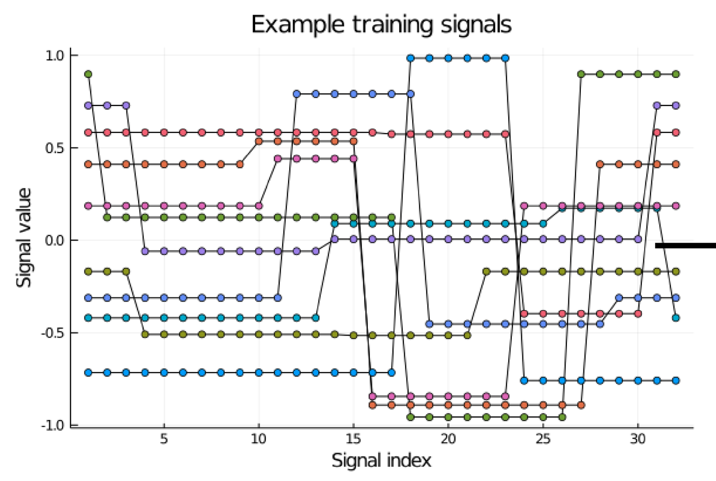
Motivating Bilevel Approaches to Filter Learning: A Case Study

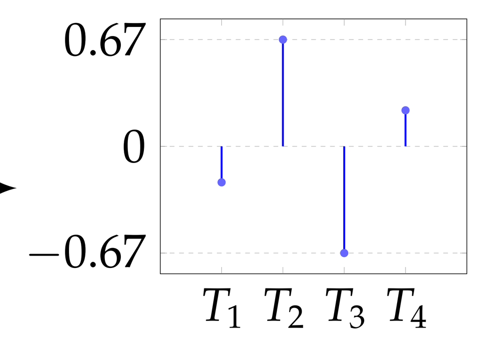
Caroline Crockett and Jeffrey A. Fessler
Department of EECS, University of Michigan, {cecroc fessler}@umich.edu

Motivation



Transform Learning:
Learn $\mathbf{T}$ to sparsify training signals

$$\operatorname*{argmin}_{\mathbf{T}\in\mathbb{T},\mathbf{z}_l} \sum_l \frac{1}{2}\|\mathbf{T}\mathbf{s}_l - \mathbf{z}_l\|_2^2 + \lambda\|\mathbf{z}_l\|_0$$



$$\hat{\mathbf{T}} = \left[-\sqrt{\tfrac{1-2d^2}{2}}\ \ d\ \ {-d}\ \ \sqrt{\tfrac{1-2d^2}{2}}\right]$$
$$d(\lambda{=}0.23) = 0.67$$

$$\mathbf{T}_{\text{TV}} = [0\ \text{-}0.707\ 0.707\ 0]$$

Why do we not learn $\mathbf{T}_{\text{TV}}$?

Conclusion

- The smoothness in $\hat{\mathbf{T}}$ results from splitting the objective function and introducing λ.
- The smoothness increases with λ.
- By construction, the learned transform will have a lower training objective value.
- However, $\mathbf{T}_{\text{TV}}$ denoises better than the smoothed $\hat{\mathbf{T}}$.
- The task-based nature of bilevel learning can reduce the smoothing effect.
- However, the bilevel learning method requires a differentiable objective function, yielding noisier images than $\mathbf{T}_{\text{TV}}$ with a 0-norm regularizer.
- The bilevel results can likely be improved with a non-convex regularizer.

Training

- Image denoising: $\hat{\mathbf{x}} = \operatorname{argmin}_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 + \beta R(\mathbf{x})$
- Goal: Learn $R(\mathbf{x})$ from training data to denoise images

Training signals

- Noiseless PWC 1D signals
  - 1,024 for transform learning: patches of these are $\mathbf{s}_l$
  - 128 for bilevel filter learning: each signal is $\mathbf{s}_j$
- At most one jump in any given length-4 patch
- $\mathbb{T}$ as the set of single, length-4 filters with unit norm

Training results

- When $R(\mathbf{x}) = \|\mathbf{T}\mathbf{x}\|_0$, the best transform/filter is:

$$\mathbf{T}_{\text{TV}} = \mathbf{h}_{\text{TV}} = \frac{1}{\sqrt{2}}[0\ 1\ \text{-}1\ 0]$$

- Compare learned Transforms and filters to $\mathbf{T}_{\text{TV}}$ using the angle between vectors: $\cos^{-1}(|\langle \mathbf{z}_1, \mathbf{z}_2\rangle| / \|\mathbf{z}_1\|\,\|\mathbf{z}_2\|)$.

Transform Learning [1]

$$\hat{\mathbf{T}} = \operatorname*{argmin}_{\underset{\uparrow\ \text{Orthonormal rows}}{\mathbf{T}\in\mathbb{T}}} \min_{\mathbf{z}_l\in\mathbb{C}^K} \sum_{l=1}^{L} \frac{1}{2}\|\mathbf{T}\mathbf{s}_l - \mathbf{z}_l\|_2^2 + \underbrace{\lambda}_{\text{Tuning parameter}}\|\mathbf{z}_l\|_0$$

"Dummy rows"

$$\mathbf{W} = [\boldsymbol{I}_K\ \mathbf{0}]$$

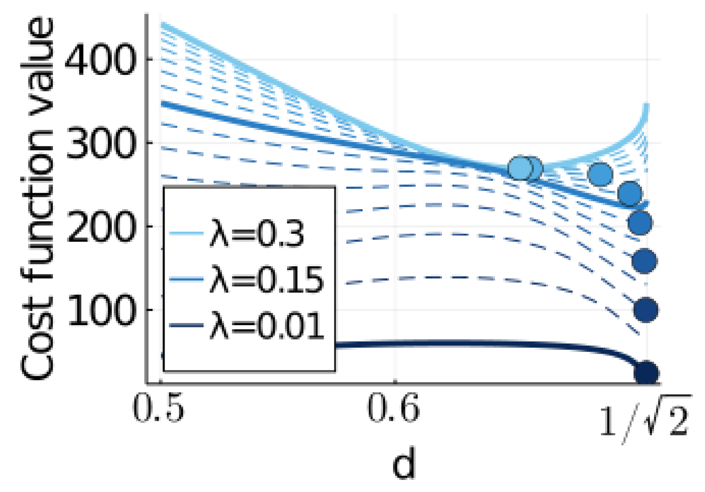
$$\hat{\tilde{\mathbf{T}}} = \operatorname*{argmin}_{\tilde{\mathbf{T}}\in\tilde{\mathbb{T}}} \min_{\mathbf{z}_l\in\mathbb{C}^D} \sum_{l=1}^{L} \frac{1}{2}\|\tilde{\mathbf{T}}\mathbf{s}_l - \mathbf{z}_l\|_2^2 + \lambda\|\mathbf{W}\mathbf{z}_l\|_0$$

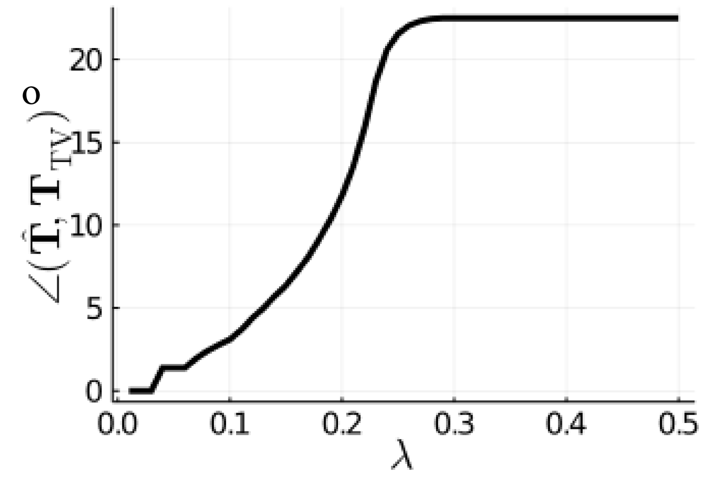
Block coordinate minimization

$$\mathbf{Z} = \tilde{\mathbf{T}}^{(n-1)}\mathbf{S}$$
$$\mathbf{Z}_{1:K,:} = \operatorname{prox.}(\mathbf{Z}_{1:K,:})$$

and

$$\hat{\tilde{\mathbf{T}}}^{(n)} = \underbrace{\operatorname*{argmin}_{\tilde{\mathbf{T}}\in\mathbb{T}} \|\tilde{\mathbf{T}}\mathbf{S} - \mathbf{Z}^{(n)}\|_{\text{F}}^2}_{\text{Procrustes problem [2]}}$$





Bilevel Method: Convolutional Filters [3]–[5]

Upper-level loss

$$\hat{\gamma} = \operatorname*{argmin}_{\gamma} \sum_{j=1}^{J} \frac{1}{2}\|\hat{\mathbf{x}}_j(\gamma) - \mathbf{s}_j\|_2^2 \text{ where}$$

$$\gamma = [\beta, \mathbf{h}]$$

$$\mathbf{y}_j = \mathbf{s}_j + \mathbf{n}$$

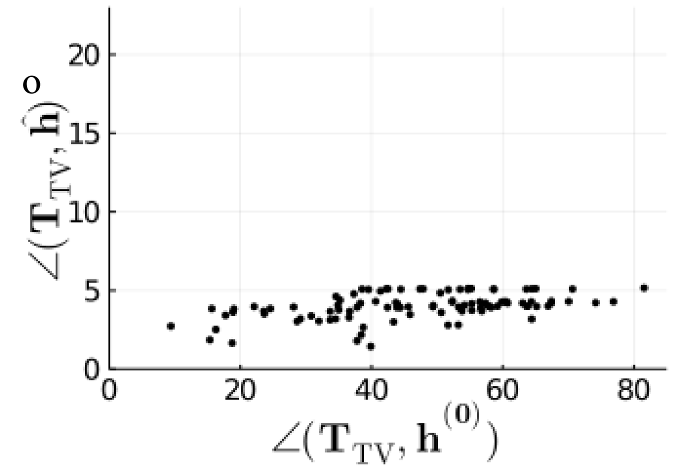
Lower-level task

$$\hat{\mathbf{x}}_j(\gamma) = \operatorname*{argmin}_{\mathbf{x}} \frac{1}{2}\left\|\mathbf{x} - \mathbf{y}_j\right\|_2^2 + \underbrace{\sum_{k=1}^{K} e^{\beta_k}\mathbf{1}'\psi.(\mathbf{h}_k \circledast \mathbf{x})}_{R(\mathbf{x};\gamma)}$$

$$\psi(z) = \sqrt{z^2 + 0.1^2}$$

Unrolled algorithm: $\mathbf{x}_j^{(i+1)} = \mathbf{x}_j^{(i)} - \frac{1}{L}\left(\mathbf{x}_j^{(i)} - \mathbf{y}_j + \sum_k e^{\beta_k}(\tilde{\mathbf{h}}_k \circledast \psi.(\mathbf{h}_k \circledast \mathbf{x}_j^{(i)}))\right)$

- Unroll enough lower-level gradient descent iterations to reach convergence [6], [7]
- Use Adam [8] on unrolled algorithm to learn γ
- Test 100 random initializations for $\mathbf{h}$
- All learned filters within 1.44 to 5.16 degrees of $\mathbf{h}_{\text{TV}}$



Denoising

Testing signals

- $\mathbf{s}_1$: length-1000 signal with 50 jumps (a slight generalization of our training data),
- $\mathbf{s}_2$: collection of 128 signals created in the same way as the training data but with a different random seed.
- Noisy data: the true signal plus mean zero Gaussian noise with a standard deviation of 0.1

Testing results

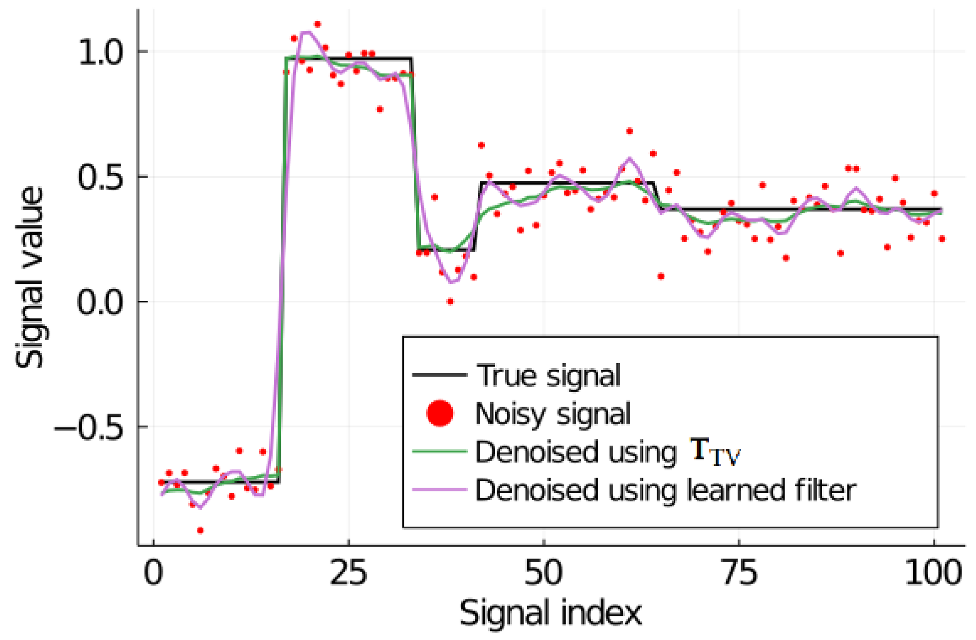
- Report the average root mean square error (RMSE):
  - $\sqrt{\frac{1}{N}\|\hat{\mathbf{x}} - \mathbf{s}\|^2}$
  - N is the signal length

$$\hat{\mathbf{x}}(\mathbf{y}) = \operatorname*{argmin}_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 + \beta\underbrace{\sum_l \min_{\mathbf{z}_l}\|\mathbf{T}\mathbf{P}_l\mathbf{x} - \mathbf{z}_l\|_2^2 + \alpha\|\mathbf{z}_l\|_0}_{R(\mathbf{x})}$$

Grid search

- The (smoothed) learned transform denoises worse than $\mathbf{T}_{\text{TV}}$.
- One could do a grid search over λ, but that would not be practical for many real-world datasets.

	$\mathbf{T}_{\text{TV}}$	$\hat{\mathbf{T}}(\lambda{=}0.23)$
$\mathbf{s}_1 \in \mathbb{R}^{1000}$	4.0	6.2
$\mathbf{s}_2 \in \mathbb{R}^{32}$	5.2	8.2



$$\hat{\mathbf{x}}(\gamma) = \operatorname*{argmin}_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 + \underbrace{e^{\hat{\beta}_1}\mathbf{1}'\psi.(\hat{\mathbf{h}}_1 \circledast \mathbf{x})}_{R(\mathbf{x};\gamma)}$$

Results for best and worse RMSE across random initializations.

	$\mathbf{h}_{\text{TV}}$	$\hat{\mathbf{h}}_{\text{best}}$	$\hat{\mathbf{h}}_{\text{worst}}$
$\mathbf{s}_1 \in \mathbb{R}^{1000}$	4.4	5.1	6.3
$\mathbf{s}_2 \in \mathbb{R}^{32}$	5.4	5.5	6.6

- No separate grid search needed.
- Learned filters denoise better than $\hat{\mathbf{T}}(\lambda = 0.23)$
- Learned filters are especially good for $\mathbf{s}_2$, which mimics the training data
- $\mathbf{T}_{\text{TV}}$ with the zero-norm outperforms learned filters with corner rounded 1-norm

[1] S. Ravishankar and Y. Bresler, "Learning Sparsifying Transforms," *IEEE Trans. on Signal Process.*, vol. 61, no. 5, pp. 1072–1086, 2013. DOI: 10.1109/TSP.2012.2226449.

[2] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966. DOI: 10.1007/BF02289451.

[3] M. T. McCann and S. Ravishankar, *Supervised Learning of Sparsity-Promoting Regularizers for Denoising*, 2020. arXiv: 2006.05521.

[4] G. Peyré and J. M. Fadili, *Learning Analysis Sparsity Priors*, Singapour, Singapore, 2011. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00542016/document.

[5] Y. Chen, T. Pock, and H. Bischof, "Learning l_1-based analysis and synthesis sparsity priors using bi-level optimization," in *Neural Information Processing Systems Confercence (NIPS)*, 2014. arXiv: 1401.4105 [cs].

[6] L. Franceschi, M. Donini, P. Frasconi, and M. Pontil, "Forward and Reverse Gradient-Based Hyperparameter Optimization," in *Proceedings of the 34th ICML*, Sydney, Australia, 2017, pp. 1165–1173. [Online]. Available: http://proceedings.mlr.press/v70/franceschi17a.html.

[7] H. Antil, Z. Di, and R. Khatri, "Bilevel Optimization, Deep Learning and Fractional Laplacian Regularization with Applications in Tomography," *Inverse Problems*, Mar. 18, 2020, ISSN: 0266-5611, 1361-6420. DOI: 10.1088/1361-6420/ab80d7.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015. arXiv: 1412.6980.

This work was supported in part by NIH grants R01 EB023618 and NSF grant IIS 1838179.