of this problem. The 3 following steps are applied alternatively:

- **Z-step:** given a fixed estimate of the atom, compute the activation signal z_k^n associated to each signal X^n. (LGCD)
- **u-step:** given a fixed estimate of the activation and temporal pattern, update the spatial pattern u_k. (PGD)
- **v-step:** given a fixed estimate of the activation and spatial pattern, update the temporal pattern v_k. (PGD)

Good scaling in the number of channels P

Scaling relative to P on somato dataset with $T = 134,700$, $K = 2$, and $L = 128$



Test the pattern recovery capabilities of our method on simulated data,

$$X^n = \sum_{k=1}^{2} z_k * (u_k v_k^\top) + \mathcal{E}$$

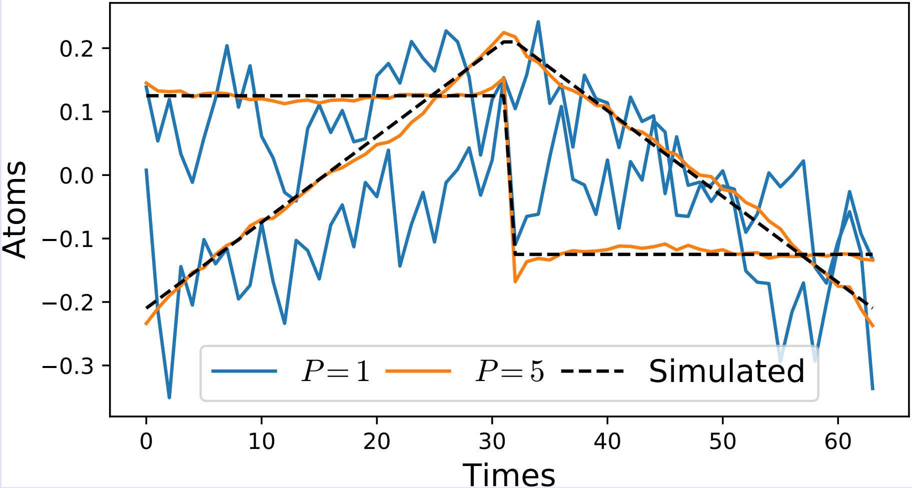
where (u_k, v_k) are chosen patterns of rank-1 and the activated coefficient $z_k^n[t]$ are drawn uniformly and their value are uniform in $[0, 1]$.

The noise $\mathcal{E}$ is generated as a gaussian white noise with variance σ.

We set $N = 100$, $L = 64$ and $\widetilde{T} = 640$

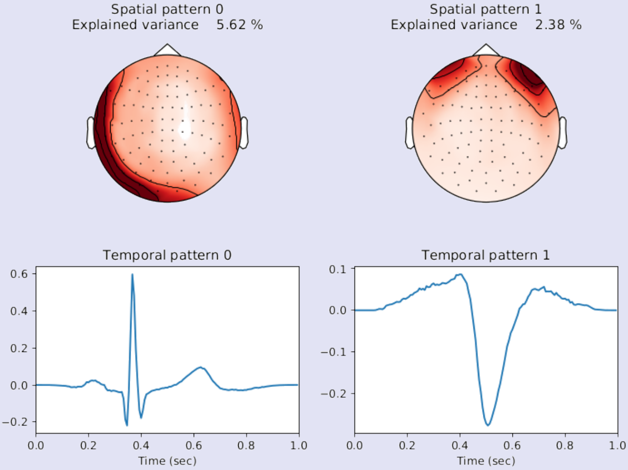
Pattern recovery

Patterns recovered with $P = 1$ and $P = 5$. The signals were generated with the two simulated temporal patterns and with $\sigma = 10^{-3}$.

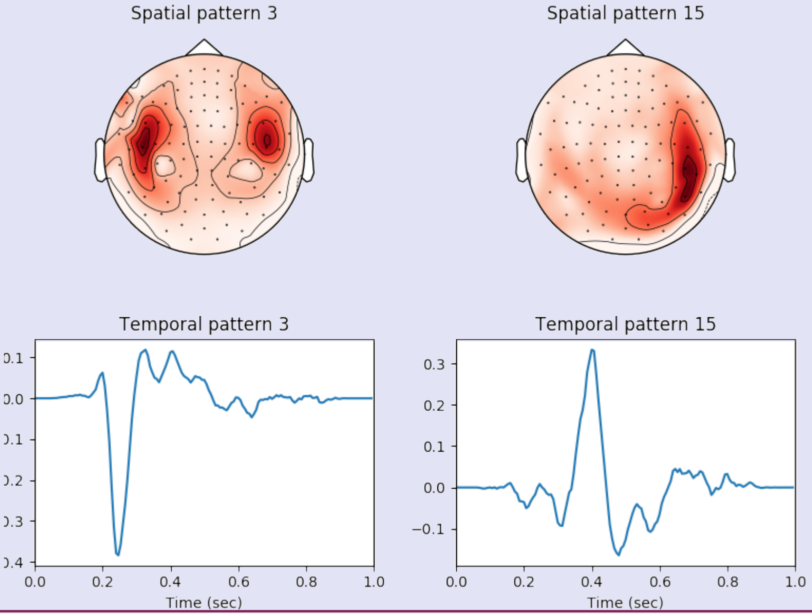


MNE sample data

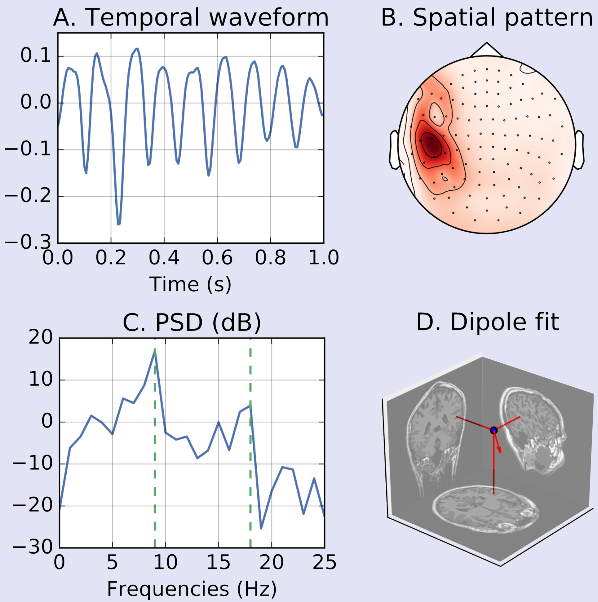
A selection of temporal waveforms of the atoms learned on the MNE sample dataset.



Learned atoms – Evoked response



Learned atoms – Induced responses



Search the docs ...

alphaCSC: Convolution sparse coding for time-series

unittests passing codecov 82%

On this page
Installation
Quickstart
Dicodile backend

This is a library to perform shift-invariant sparse dictionary lea
(CSC), on time-series data. It includes a number of different r

1. univariate CSC
2. multivariate CSC
3. multivariate CSC with a rank-1 constraint [1]
4. univariate CSC with an alpha-stable distribution [2]

A mathematical descriptions of these models is available in the documentation.

Python code online:
https://alphacsc.github.io

```
pip install alphacsc
```

Installation

To install this package, the easiest way is using `pip`. It will install this package and its dependencies. The `setup.py` depends on `numpy` and `cython` for the installation so it is advised to install them beforehand. To install this package, please run one of the two commands:

Examples reproduce figures from this talk!

(Latest stable version)

```
pip install alphacsc
```

(Development version)

```
pip install git+https://github.com/alphacsc/alphacsc.git#egg=alphacsc
```

(Dicodile backend)

```
pip install numpy cython
pip install alphacsc[dicodile]
```

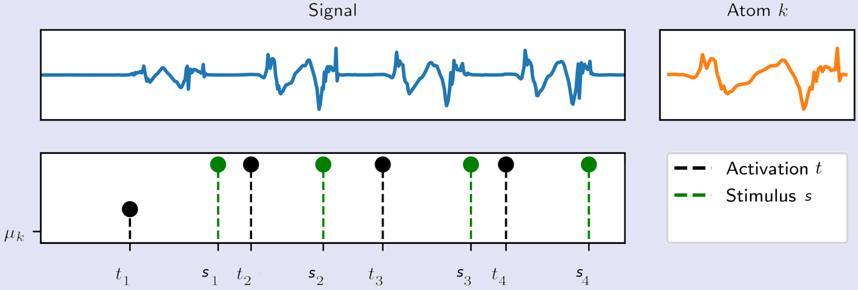
Modeling stimuli induced patterns with Point Processes

References

- Allain, C., Gramfort, A., and **Moreau, T.** (2022). DriPP: Driven Point Process to Model Stimuli Induced Patterns in M/EEF Signals.
  In *International Conference on Learning Representations (ICLR)*
- Staerman, G., Allain, C., Gramfort, A., and **Moreau, T.** (2023). FaDIn: Fast Discretized Inference for Hawkes Processes with General Parametric Kernels.
  In *International Conference on Machine Learning (ICML)*, Honolulu, HI, USA. PMLR

Stimuli Induced Patterns

- Manual pattern identification
- No quantification of how stimuli influence patterns activation.



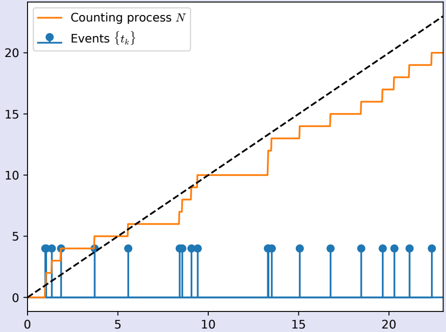
Activations and stimuli can be seen as *Point Processes*.

- Stochastic model for stream of events
- Time of arrival $\{t_k\}$ associated with counting process $N(t)$
- Characterized by the intensity:

$$\lambda(t|\mathcal{F}_t) = \lim_{dt \to 0} \frac{P(N(t+dt) - N(t) = 1|\mathcal{F}_t)}{dt}$$

Poisson process with constant probability of arrival

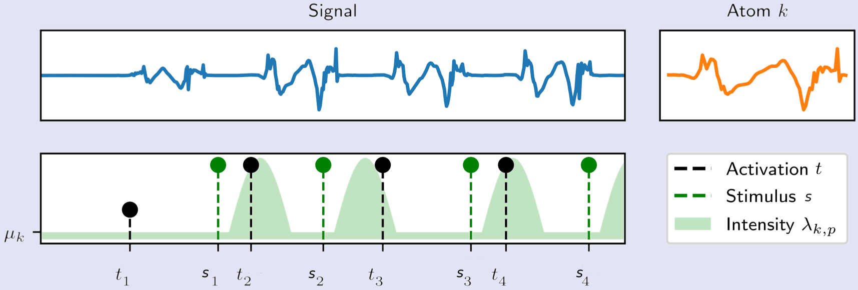
$$\lambda(t) = \mu_0$$



DriPP – Driven Point Process

Idea: Model the probability of activation $\{t_k\}$ depending on the PP from the stimuli $\{s_p\}$.

$$\lambda(t|\mathcal{F}_t) = \lambda(t|\{s_p; s_p < t\}) = \mu_0 + \sum_{s_p<t} \kappa(t - s_p)$$

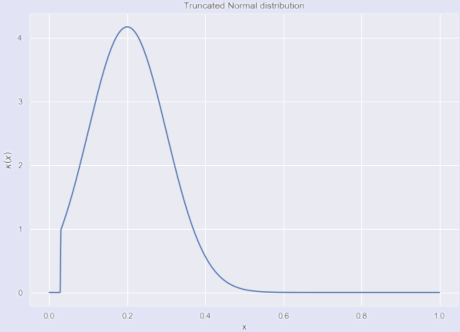


Modeling latency

Chosing a model for stimuli based modeling:

$$\lambda(t|\mathcal{F}_t) = \mu_0 + \sum_{s_p<t} \alpha\kappa(t - s_p)$$

- $\mu_0 \geq 0$: spontaneous activity.
- $\alpha \geq 0$: allow for stimuli to have no effect.
- $\kappa(\tau)$: pdf of a truncated Gaussian $\mathcal{N}(m, \sigma^2)$ to model latency.



Parameters estimation

The negative log-likelihood of the model can be computed using the intensity λ:

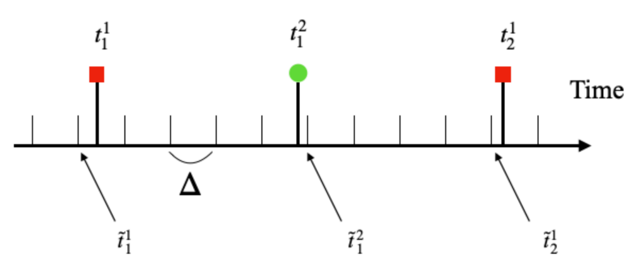
$$\mathcal{L}(\{t_k\};\Theta) = \int_0^T \lambda(t)dt - \sum_{t_k} \log \lambda(t_k)$$
$$= \mu_0 T + \alpha|\{t_k\}| - \sum_{t_k} \log(\mu_0 \sum_{s_p<t_k} \alpha\kappa(t_k - s_p))$$

with $\Theta = (\mu_0, \alpha, m, \sigma^2)$

$\Rightarrow$ Parameter estimation with an EM algorithm.

- Slow EM algorithm
- Not general for parametric kernels.

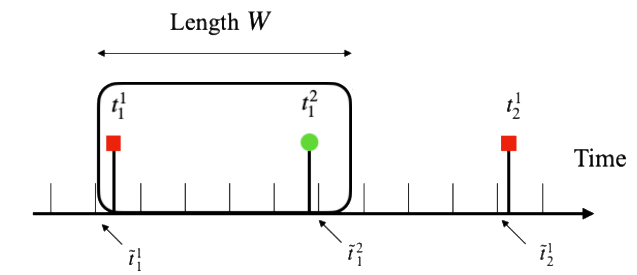
FaDIn Inference method for general parametric kernels



Discretization

FaDIn Inference method for general parametric kernels

- **Discretization**



Finite support

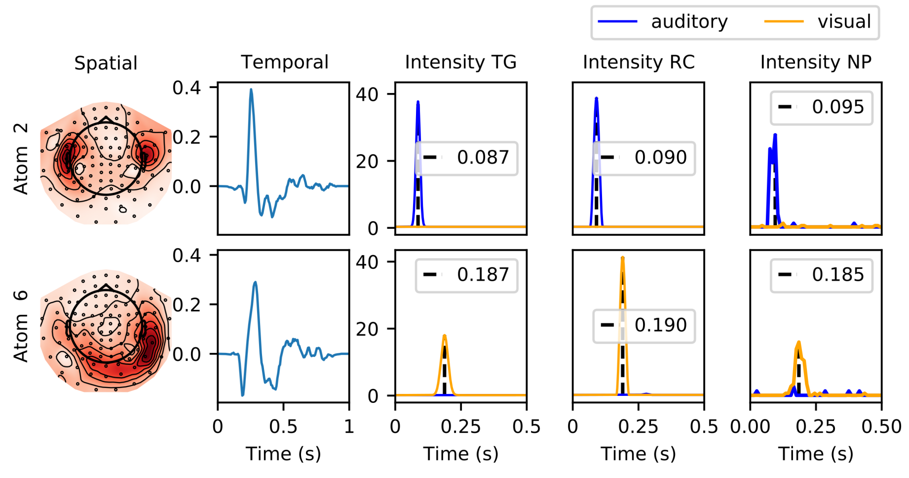
FaDIn Inference method for general parametric kernels

- **Discretization**
- **Finite support**
- ℓ_2 **loss**

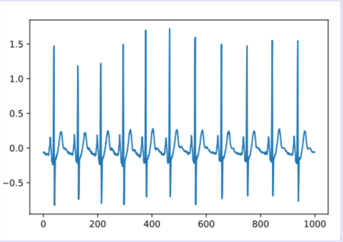
$$\mathcal{L}(\{t_k\};\theta) = \sum_{t=0}^{T} \frac{1}{2} \|z[t] - (z * k)[t]\|_2^2$$

with $z[t] = 1$ if $t \in \{t_k\}$, 0 otherwise.

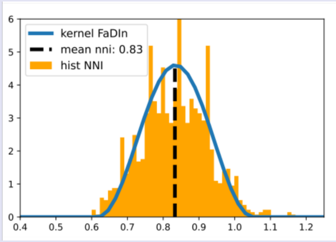
Results for evoked atoms - samples



Results for heart rate variability



CDL + FaDIn



Conclusion

- CDL can learn recurring patterns in multivariate signals.
- Converts the signal into a stream of events.
- PP framework can model the activation distribution.

Limitations and on-going work:

- Not easy to apply to population level.
- DriPP does not model inhibition.
- CDL and PP are separated.

Benchopt

References

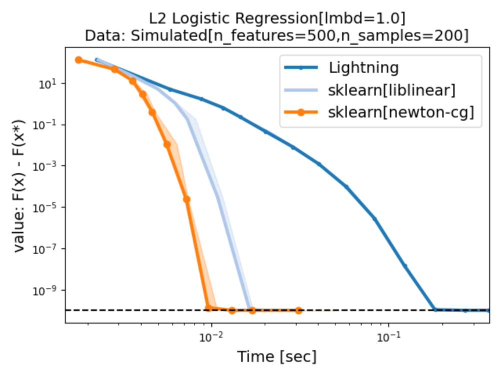
- **Moreau, T.**, Massias, M., Gramfort, A., Ablin, P., Bannier, P.-A., Charlier, B., Dagréou, M., la Tour, T. D., Durif, G., Dantas, C. F., Klopfenstein, Q., Larsson, J., Lai, E., Lefort, T., Malézieux, B., Moufad, B., Nguyen, B. T.,

  Rakotomamonjy, A., Ramzi, Z., Salmon, J., and Vaiter, S. (2022). Benchopt: Reproducible, efficient and collaborative optimization benchmarks.

  In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, New-Orlean, LA, USA. Curran Associates, Inc.

Doing a benchmark for the ℓ_2 regularized logistic regression with multiple solvers and datasets is now easy as calling:

```
git clone https://github.com/benchopt/benchmark_logreg_l2
benchopt run ./benchmark_logreg_l2
```



Benchmark: principle

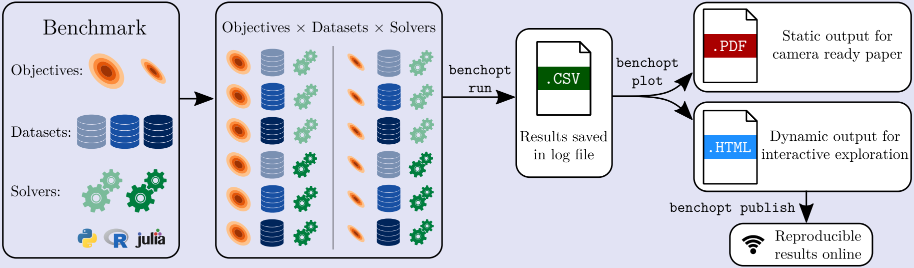
A benchmark is a directory with:

- An `objective.py` file with an `Objective`
- A directory `solvers` with one file per `Solver`
- A directory `datasets` with `Dataset` generators/fetchers

```
my_benchmark/
├── README.rst
├── datasets
│   ├── simulated.py  # some dataset
│   └── real.py  # some dataset
├── objective.py  # contains the definition of the objective
└── solvers
    ├── solver1.py  # some solver
    └── solver2.py  # some solver
```

The `benchopt` client runs a cross product and generates a csv file + convergence plots like above.

Benchopt: principle



$\Rightarrow$ Each object can be parametrized so multiple scenario can be tested.

Automatizing tasks:

- Automatic installation of competitors solvers.
- Parametrized datasets, objectives and solvers and run on cross products.
- Make sure to quantify the variance.
- Automatic caching.
- Interactive visualization of the results
- Automatic parallelization, run on SLURM,
- ...?

Thanks for your attention!

Code available online:

alphacsc : alphacsc.github.io

DriPP : github.com/CedricAllain/dripp

benchopt : benchopt.github.io

Slides are on my web page:

tommoral.github.io @tomamoral

Z-step: Locally greedy coordinate descent (LGCD)

N independent problem such that

$$\min_{z_k^n \geq 0} \frac{1}{2} \left\| X^n - \sum_{k=1}^{K} z_k^n * D_k \right\|_2^2 + \lambda \sum_{k=1}^{K} \left\| z_k^n \right\|_1 \quad .$$

This problem is convex in z_k and can be solved with different techniques:

- FISTA [Chalasani et al., 2013]
- ADMM [Bristow et al., 2013]
- L-BFGS [Jas et al., 2017]
- Greedy CD [Kavukcuoglu et al., 2010]

⇒ These methods can be slow for long signals as the complexity of each iteration is at least linear in the length of the signal.

Z-step: Locally greedy coordinate descent (LGCD)

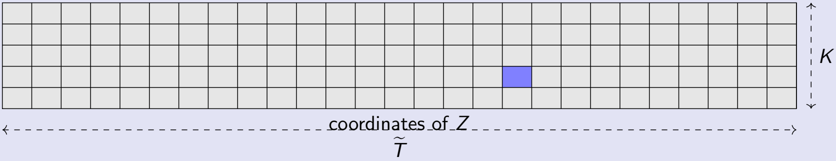
Coordinate Descent: only 1 coordinate is updated at each iteration:

1. The coordinate $z_{k_0}[t_0]$ is updated to its optimal value $z'_{k_0}[t_0]$ when all other coordinate are fixed.

2. The updated coordinate is chosen

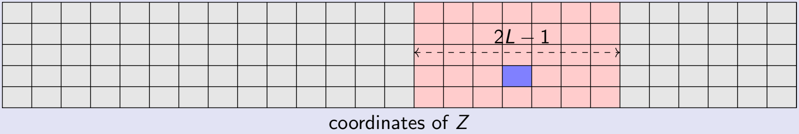
   - Cyclic: $\mathcal{O}(1)$ [Friedman et al., 2007]
   - Randomized: $\mathcal{O}(1)$ [Nesterov, 2010]
   - Greedy: $\mathcal{O}(K\widetilde{T})$ [Osher and Li, 2009]
     by maximizing $|z_k[t] - z'_k[t]|$
   - Locally Greedy: $\mathcal{O}(K\widetilde{L})$ [Moreau et al., 2018]
     by maximizing $|z_k[t] - z'_k[t]|$ on a window

We introduced the LGCD method which is an extension of GCD.



GCD has $\mathcal{O}(K\widetilde{T})$ computational complexity.

We introduced the LGCD method which is an extension of GCD.

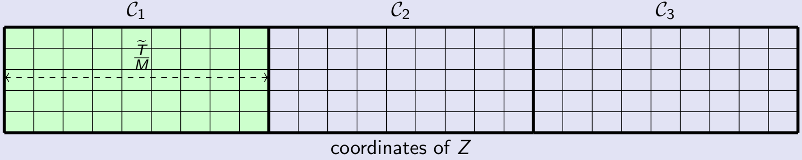


GCD has $\mathcal{O}(K\widetilde{T})$ computational complexity.

But the update itself has complexity $\mathcal{O}(KL)$

Locally Greedy Coordinate Descent [Moreau et al., 2018]

We introduced the LGCD method which is an extension of GCD.

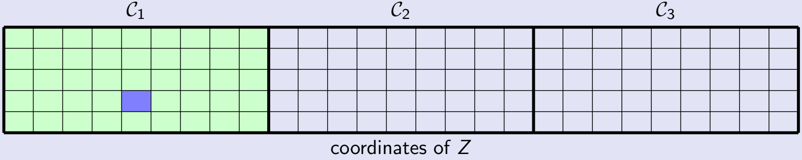


With a partition $\mathcal{C}_m$ of the signal domain $[1, K] \times [0, \widetilde{T}[$,

$$\mathcal{C}_m = [1, K] \times [\frac{(m-1)\widetilde{T}}{M}, \frac{m\widetilde{T}}{M}[$$

Locally Greedy Coordinate Descent [Moreau et al., 2018]

We introduced the LGCD method which is an extension of GCD.



With a partition $\mathcal{C}_m$ of the signal domain $[1, K] \times [0, \widetilde{T}[$,

$$\mathcal{C}_m = [1, K] \times [\frac{(m-1)\widetilde{T}}{M}, \frac{m\widetilde{T}}{M}[$$

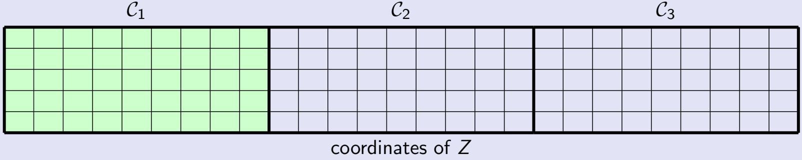
The coordinate to update is chosen greedily on a sub-domain $\mathcal{C}_m$

$$\frac{\widetilde{T}}{M} = 2L - 1 \quad \Rightarrow \quad \mathcal{O}(\text{Coordinate selection}) = \mathcal{O}(\text{Coordinate Update})$$

The overall iteration complexity is $\mathcal{O}(KL)$ instead of $\mathcal{O}(K\widetilde{T})$.

$\Rightarrow$ Efficient for sparse Z

We introduced the LGCD method which is an extension of GCD.



With a partition $\mathcal{C}_m$ of the signal domain $[1, K] \times [0, \widetilde{T}[$,

$$\mathcal{C}_m = [1, K] \times [\frac{(m-1)\widetilde{T}}{M}, \frac{m\widetilde{T}}{M}[$$

The coordinate to update is chosen greedily on a sub-domain $\mathcal{C}_m$

$$\frac{\widetilde{T}}{M} = 2L - 1 \quad \Rightarrow \quad \mathcal{O}(\text{Coordinate selection}) = \mathcal{O}(\text{Coordinate Update})$$

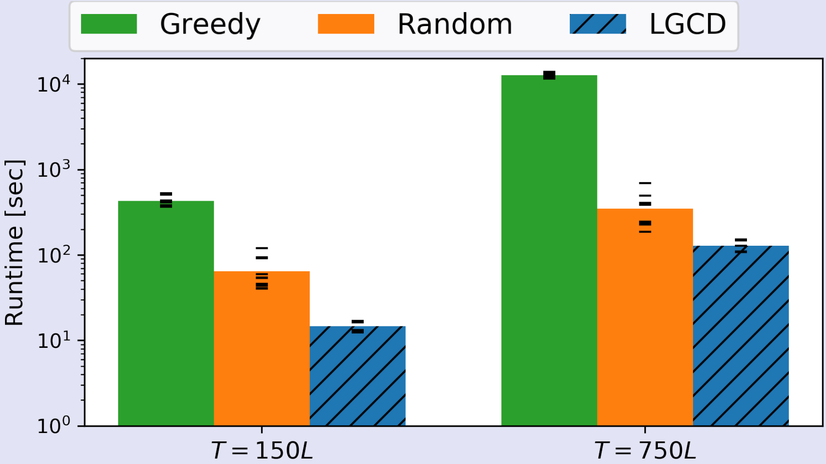
The overall iteration complexity is $\mathcal{O}(KL)$ instead of $\mathcal{O}(K\widetilde{T})$.

$\Rightarrow$ Efficient for sparse Z $\Rightarrow$ Can be efficiently parallelized.

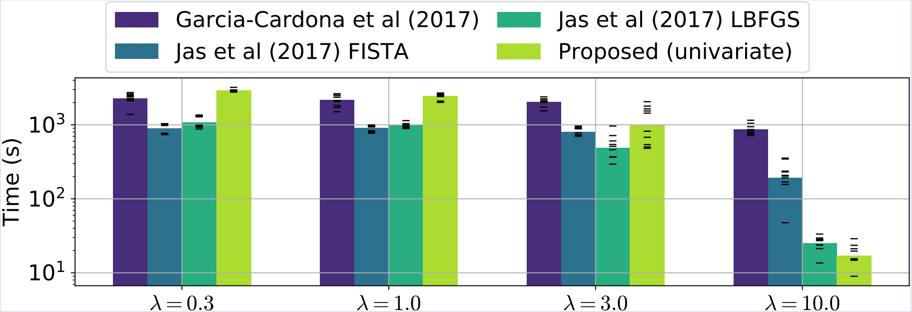
Fast optimization

Comparison of the coordinate selection strategy for CD on simulated signals

We set $K = 10$, $L = 150$, $\lambda = 0.1\lambda_{\text{max}}$



Comparison with univariate methods on somato dataset with
$T = 134,700$, $K = 8$ and $L = 128$



Fast optimization

Comparison with multivariate methods on somato dataset with $T = 134,700,\ K = 8,\ P = 5$ and $L = 128$