Learning to solve TV regularised problems with unrolled algorithms

Hamza Cherkaoui, Jeremias Sulam, Thomas Moreau

CEA-Neurospin/SHFJ — INRIA-Saclay — Johns Hopkins University

hamza.cherkaoui@inria.fr

July 7, 2020

Overview

Motivation

Deconvolution problem in fMRI

The commun model for the BOLD signal (the fMRI data) is:

$$x = h * u + \epsilon \tag{1}$$

with x the BOLD signal, h the haemodynamic response function (HRF) and u the neural activity.

If we fix the HRF, we can recover the neural activation signal from the BOLD signal.

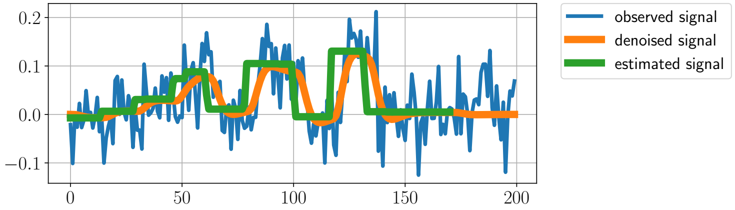

Figure: Deconvolution of the BOLD signal with a TV regularization.

Total Variation (TV) regularization

TV promotes piece-wise constant estimates by penalizing the ℓ_1-norm of the first order derivative of the estimated signal

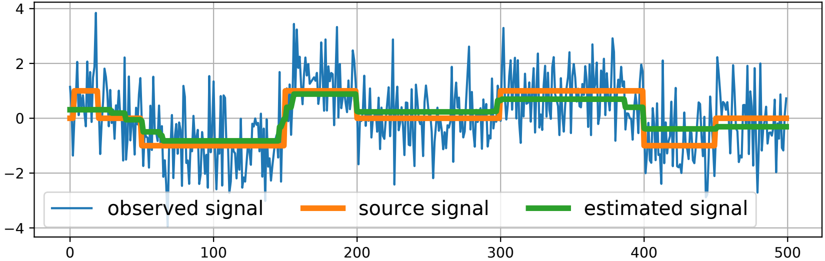

Figure: Signal denoising performed with a TV regularization.

Domain of application: machine learning, neuro-imaging, image restoration, etc

Formulation of the problem

Analysis formulation of the TV problem

Let $x \in \mathbb{R}^m$ the observed signal,
Let $\epsilon \in \mathbb{R}^m$ be an additive Gaussian noise,
Let $u \in \mathbb{R}^k$ the piece-wise constant signal,
Let $A \in \mathbb{R}^{m \times k}$ being some observation matrix,
Let $\lambda \in \mathbb{R}^+$ the regularization parameter.

$$x = Au + \epsilon \tag{2}$$

Primal analysis TV problem

$$\min_{u \in \mathbb{R}^k} P_x(u) = \frac{1}{2}\|x - Au\|_2^2 + \lambda \|u\|_{TV}, \tag{3}$$

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

Solving iteratively TV-regularized problems

Primal first order method approaches

$$u^{(t+1)} = \operatorname{prox}_{\frac{\lambda}{\rho}\|\cdot\|_{TV}}\left(u^{(t)} - \frac{1}{\rho}A^\top(Au^{(t)} - x)\right) \tag{4}$$

where $\rho = \|A\|_2^2$ and the prox-TV is defined as

$$\operatorname{prox}_{\mu\|\cdot\|_{TV}}(y) = \arg\min_{u\in\mathbb{R}^k} F_y(u) = \frac{1}{2}\|y - u\|_2^2 + \mu\|u\|_{TV}. \tag{5}$$

Dual first order method approaches

We can reformulate this analysis-primal problem to the dual:

Dual analysis TV problem

$$\min_{v \in \mathbb{R}^k} \frac{1}{2} \|{A^{\dagger}}^{\top} D^{\top} v\|_2^2 - v^{\top} D A^{\dagger} x \tag{6}$$

$$s.t. \quad \|v\|_{\infty} \leq \lambda \tag{7}$$

Solving iteratively TV-regularized problems

Dual first order method approaches

$$v^{(t+1)} = Proj_{\{\|v\|_\infty \leq \lambda\}} \left(v^{(t)} - \frac{1}{\rho} \Psi_A^\top (\Psi_A v^{(t)} - x) \right) \tag{8}$$

$$\tag{9}$$

With $\Psi_A = {A^\dagger}^\top D^\top$ and $\rho = \|\Psi_A\|_2^2$

Note: *alternatively, we can use a primal-dual descent algorithm (such as ADMM or the Vu-Condat splitting algorithm).*

Solving iteratively TV-regularized problems

Synthesis (equivalent) formulation of the TV problem

Let $z \in \mathbb{R}^k$ be the sparse source signal $s.t.\quad Lz = u$.

Primal synthesis TV problem

$$\min_{z \in \mathbb{R}^k} S_x(z) = \frac{1}{2}\|x - ALz\|_2^2 + \lambda\|Rz\|_1. \tag{10}$$

where $R = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{k \times k}$ and $L = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 1 & \dots & 1 & 1 \end{bmatrix} \in \mathbb{R}^{k \times k}$

We have $\forall (z, u) \in (\mathbb{R}^k, \mathbb{R}^k)$ $s.t.$ $u = Lz$, we have $S_x(z) = P_x(u)$.

Synthesis (equivalent) formulation of the TV problem

ISTA with a pseudo soft-thresholding operator *[Tibshirani, 1996]*

$$z^{(t+1)} = \mathrm{ST}\left(\left(z^{(t)} - \frac{1}{\rho} L^\top A^\top (ALz^{(t)} - x)\right), \frac{\lambda}{\rho}\right) \tag{11}$$

$$\tag{12}$$

with:

$$\mathrm{ST}(x) = \begin{cases} x_i, & \text{if } i = 1, \\ (|x_i| - \lambda)_+, & \text{otherwise.} \end{cases}$$

where

$$x_+ = \begin{cases} x, & \text{if } x > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Solving iteratively TV-regularized problems

Convergence rate comparison

Analysis formulation convergence rate

$$P(u^{(t)}) - P(u^*) \leq \frac{\rho}{2t}\|u^{(0)} - u^*\|_2^2, \tag{13}$$

Synthesis formulation convergence rate

$$P(u^{(t)}) - P(u^*) \leq \frac{2\widetilde{\rho}}{t}\|u^{(0)} - u^*\|_2^2, \tag{14}$$

Theorem (Lower bound for the ratio $\frac{\|AL\|_2^2}{\|A\|_2^2}$ expectation)

Let A be a random matrix in $\mathbb{R}^{m\times k}$ with i.i.d normal entries. The expectation of $\|AL\|_2^2/\|A\|_2^2$ is asymptotically lower bounded when k tends to ∞ by

$$\mathbb{E}\left[\frac{\|AL\|_2^2}{\|A\|_2^2}\right] \geq \frac{2k+1}{4\pi^2} + o(1)$$

Convergence rate comparison

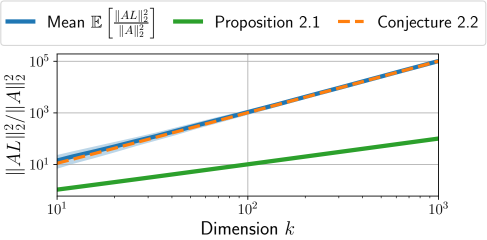

Figure: Evolution of $\mathbb{E}\left[\frac{\|AL\|_2^2}{\|A\|_2^2}\right]$ *w.r.t* the dimension k for random matrices A with *i.i.d* normal entries. In light blue is the confidence interval [0.1, 0.9] computed with the quantiles.

So, we can expect that $\widetilde{\rho}/\rho$ scales as $\Theta(k^2)$.
Which leads to $\frac{\widetilde{\rho}}{2} \gg \rho$ in large enough dimension.

The analysis formulation should be much more efficient in terms of iterations than the synthesis formulation.

Solving iteratively TV-regularized problems

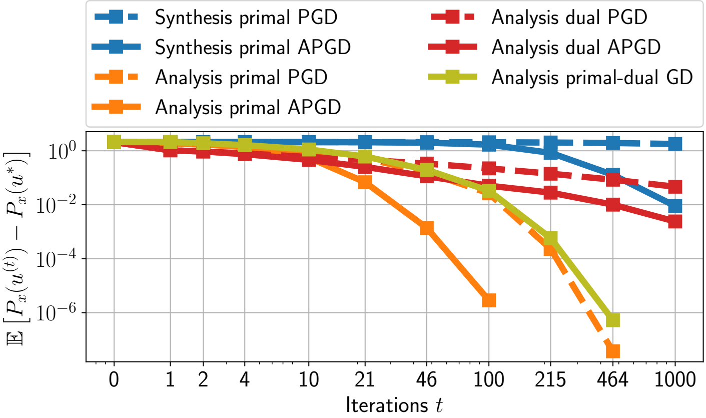

Figure: **Performance comparison** $\lambda = 0.1\lambda_{\text{max}}$ between the iterative solver for the synthesis and analysis formulation with the corresponding primal, dual or primal-dual re-parametrization.

Solving iteratively TV-regularized problems

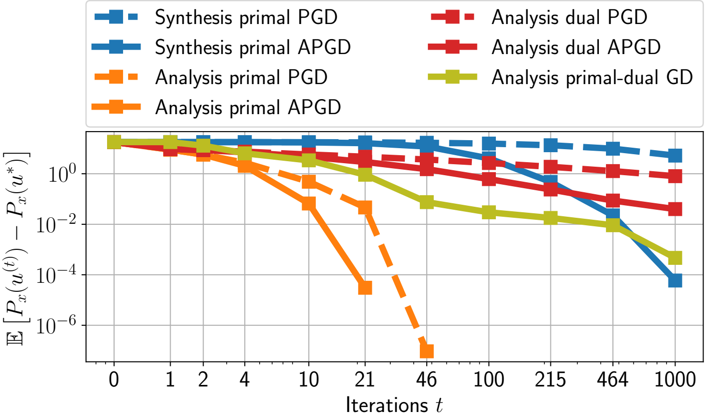

Figure: **Performance comparison** $\lambda = 0.8\lambda_{\max}$ between the iterative solver for the synthesis and analysis formulation with the corresponding primal, dual or primal-dual re-parametrization.

Principle of unrolling

Consider the following generic problem *[Gregor and Le Cun, 2010]*:

$$\arg\min_{u \in \mathbb{R}^k} \mathcal{L}(x, u) = \frac{1}{2}\|x - Bu\|_2^2 + \lambda g(u) \ , \tag{15}$$

If we defined:

$$W_x^{(t)} = \frac{1}{\rho} B^\top, \quad W_u^{(t)} = (\mathrm{Id} - \frac{1}{\rho} B^\top B) \ , \quad \mu^{(t)} = \frac{\lambda}{\rho}, \quad \text{with } \rho = \|B\|_2^2 \ . \tag{16}$$

The recursive equation to minimize Eq:15 reads:

$$u^{(0)} = B^\dagger x \ ; \qquad u^{(t)} = \mathrm{prox}_{\mu^{(t)} g}(W_x^{(t)} x + W_u^{(t)} u^{(t-1)}) \ . \tag{17}$$

Principle of unrolling

$$u^{(0)} = B^{\dagger} x \; ; \qquad u^{(t)} = \mathrm{prox}_{\mu^{(t)} g}\left(W_x^{(t)} x + W_u^{(t)} u^{(t-1)}\right) \; . \tag{18}$$

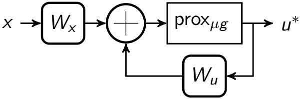

Figure: **PGD** - Recurrent Neural Network

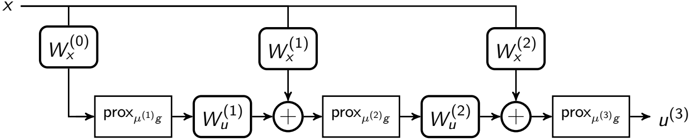

Figure: **LPGD** - Unfolded network for Learned PGD with $T = 3$

Neural network training

Let $\Theta^{(T)}$ be the weights of the T first layers of the neural network,
Let $\Phi_{\Theta^{(T)}}$ be the neural network defined with those weights,
Let $(x_i)_1^N$ be the training samples.

To train the neural network, we minimize:

$$\min_{\Theta^{(T)}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(x_i, \phi_{\Theta^{(T)}}(x_i)) \quad . \tag{19}$$

Derivative of prox-TV

Back-propagate through the prox-TV step

To learn the weights of the defined neural network, we need to back-propagate the error.

Let $h = W_x^{(t)} x + W_u^{(t)} \phi_{\Theta^{(t-1)}}(x)$ and $u = \text{prox}_{\mu^{(t)} \|\cdot\|_{TV}}(h)$

The chain rule gives use:

$$\frac{\partial \mathcal{L}}{\partial h} = J_x(h, \mu^{(t)})^\top \frac{\partial \mathcal{L}}{\partial u} \ , \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial \mu^{(t)}} = J_\mu(h, \mu^{(t)})^\top \frac{\partial \mathcal{L}}{\partial u} \ , \tag{20}$$

We need to compute $J_x(h, \mu) \in \mathbb{R}^{k \times k}$ and $J_\mu(h, \mu) \in \mathbb{R}^{k \times 1}$

Derivative of prox-TV

Theorem (Jacobian of prox-TV)

Let $x \in \mathbb{R}^k$ *and* $u = \text{prox}_{\mu\|\cdot\|_{TV}}(x)$, *and denote by* $\mathcal{S}$ *the support of* $z = \widetilde{D}u$. *Then, the Jacobian* J_x *and* J_μ *of the prox-TV relative to* x *and* μ *can be computed as*

$$J_x(x, \mu) = L_{:,\mathcal{S}}(L_{:,\mathcal{S}}^\top L_{:,\mathcal{S}})^{-1} L_{:,\mathcal{S}}^\top$$

and

$$J_\mu(x, \mu) = -L_{:,\mathcal{S}}(L_{:,\mathcal{S}}^\top L_{:,\mathcal{S}})^{-1} \operatorname{sign}(Du)_{\mathcal{S}}$$

Derivative of prox-TV

Remarks on the Jacobians J_x and J_μ

- They invoked a matrix inversion, which have a $\Theta(k^3)$ complexity
- Those inversions need to be computed at every iterations... but only for the training step!
- Those Jacobians are zero outside the support of z: the smaller the support of z the lesser we 'learn'

Process summary

- Forward pass: use the Taut-string algorithm ($\Theta(k)$ complexity in most cases).
- Back-propagation pass: use the automatic-differentiation along with the analytic formulas of J_x and J_μ.

Unrolled prox-TV

Similary, we can defined an inner neural network to solve:

$$z^* = \underset{z \in \mathbb{R}^k}{\arg\min} \frac{1}{2} \|h - Lz\|_2^2 + \mu \|Rz\|_1 \tag{21}$$

Process summary

- Forward pass: use the forward inner neural network.
- Back-propagation pass: use the automatic-differentiation through the inner neural network.

Performance investigation

We generate $n = 2000$ times series,
Such as $(u_i)_{i=1}^n \in \mathbb{R}^{n \times k}$ with $k = 8$
Each u_i has a support of $|S| = 2$ non-zero coefficients,
Let $A \in \mathbb{R}^{m \times k}$ as a Gaussian matrix with $m = 5$,
We add Gaussian noise to measurements $x_i = Au_i$ with a SNR of 1.0.

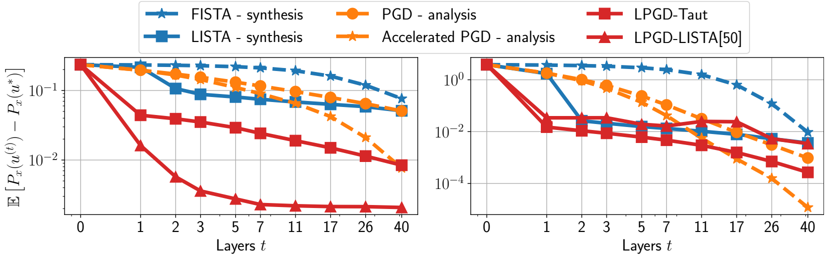

Figure: **Performance comparison** for different regularisation levels (*left*) $\lambda = 0.1$, (*right*) $\lambda = 0.8$.

Simulation

Inexact Prox-TV error investigation

(Same experimental configuration than previously).

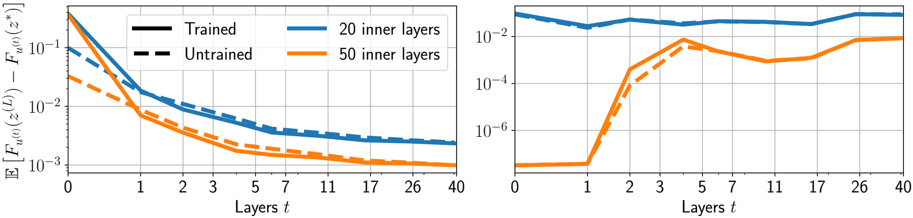

Figure: **Proximal operator error comparison** for different regularisation levels (*left*) $\lambda = 0.1$, (*right*) $\lambda = 0.8$.

fMRI data deconvolution

Performance investigation

We used UK Bio Bank (UKBB) dataset,
We retain only 8000 time-series of 250 time-frames (3 minute 03 seconds),
We fix the HRF h and estimate the neural activity signal u for each voxels.

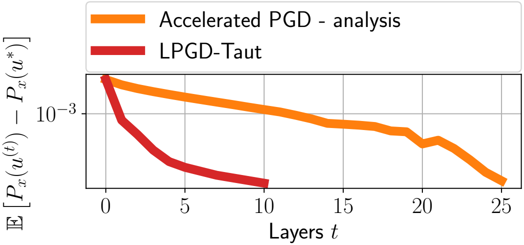

Figure: **Performance comparison** $\lambda = 0.1\lambda_{\text{max}}$ between LPGD-Taut and iterative PGD for the analysis formulation for the HRF deconvolution problem with fMRI data.

Conclusion

Take-home message:

- The analysis formulation can be solved more efficiently with PGD than the synthesis formulation
- Unrolling the algorithm in the analysis allows to learn more efficient algorithm than unrolling in the synthesis
- We have a control over the error in the case of the inexact proximal operator, but in practice the obtained T_{in} can be too 'high'.
- We will extend this work to the 2D case

Questions?