Learning to optimize with unrolled algorithms

Thomas Moreau INRIA Saclay

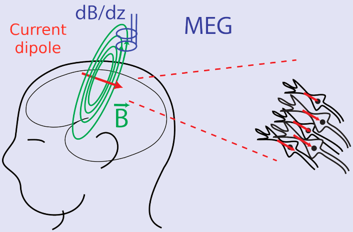
Joint work with Pierre Ablin; Mathurin Massias; Alexandre Gramfort; Hamza Cherkaoui; Jeremias Sulam; Joan Bruna

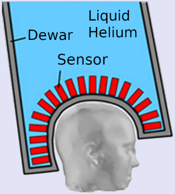
PARIETAL

Inria informatics mathematics

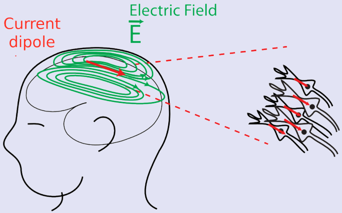
Electrophysiology

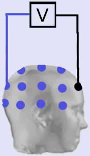
Magnetoencephalography



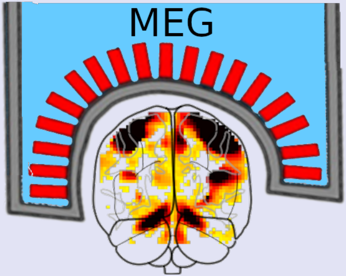


Electroencephalography

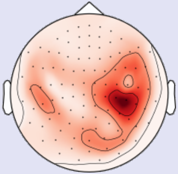




Inverse problems



Maxwell's Equations →



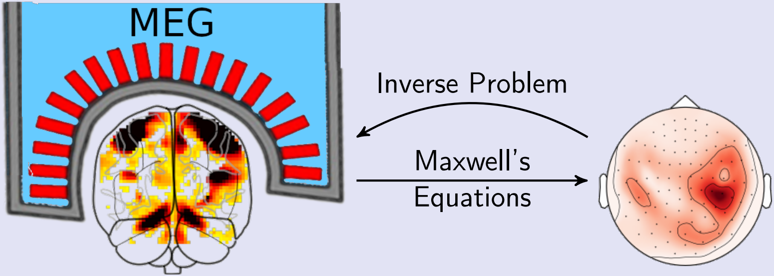
z — Electrical activity

D

x — Observed signal

Forward model: $x = Dz$

Inverse problems



z — Electrical activity

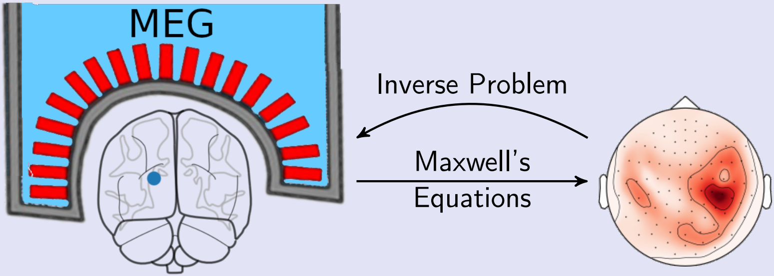
D

x — Observed signal

Forward model: $x = Dz$

Inverse problem: $z = f(x)$ (ill-posed)

Inverse problems



z D x

Electrical activity **Observed signal**

Forward model: $x = Dz$ **Inverse problem:** $z = f(x)$ (ill-posed)

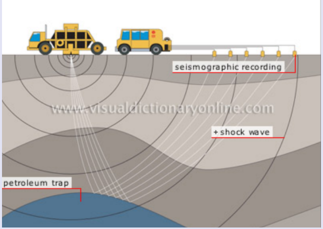
Optimization with a regularization $\mathcal{R}$ encoding prior knowledge

$$\text{argmin}_z \|x - Dz\|_2^2 + \mathcal{R}(z)$$

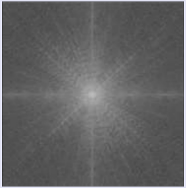
Example: sparsity with $\mathcal{R} = \lambda \| \cdot \|_1$

Other Inverse Problems

Ultra sound

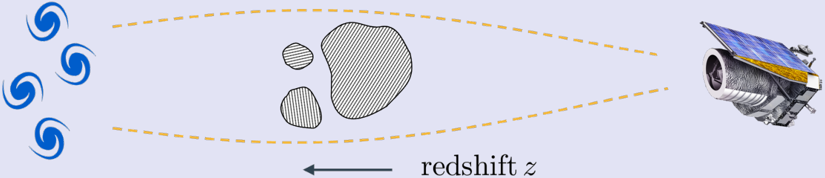


fMRI - compress sensing



Astrophysic

galaxies here ...tell us about... structures here



redshift z

Classical Sparse IP resolution - the Lasso [Tibshirani 1996]

Given a forward operator $D \in \mathbb{R}^{n \times m}$ and $\lambda > 0$, the Lasso for $x \in \mathbb{R}^n$ is

$$z^* = \underset{z}{\operatorname{argmin}}\, F_x(z) = \underbrace{\frac{1}{2}\|x - Dz\|_2^2}_{f_x(z)} + \lambda \|z\|_1$$

a.k.a. sparse coding, sparse linear regression, ...

We are interested in the over-complete case where $m > n$.

Classical Sparse IP resolution - the Lasso [Tibshirani 1996]

Given a forward operator $D \in \mathbb{R}^{n \times m}$ and $\lambda > 0$, the Lasso for $x \in \mathbb{R}^n$ is

$$z^* = \underset{z}{\operatorname{argmin}}\, F_x(z) = \underbrace{\frac{1}{2}\|x - Dz\|_2^2}_{f_x(z)} + \lambda \|z\|_1$$

a.k.a. sparse coding, sparse linear regression, ...

We are interested in the over-complete case where $m > n$.

Some Properties

- The problem is convex in z but not strongly convex in general.
- The problem is L-smooth and proximable.
- Most of the time, there is a unique solution (but not always).

Classical Optimizationd

- (Fast-) Iterative Shrinkage-Thresholding Algorithm (ISTA). [Daubechies et al. 2004; Beck and Teboulle 2009]
- Coordinate Descent. [Friedman et al. 2007; Osher and Li 2009]
- Least-Angle Regression (LARS). [Efron et al. 2004]

Convergence rates – Worst case analysis: For any x,

$$F_x(z^{(t)}) - F_x^* \leq \mathcal{O}\left(\frac{1}{t^2}\right)$$

$\Rightarrow$ Guaranteed convergence for any x.

How to solve efficiently many inverse problems

Given multiple inputs x_i, we would like to solve efficiently:

$$\min_{z_i} \sum_{i=1}^{N} F_{x_i}(z_i)$$

How to solve efficiently many inverse problems

Given multiple inputs x_i, we would like to solve efficiently:

$$\min_{z_i} \sum_{i=1}^{N} F_{x_i}(z_i)$$

Here, each problem is independent, so with an infinite budget, there is no point in considering this problem.

How to solve efficiently many inverse problems

Given multiple inputs x_i, we would like to solve efficiently:

$$\min_{z_i} \sum_{i=1}^{N} F_{x_i}(z_i)$$

Here, each problem is independent, so with an infinite budget, there is no point in considering this problem.

However, if your aim is to chose an algorithm f_L for a given computational budget L such that

$$\operatorname*{argmin}_{f_L} \frac{1}{N} \sum_{i=1}^{N} F_{x_i}(f_L(x_i))$$

Can you do better than worst case algorithms?

How to solve efficiently many inverse problems

Given multiple inputs x_i, we would like to solve efficiently:

$$\min_{z_i} \sum_{i=1}^{N} F_{x_i}(z_i)$$

Here, each problem is independent, so with an infinite budget, there is no point in considering this problem.

However, if your aim is to chose an algorithm f_L for a given computational budget L such that

$$\operatorname*{argmin}_{f_L} \frac{1}{N} \sum_{i=1}^{N} F_{x_i}(f_L(x_i))$$

Can you do better than worst case algorithms?

Related to average case complexity analysis?

[Scieur and Pedregosa 2020; Pedregosa and Scieur 2020]

Unrolled optimization algorithms

ISTA: Iterative Shrinkage-Thresholding Algorithm

[Daubechies et al. 2004]

f_x is a L-smooth function with $L = \|D\|_2^2$ and

$$\nabla f_x(z^{(t)}) = D^\top (Dz^{(t)} - x)$$

The ℓ_1-norm is proximable with a separable proximal operator

$$\text{prox}_{\mu\|\cdot\|_1}(x) = \text{sign}(x) \max(0, |x| - \mu) = ST(x, \mu)$$

ISTA: Iterative Shrinkage-Thresholding Algorithm

[Daubechies et al. 2004]

f_x is a L-smooth function with $L = \|D\|_2^2$ and

$$\nabla f_x(z^{(t)}) = D^\top(Dz^{(t)} - x)$$

The ℓ_1-norm is proximable with a separable proximal operator

$$\mathrm{prox}_{\mu\|\cdot\|_1}(x) = \mathrm{sign}(x)\max(0, |x| - \mu) = ST(x, \mu)$$

We can use the proximal gradient descent algorithm (ISTA)

$$z^{(t+1)} = \mathrm{ST}\left(z^{(t)} - \rho \underbrace{\nabla f_x(z^{(t)})}_{D^\top(Dz^{(t)} - x)}, \rho\lambda\right)$$

Here, ρ play the role of a step size (in $[0, \frac{2}{L}[$).

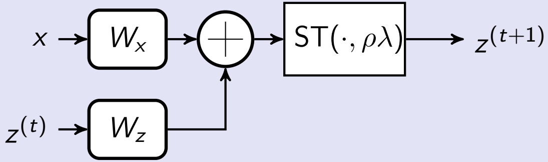
ISTA

$$z^{(t+1)} = \mathrm{ST}\left(z^{(t)} - \rho D^\top (Dz^{(t)} - x), \rho\lambda\right)$$

Let $W_z = I_m - \rho D^\top D$ and $W_x = \rho D^\top$. Then

$$z^{(t+1)} = \mathrm{ST}(W_z z^{(t)} + W_x x, \rho\lambda)$$

One step of ISTA



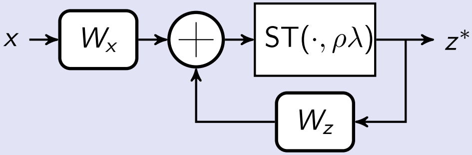
ISTA

$$z^{(t+1)} = \mathrm{ST}\left(z^{(t)} - \rho D^\top (Dz^{(t)} - x), \rho\lambda\right)$$

Let $W_z = I_m - \rho D^\top D$ and $W_x = \rho D^\top$. Then

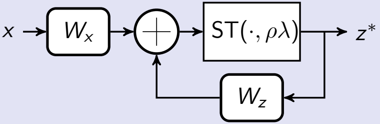
$$z^{(t+1)} = \mathrm{ST}(W_z z^{(t)} + W_x x, \rho\lambda)$$

RNN equivalent to ISTA

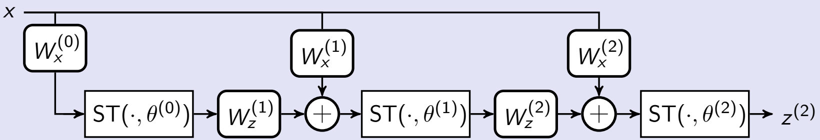


Recurrence relation of ISTA define a RNN

$$z^{(t+1)} = \mathrm{ST}\left(z^{(t)} - \frac{1}{L}D^\top(Dz^{(t)} - x), \frac{\lambda}{L}\right)$$



This RNN can be unfolded as a feed-forward network.



Let $\Phi_{\Theta^{(T)}}$ denote a network with T layers parametrized with $\Theta^{(T)}$.

If $W_x^{(i)} = W_x$ and $W_z^{(i)} = W_z$, then $\Phi_{\Theta^T}(x) = z^{(t)}$.

LISTA – Training

Empirical risk minimization : We need a training set of $\{x_1, \dots x_N\}$ training sample and our goad is to accelerate ISTA on unseen data $x \sim p$.

The training solves

$$\tilde{\Theta}^{(T)} \in \arg\min_{\Theta^{(T)}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_x(\Phi_{\Theta^{(T)}}(x_i)) \ .$$

for a loss $\mathcal{L}_x$.

$\Rightarrow$ Choice of loss $\mathcal{L}_x$?

LISTA – Training

Supervised: a ground truth $z^*(x)$ is known

$$\mathcal{L}_x(z) = \frac{1}{2}\|z - z^*(x)\|$$

Solving the inverse problem.

LISTA – Training

Supervised: a ground truth $z^*(x)$ is known

$$\mathcal{L}_x(z) = \frac{1}{2}\|z - z^*(x)\|$$

Solving the inverse problem.

Semi-supervised: the solution of the Lasso $z^*(x)$ is known

$$\mathcal{L}_x(z) = \frac{1}{2}\|z - z^*(x)\|$$

Accelerating the resolution of the Lasso.

LISTA – Training

Supervised: a ground truth $z^*(x)$ is known

$$\mathcal{L}_x(z) = \frac{1}{2}\|z - z^*(x)\|$$

Solving the inverse problem.

Semi-supervised: the solution of the Lasso $z^*(x)$ is known

$$\mathcal{L}_x(z) = \frac{1}{2}\|z - z^*(x)\|$$

Accelerating the resolution of the Lasso.

Unsupervised: there is no ground truth

$$\mathcal{L}_x(z) = F_x(z) = \frac{1}{2}\|x - Dz\|_2^2 + \lambda\|z\|_1$$

Solving the Lasso.

LISTA – Training

Supervised: a ground truth $z^*(x)$ is known

$$\mathcal{L}_x(z) = \frac{1}{2}\|z - z^*(x)\|$$

Solving the inverse problem.

Semi-supervised: the solution of the Lasso $z^*(x)$ is known

$$\mathcal{L}_x(z) = \frac{1}{2}\|z - z^*(x)\|$$

Accelerating the resolution of the Lasso.

Unsupervised: there is no ground truth

$$\mathcal{L}_x(z) = F_x(z) = \frac{1}{2}\|x - Dz\|_2^2 + \lambda\|z\|_1$$

Solving the Lasso.

LISTA – Parametrizations

General LISTA model [Gregor and Le Cun 2010]

$$z^{(t+1)} = \mathrm{ST}\left(\mathrm{W}_{\mathrm{e}}^{(\mathrm{t})} z^{(t)} + \mathrm{W}_{\mathrm{x}}^{(\mathrm{t})} x, \theta^{(\mathrm{t})}\right)$$

The structure of D is lost in the linear transform.

Coupled LISTA [Chen et al. 2018]

$$z^{(t+1)} = \mathrm{ST}\left(z^{(t)} - \alpha^{(\mathrm{t})} \mathrm{W}^{(\mathrm{t})}(Dz^{(t)} - x), \beta^{(\mathrm{t})}\right)$$

Can be seen as learning

- Pre-conditionner $W^{(t)} \in \mathbb{R}^{m \times n}$
- Step-size $\alpha^{(t)} \in \mathbb{R}_+$
- Threshold $\beta^{(t)} \in \mathbb{R}_+$

LISTA – Parametrizations

General LISTA model [Gregor and Le Cun 2010]

$$z^{(t+1)} = \mathrm{ST}\left(\mathrm{W}_e^{(t)} z^{(t)} + \mathrm{W}_x^{(t)} x, \theta^{(t)}\right)$$

The structure of D is lost in the linear transform.

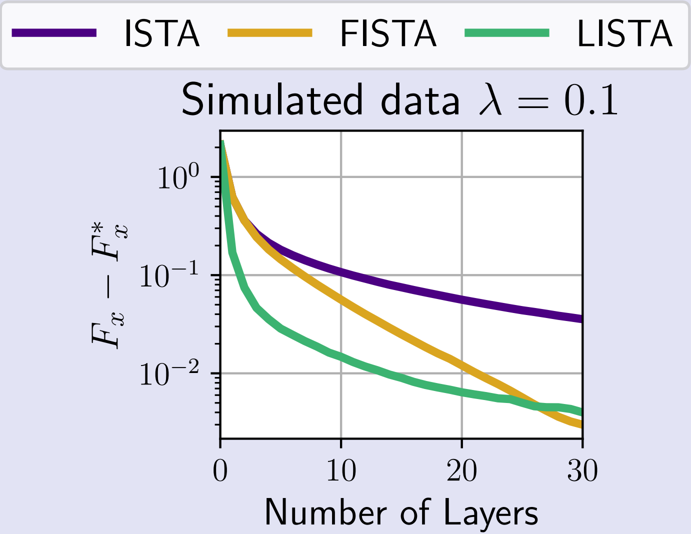
Coupled LISTA [Chen et al. 2018]

$$z^{(t+1)} = \mathrm{ST}\left(z^{(t)} - \alpha^{(t)} \mathrm{W}^{(t)}(Dz^{(t)} - x), \beta^{(t)}\right)$$

Can be seen as learning

- Pre-conditionner $W^{(t)} \in \mathbb{R}^{m \times n}$
- Step-size $\alpha^{(t)} \in \mathbb{R}_+$
- Threshold $\beta^{(t)} \in \mathbb{R}_+$

$\Rightarrow$ Justified theoretically for (un)supervised convergence


ISTA
FISTA
LISTA
Simulated data $\lambda = 0.1$
$F_x - F_x^*$
10^0
10^{-1}
10^{-2}
0
10
20
30
Number of Layers

What is learned in unrolled algorithms?

Coupled LISTA Parametrizations

Coupled LISTA [Chen et al. 2018]

$$z^{(t+1)} = \mathrm{ST}\left(z^{(t)} - \alpha^{(t)} \mathrm{W}^{(t)} (Dz^{(t)} - x), \beta^{(t)}\right)$$

Can be seen as learning

- Pre-conditionner $W^{(t)} \in \mathbb{R}^{m \times n}$
- Step-size $\alpha^{(t)} \in \mathbb{R}_+$
- Threshold $\beta^{(t)} \in \mathbb{R}_+$

Theorem – Asymptotic convergence of the weights

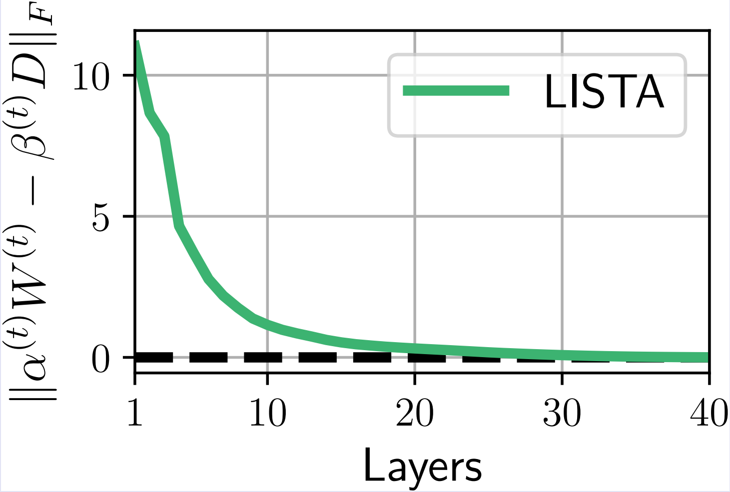
Consider a sequence of nested networks $\Phi_{\Theta^{(T)}}$ *s.t.* $\Phi_{\Theta^{(t)}}(x) = \phi_{\theta^{(t)}}(\Phi_{\Theta^{(t+1)}}(x), x)$. Assume that

1. the sequence of parameters converges *i.e.* $\theta^{(t)} \xrightarrow[t\to\infty]{} \theta^* = (W^*, \alpha^*, \beta^*)$,
2. the output of the network converges toward a solution $z^*(x)$ of the Lasso uniformly over the equiregularization set $\mathcal{B}_\infty$, *i.e.* $\sup_{x\in\mathcal{B}_\infty} \|\Phi_{\Theta^{(T)}}(x) - z^*(x)\| \xrightarrow[T\to\infty]{} 0$.

Then $\frac{\alpha^*}{\beta^*} W^* = D$.

Idea of the proof: each unit vector needs to be a fixed point of the network.

Numerical verification



40-layers LISTA network trained on a 10×20 problem with $\lambda = 0.1$

The weights $W^{(t)}$ align with D and α, β get coupled.

Is there a point in learning step sizes?

LISTA with restricted parametrization : Only learn a step-size $\alpha^{(t)}$

$$z^{(t+1)} = \mathrm{ST}\left(z^{(t)} - \alpha^{(\mathrm{t})} D^\top (Dz^{(t)} - x), \lambda\alpha^{(\mathrm{t})}\right)$$

Fewer parameters: T instead of $(2 + mn)T$.

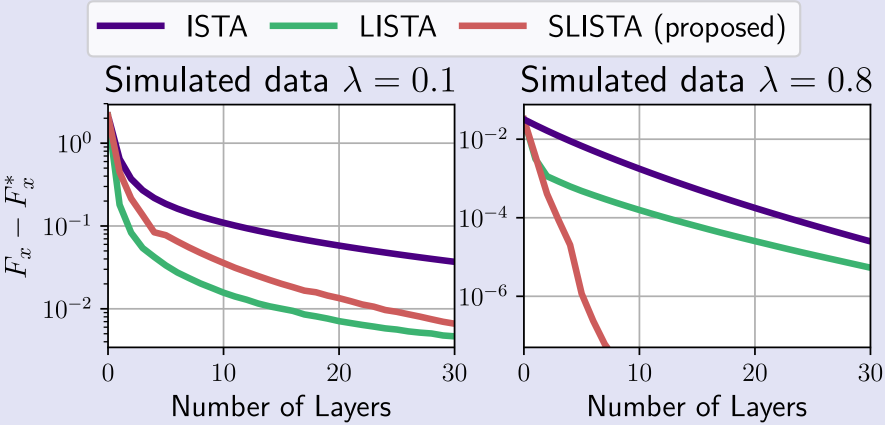
$\Rightarrow$ Easier to learn

$\Rightarrow$ Reduced performances?

Goal: Learn step sizes for ISTA adapted to input distribution.

Simulated data: $m = 256$ and $n = 64$

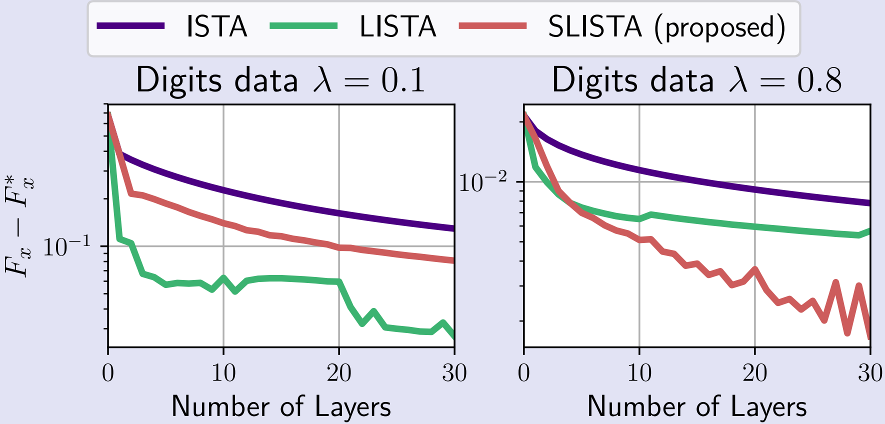
$$D_k \sim \mathcal{U}(\mathcal{S}^{n-1}) \text{ and } x = \frac{\widetilde{x}}{\|D^\top \widetilde{x}\|_\infty} \text{ with } \widetilde{x}_i \sim \mathcal{N}(0,1)$$



Performance on semi-real datasets

Digits: 8×8 images from `scikit-learn`

D_k and $\widetilde{x}$ sampled uniformly from the digits and $x = \frac{\widetilde{x}}{\|D^\top \widetilde{x}\|_\infty}$.



ISTA: Majoration-Minimization

Taylor expansion of f_x in $z^{(t)}$

$$
\begin{aligned}
F_x(z) &= f_x(z^{(t)}) + \nabla f_x(z^{(t)})^\top (z - z^{(t)}) + \frac{1}{2}\|D(z - z^{(t)})\|_2^2 + \lambda\|z\|_1 \\
&\leq f_x(z^{(t)}) + \nabla f_x(z^{(t)})^\top (z - z^{(t)}) + \frac{L}{2}\|z - z^{(t)}\|_2^2 + \lambda\|z\|_1
\end{aligned}
$$

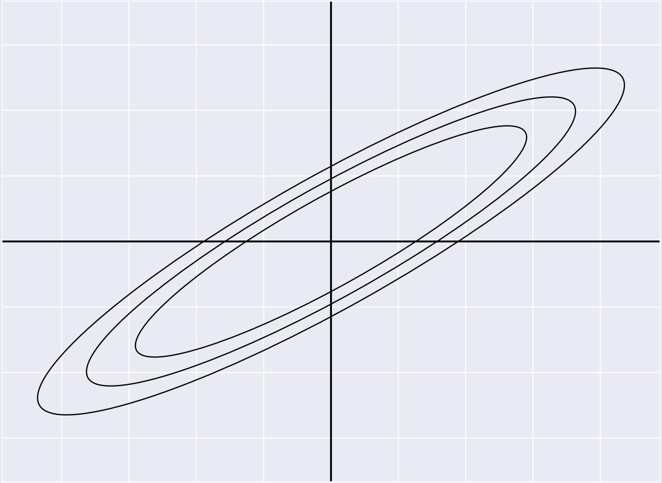
$\Rightarrow$ Replace the Hessian $D^\top D$ by L **Id**.

Separable function that can be minimized in close form

$$
\begin{aligned}
\operatorname*{argmin}_z \frac{L}{2}\left\|z^{(t)} - \frac{1}{L}\nabla f_x(z^{(t)}) - z\right\|_2^2 + \lambda\|z\|_1 &= \operatorname{prox}_{\frac{\lambda}{L}}\left(z^{(t)} - \frac{1}{L}\nabla f_x(z^{(t)})\right) \\
&= \operatorname{ST}\left(z^{(t)} - \frac{1}{L}\nabla f_x(z^{(t)}), \frac{\lambda}{L}\right)
\end{aligned}
$$

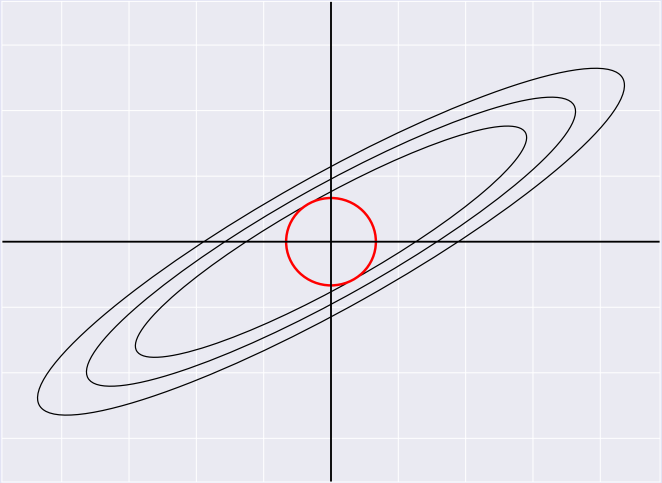
ISTA: Majoration for the data-fit

- Level sets from $z^\top D^\top D z$

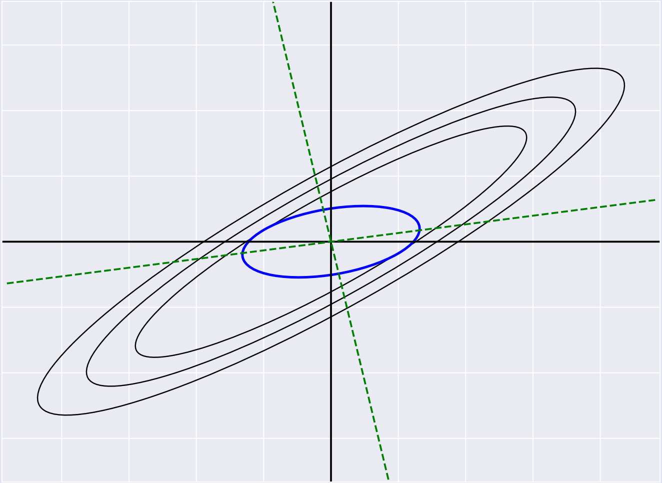


ISTA: Majoration for the data-fit

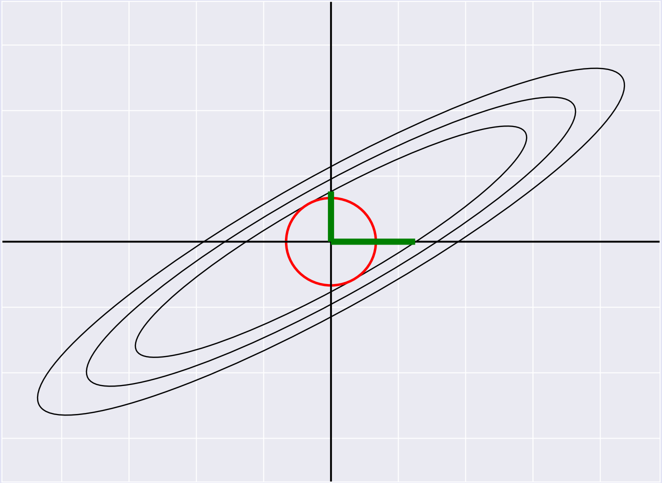
- Level sets from $z^\top D^\top D z \leq L\|z\|_2$



- Level sets from $z^\top D^\top D z \leq z^\top A^\top \Lambda A z$ [Moreau and Bruna 2017]



- Level sets from $z^\top D^\top Dz \leq L_S\|z\|_2$ for $Supp(z) \subset S$

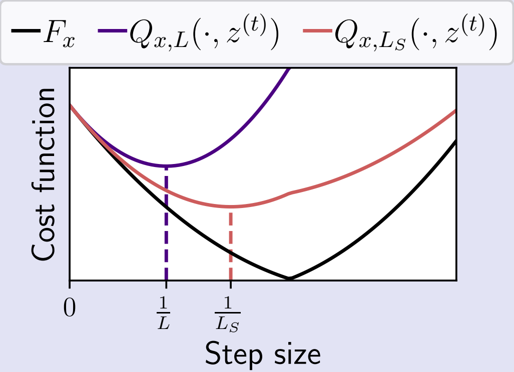


Oracle ISTA: Majoration-Minimization

For all z such that $\mathrm{Supp}(z) \subset S \doteq \mathrm{Supp}(z^{(t)})$,

$$F_x(z) \leq f_x(z^{(t)}) + \nabla f_x(z^{(t)})^\top (z - z^{(t)}) + \frac{L_S}{2}\|z - z^{(t)}\|_2^2 + \lambda \|z\|_1$$

with $L_S = \|D_{\cdot,S}\|_2^2$.



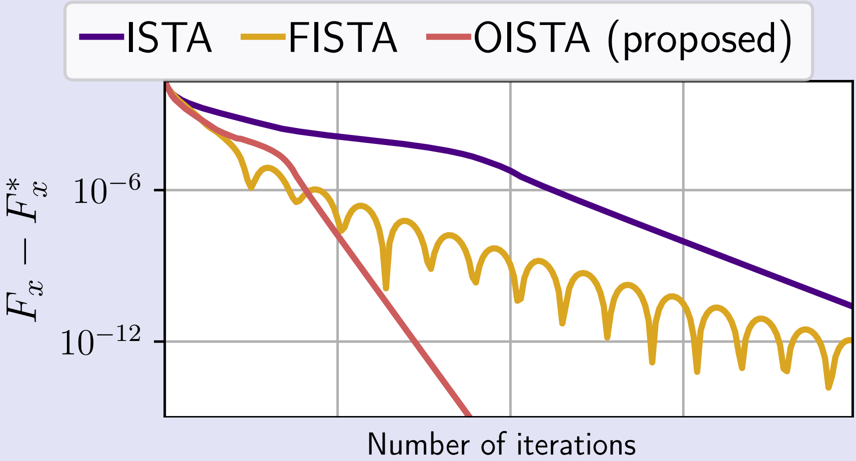
Oracle ISTA (OISTA):

1. Get the Lipschitz constant L_S associated with support $S = \text{Supp}(z^{(t)})$.
2. Compute $y^{(t+1)}$ as a step of ISTA with a step-size of $1/L_S$

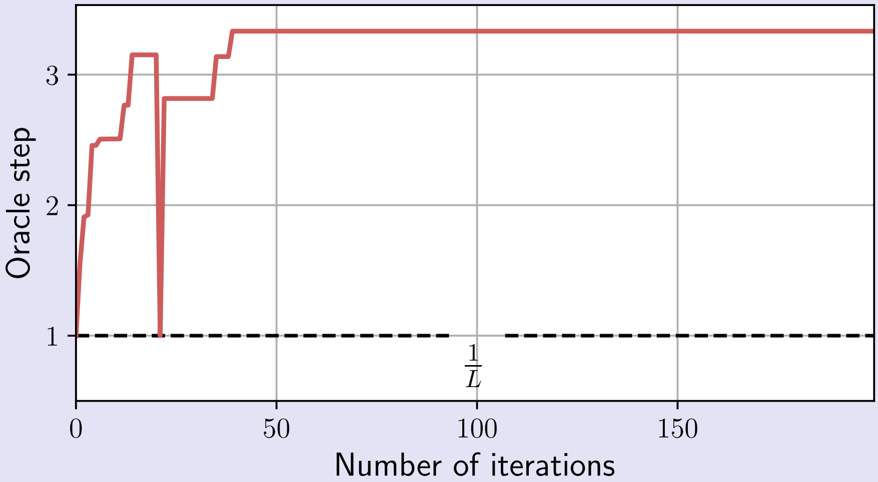
$$y^{(t+1)} = \text{ST}\left(z^{(t)} - \frac{1}{L_S}D^\top(Dz^{(t)} - x), \frac{\lambda}{L_S}\right)$$

3. If $\text{Supp}(y^{t+1}) \subset S$, accept the update $z^{(t+1)} = y^{(t+1)}$.
4. Else, $z^{(t+1)}$ is computed with step size $1/L$.

OISTA: Performances

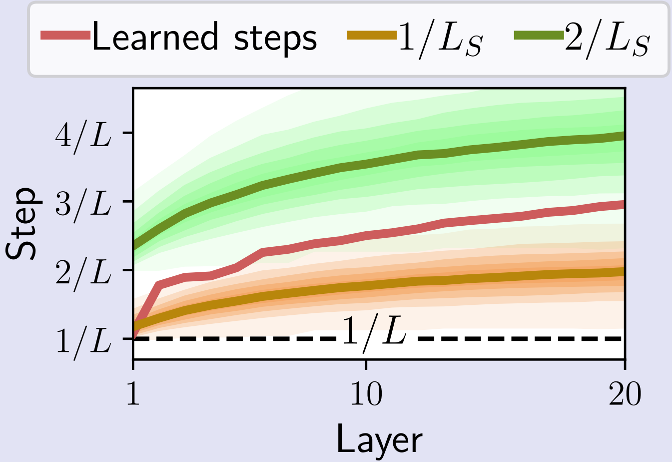


OISTA – Step-size



OISTA – Limitation

- OISTA is not practical, as you need to compute L_S at each iteration and this is costly.
- No precomputation possible: there is an exponential number of supports S.



The learned step-sizes are linked to the distribution of $1/L_S$

My take on unrolled algorithms

What unrolled algorithms *can* do:

- Improve constants in convergence rate.
  [Moreau and Bruna 2017]
- Learn to better optimize for a non-uniform input distribution.
  [Ablin et al. 2019]
- Make inverse problem solution differentiable.
  [Ablin et al. 2020; Mehmood and Ochs 2020]

My take on unrolled algorithms

What unrolled algorithms *can* do:

- Improve constants in convergence rate.
  [Moreau and Bruna 2017]
- Learn to better optimize for a non-uniform input distribution.
  [Ablin et al. 2019]
- Make inverse problem solution differentiable.
  [Ablin et al. 2020; Mehmood and Ochs 2020]

What unrolled algorithms *can't* do:

- Faster convergence rates for solvers.
- Uniform convergence with modified structure.

My take on unrolled algorithms

What unrolled algorithms *can* do:

- Improve constants in convergence rate.
  [Moreau and Bruna 2017]
- Learn to better optimize for a non-uniform input distribution.
  [Ablin et al. 2019]
- Make inverse problem solution differentiable.
  [Ablin et al. 2020; Mehmood and Ochs 2020]

What unrolled algorithms *can't* do:

- Faster convergence rates for solvers.
- Uniform convergence with modified structure.

$\Rightarrow$ Can we extend these results to other problems?

Conclusion

Take home messages:

First order structure is needed in optimization.
No hope to learn an algorithm better than ISTA.

(except for step-sizes!)

Unrolled algorithms are useful to learn to solve optimization problems in average.

(typical in bi-level optimization?)

Code to reproduce the figures is available online:

- **adopty** : github.com/tommoral/adopty
- **carpet** : github.com/hcherkaoui/carpet

Slides will be on my web page:

tommoral.github.io   @tomamoral

Interlude – regularization λ

Importance of the parameter λ

$$\mathcal{L}_x(z) = \frac{1}{2}\|x - Dz\|_2^2 + \lambda\|z\|_1$$

$$z^{(t+1)} = \text{ST}\left(z^{(t)} - \alpha^{(\text{t})} D^\top (Dz^{(t)} - x), \lambda\alpha^{(\text{t})}\right)$$

Control the distribution of $z^*(x)$ sparsity.

Maximal value

$\lambda_{\text{max}} = \|D^\top x\|_\infty$ is the minimal value of λ for which

$$z^*(x) = 0$$

Equiregularization set

Set in $\mathbb{R}^n$ for which $\lambda_{\text{max}} = 1$

$$\mathcal{B}_\infty = \{x \in \mathbb{R}^n \,;\; \|D^\top x\|_\infty = 1\}$$

$\Rightarrow$ Training performed with points sampled in $\mathcal{B}_\infty$

TV regularized problems

TV regularized problems

Given a forward operator $D \in \mathbb{R}^{n \times m}$ and $\lambda > 0$, the Lasso for $x \in \mathbb{R}^n$ is

$$z^* = \operatorname*{argmin}_z P_x(z) = \underbrace{\frac{1}{2}\|x - Dz\|_2^2}_{f_x(z)} + \lambda \|z\|_{TV}$$

where $\|z\|_{TV} = \|\nabla z\|_1$, and $\nabla = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 1 \end{bmatrix} \in \mathbb{R}^{k-1 \times k}$

Why consider this? equivalent formulation with Lasso:

$$\min_{u \in \mathbb{R}^k} S_x(u) = \frac{1}{2}\|x - DLu\|_2^2 + \lambda \|Ru\|_1 .$$

where R is diagonal and L is the discrete integration operator.

$\Rightarrow$

Why not just use the synthesis formulation? [?]

Convergence rate comparison

Both cvg rates are in $\mathcal{O}(1/t)$ but scale with $\rho = \|D\|_2^2$ or $\widetilde{\rho} = \|D\nabla\|_2^2$.

Theorem (Lower bound for the ratio $\frac{\widetilde{\rho}}{\rho}$ expectation)

Let D be a random matrix in $\mathbb{R}^{m\times k}$ with iid normal entries. The expectation of $\widetilde{\rho}/\rho$ is asymptotically lower bounded when k tends to ∞ by

$$\mathbb{E}\left[\frac{\widetilde{\rho}}{\rho}\right] \geq \frac{2k+1}{4\pi^2} + o(1)$$

Empirical evidences also push for a $\mathcal{O}(k^2)$ scaling.

Analysis is more efficient in terms of iterations than Synthesis.

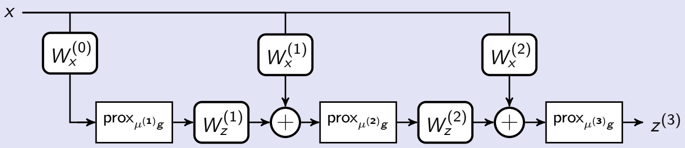


Figure: **LPGD** - Unfolded network for Learned PGD with $T = 3$

Main blocker:

How to compute $\mathrm{prox}_{\mu g}$ efficiently and in a differentiable way?

- Use dedicated solver and compute gradient with implicit function theorem.
- Use an unrolled algorithm (LISTA) to solve the prox.

Performance investigation

Very low dimensional simulation $k, m = 5, 8$ (because of memory issue).

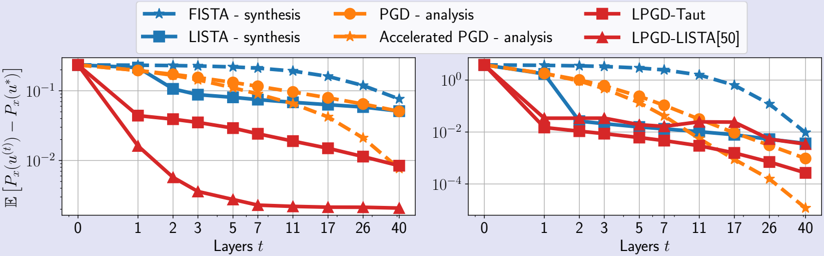


Figure: **Performance comparison** for different regularisation levels (*left*) $\lambda = 0.1$, (*right*) $\lambda = 0.8$.

fMRI data deconvolution (UKBB)

We retain only 8000 time-series of 250 time-frames (3 minute 03 seconds), Deconvolution for a fixed kernel h and estimate the neural activity signal z for each voxels.

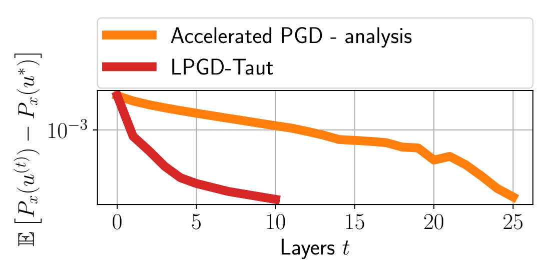


Figure: **Performance comparison** $\lambda = 0.1\lambda_{\text{max}}$ between LPGD-Taut and iterative PGD for the analysis formulation for the HRF deconvolution problem with fMRI data.

Weights coupling

We denote $\theta = (W, \alpha, \beta)$ the parameters of a given layer ϕ_θ.

$$\phi_\theta(z, x) = \text{ST}\left(z - \alpha D^\top (Dz - x), \lambda\alpha\right)$$

Assumption 1:
$D \in \mathbb{R}^{n \times m}$ is a dictionary with non-duplicated unit-normed columns.

Lemma 4.3 – Weight coupling

If for all the couples $(z^*(x), x) \in \mathbb{R}^m \times \mathcal{B}_\infty$ such that $z^*(x) \in \operatorname{argmin} F_x(z)$, it holds $\phi_\theta(z^*(x), x) = z^*(x)$. Then, $\frac{\alpha}{\beta} W = D$.

The solution of the Lasso is a fixed point of a given layer ϕ_θ if and only if ϕ_θ is equivalent to a step of ISTA with a given step-size.

ISTA – Convergence

Convergence rates

If f_x is μ-strongly convex, *i.e.* $\sigma_{\min}(D^T D) \geq \mu > 0$

$$F_x(z^{(t)}) - F_x(z^*) \leq \left(1 - \frac{\mu}{L}\right)^t (F_x(0) - F_x(z^*))$$

In the general case, $F_x(z^{(t)}) - F_x(z^*) \leq \frac{L\|z^*\|_2}{t}$

OISTA – Convergence

Proposition 3.1: Convergence

When D is such that the solution is unique for all x and $\lambda > 0$, the sequence $(z^{(t)})$ generated by the algorithm converges to $z^* = \operatorname{argmin} F_x$.
Further, there exists an iteration T^* such that for $t \geq T^*$, $\operatorname{Supp}(z^{(t)}) = \operatorname{Supp}(z^*) \triangleq S^*$.

Proposition 3.2: Convergence rate

For $t > T^*$,

$$F_x(z^{(t)}) - F_x(z^*) \leq L_{S^*} \frac{\|z^* - z^{(T^*)}\|^2}{2(t - T^*)} .$$

If moreover, $\lambda_{\min}(D_{S^*}^\top D_{S^*}) = \mu^* > 0$, then

$$F_x(z^{(t)}) - F_x(z^*) \leq (1 - \frac{\mu^*}{L_{S^*}})^{t - T^*}(F_x(z^{(T^*)}) - F_x(z^*)) .$$