Bilevel Learning for Inverse Problems

Matthias J. Ehrhardt

Institute for Mathematical Innovation, University of Bath, UK

February 12, 2021

Joint work with:
F. Sherry, M. Graves, G. Maierhofer, G. Williams, C.-B. Schönlieb (all Cambridge, UK), M. Benning (Queen Mary, UK), J.C. De los Reyes (EPN, Ecuador)

L. Roberts (ANU, Australia)

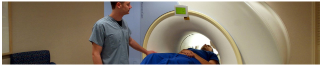
The Leverhulme Trust

Engineering and Physical Sciences Research Council

THE FARADAY INSTITUTION

Outline

1) Motivation



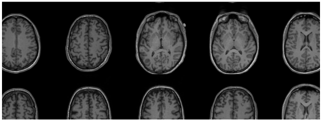
$$\min_x \tfrac{1}{2}\|SFx - y\|_2^2 + \lambda\mathcal{R}(x)$$

2) Bilevel Learning

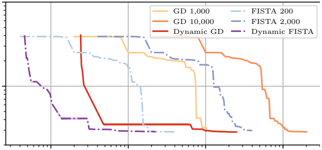
$$\min_{x,y} f(x, y)$$

$$x \in \arg\min_z g(z, y)$$

3) Learn sampling pattern in MRI
F. Sherry et al., "Learning the Sampling Pattern for MRI," IEEE TMI 2020.



4) Inexact algorithms for bilevel learning M. J. Ehrhardt and L. Roberts, "Inexact Derivative-Free Optimization for Bilevel Learning," Accept. by JMIV 2020.



Inverse problems

$$Ax = y$$

x : desired solution

y : observed data

A : mathematical model

Goal: recover x given y

Hadamard (1902): We call an inverse problem $Ax = y$ **well-posed** if

(1) a solution x^* **exists**

(2) the solution x^* is **unique**

(3) x^* depends **continuously** on data y.

Otherwise, it is called **ill-posed**.

Jacques Hadamard

Most interesting problems are **ill-posed**.

How to solve inverse problems?

Variational regularization ($\sim$1990)
Approximate a solution x^* of $Ax = y$ via

$$\hat{x} \in \arg\min_x \left\{ \mathcal{D}(Ax, y) + \lambda \mathcal{R}(x) \right\}$$

- $\mathcal{R}$ **regularizer**: penalizes unwanted features, ensures stability and uniqueness
- λ **regularization parameter**: $\lambda \geq 0$. If $\lambda = 0$, then an original solution is recovered. If $\lambda \to \infty$, more and more weight is given to the regularizer $\mathcal{R}$.

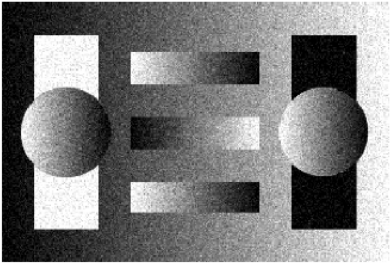
textbooks: Scherzer et al. 2008, Ito and Jin 2015, Benning and Burger 2018

Example: Regularizers

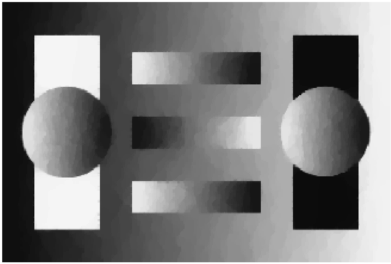
- Tikhonov regularization (∼1960): $\mathcal{R}(x) = \frac{1}{2}\|x\|_2^2$
- H^1 (∼1960-1990?) $\mathcal{R}(x) = \frac{1}{2}\|\nabla x\|_2^2$

Example: Regularizers

- Tikhonov regularization (∼1960): $\mathcal{R}(x) = \frac{1}{2}\|x\|_2^2$
- H^1 (∼1960-1990?) $\mathcal{R}(x) = \frac{1}{2}\|\nabla x\|_2^2$
- Total Variation $\mathcal{R}(x) = \|\nabla x\|_1$ Rudin, Osher, Fatemi 1992



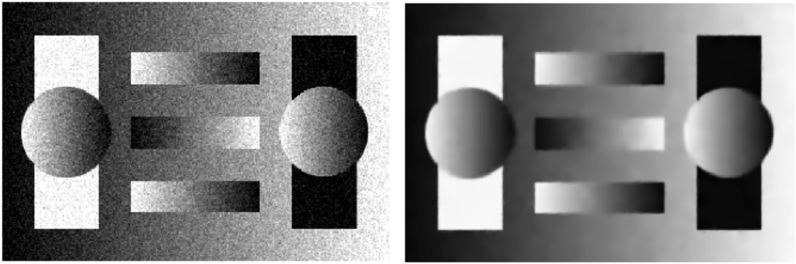
Noisy image



TV denoised image

Example: Regularizers

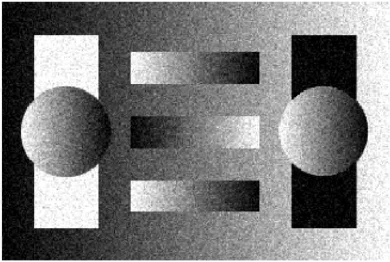
- Tikhonov regularization (∼1960): $\mathcal{R}(x) = \frac{1}{2}\|x\|_2^2$
- H^1 (∼1960-1990?) $\mathcal{R}(x) = \frac{1}{2}\|\nabla x\|_2^2$
- Total Variation $\mathcal{R}(x) = \|\nabla x\|_1$ Rudin, Osher, Fatemi 1992
- "Higher Order" Total Variation $\mathcal{R}(x) = \|\nabla^2 x\|_1$?



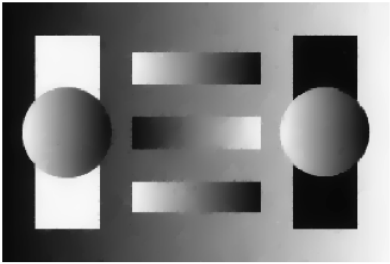
Noisy image　　TV² denoised image

Example: Regularizers

- Tikhonov regularization (∼1960): $\mathcal{R}(x) = \frac{1}{2}\|x\|_2^2$
- H^1 (∼1960-1990?) $\mathcal{R}(x) = \frac{1}{2}\|\nabla x\|_2^2$
- Total Variation $\mathcal{R}(x) = \|\nabla x\|_1$ Rudin, Osher, Fatemi 1992
- "Higher Order" Total Variation $\mathcal{R}(x) = \|\nabla^2 x\|_1$?
- Total Generalized Variation
  $\mathcal{R}(x) = \inf_v \|\nabla x - v\|_1 + \beta\|\nabla v\|_1$ Bredies, Kunisch, Pock 2010



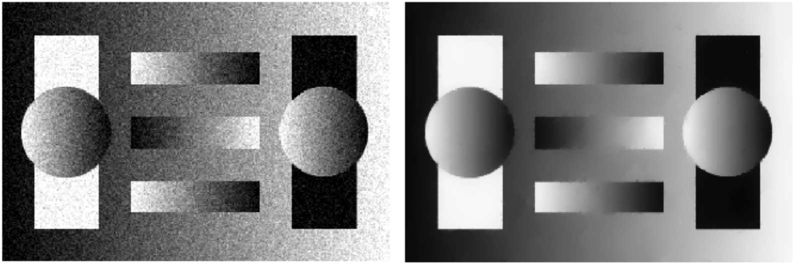
Noisy image



TGV^2 denoised image

Example: Regularizers

- Tikhonov regularization ($\sim$1960): $\mathcal{R}(x) = \frac{1}{2}\|x\|_2^2$
- H^1 ($\sim$1960-1990?) $\mathcal{R}(x) = \frac{1}{2}\|\nabla x\|_2^2$
- Total Variation $\mathcal{R}(x) = \|\nabla x\|_1$ Rudin, Osher, Fatemi 1992
- "Higher Order" Total Variation $\mathcal{R}(x) = \|\nabla^2 x\|_1$?
- Total Generalized Variation
  $\mathcal{R}(x) = \inf_v \|\nabla x - v\|_1 + \beta\|\nabla v\|_1$ Bredies, Kunisch, Pock 2010

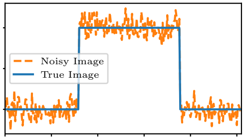


Noisy image — TGV² denoised image

How to choose the regularization?

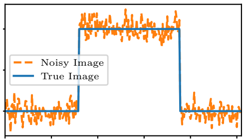
More "complicated" regularizers

$$\min_x \ \frac{1}{2}\|Ax-y\|_2^2 \ + \ \alpha \left(\underbrace{\sum_j \|(\nabla x)_j\|_2}_{=\mathrm{TV}(x)} \qquad\qquad\qquad \right)$$



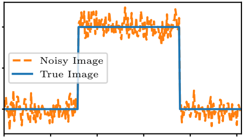
More "complicated" regularizers

$$\min_x \frac{1}{2}\|Ax-y\|_2^2+\alpha\left(\underbrace{\sum_j \sqrt{\|(\nabla x)_j\|_2^2+\nu^2}}_{\approx \mathrm{TV}(x)}\qquad\qquad\right)$$



More "complicated" regularizers

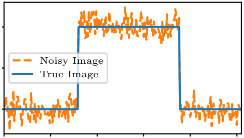
$$\min_x \frac{1}{2}\|Ax-y\|_2^2+\alpha\left(\underbrace{\sum_j \sqrt{\|(\nabla x)_j\|_2^2+\nu^2}}_{\approx \mathrm{TV}(x)}+\frac{\xi}{2}\|x\|_2^2\right)$$



- Smooth and strongly convex
- Solution depends on choices of α, ν and ξ

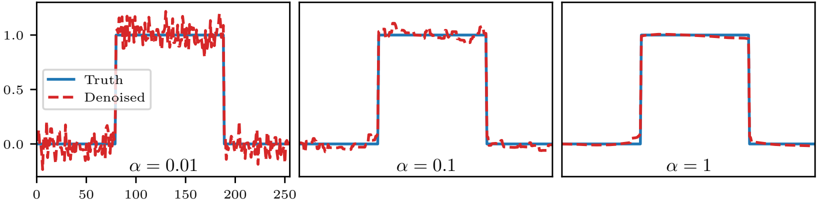
More "complicated" regularizers

$$\min_x \frac{1}{2}\|Ax-y\|_2^2+\alpha\left(\underbrace{\sum_j \sqrt{\|(\nabla x)_j\|_2^2+\nu^2}}_{\approx \mathrm{TV}(x)}+\frac{\xi}{2}\|x\|_2^2\right)$$



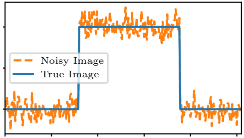
- Smooth and strongly convex
- Solution depends on choices of α, ν and ξ

Vary α ($\nu = 10^{-3}$, $\xi = 10^{-3}$)



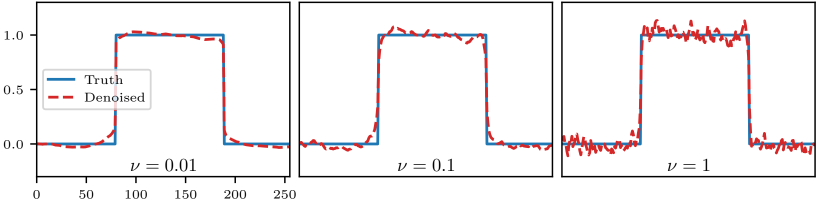
More "complicated" regularizers

$$\min_x \frac{1}{2}\|Ax-y\|_2^2+\alpha\left(\underbrace{\sum_j \sqrt{\|(\nabla x)_j\|_2^2+\nu^2}}_{\approx\mathrm{TV}(x)}+\frac{\xi}{2}\|x\|_2^2\right)$$



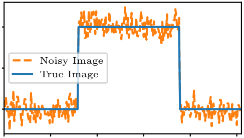
- Smooth and strongly convex
- Solution depends on choices of α, ν and ξ

Vary ν ($\alpha = 1$, $\xi = 10^{-3}$)



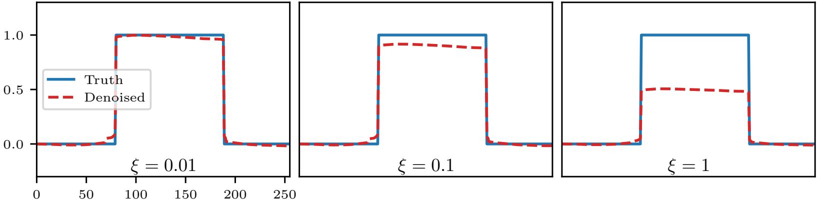
More "complicated" regularizers

$$\min_x \frac{1}{2}\|Ax-y\|_2^2+\alpha\left(\underbrace{\sum_j \sqrt{\|(\nabla x)_j\|_2^2+\nu^2}}_{\approx \mathrm{TV}(x)}+\frac{\xi}{2}\|x\|_2^2\right)$$



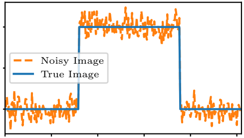
- Smooth and strongly convex
- Solution depends on choices of α, ν and ξ

Vary ξ $(\alpha = 1,\ \nu = 10^{-3})$



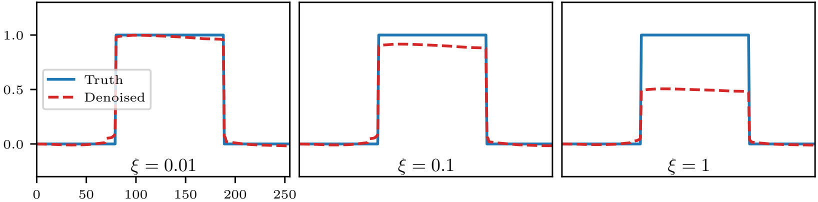
More "complicated" regularizers

$$\min_x \frac{1}{2}\|Ax-y\|_2^2+\alpha\left(\underbrace{\sum_j \sqrt{\|(\nabla x)_j\|_2^2+\nu^2}}_{\approx \mathrm{TV}(x)}+\frac{\xi}{2}\|x\|_2^2\right)$$



- Smooth and strongly convex
- Solution depends on choices of α, ν and ξ

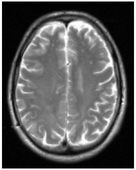
Vary ξ $(\alpha = 1,\ \nu = 10^{-3})$



How to choose all these parameters?

Example: Magnetic Resonance Imaging (MRI)

MRI scanner

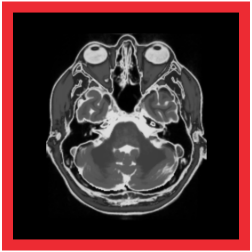


T_2^*

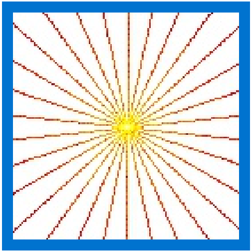
Continuous model: Fourier transform

$$Ax(s) = \int_{\mathbb{R}^2} x(s) \exp(-ist) dt$$

Dicrete model: $A = SF \in \mathbb{C}^{n \times N}$



$\rightarrow$



Solution **not unique**.

Example: MRI reconstruction

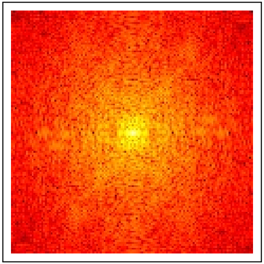
Compressed Sensing MRI:

$A = S \circ F$ Lustig, Donoho, Pauly 2007

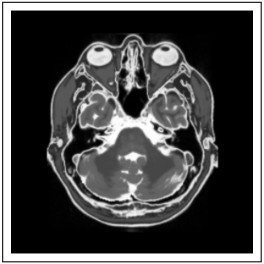
Fourier transform F, sampling $Sw = (w_i)_{i \in \Omega}$

$$\hat{x} \in \arg\min_{x} \left\{ \frac{1}{2} \|SFx - y\|_2^2 + \lambda \|\nabla x\|_1 \right\}$$

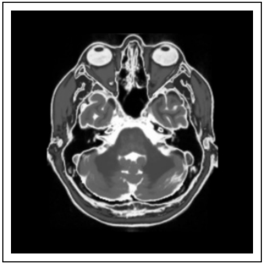
Miki Lustig



sampling $S^* y$



$\lambda = 0$



$\lambda = 1$

Example: MRI reconstruction

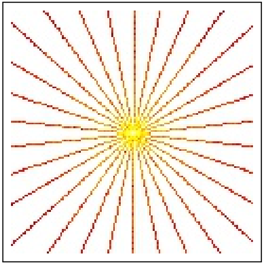
Compressed Sensing MRI:

$A = S \circ F$ Lustig, Donoho, Pauly 2007

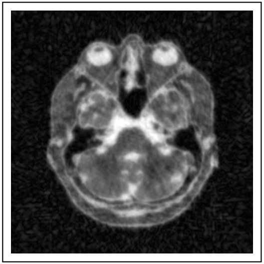
Fourier transform F, sampling $Sw = (w_i)_{i \in \Omega}$

$$\hat{x} \in \arg\min_x \left\{ \frac{1}{2} \|SFx - y\|_2^2 + \lambda \|\nabla x\|_1 \right\}$$

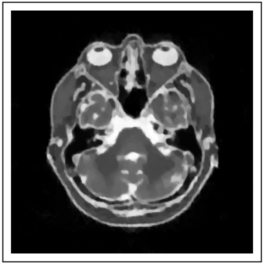
Miki Lustig



sampling $S^* y$



$\lambda = 0$



$\lambda = 10^{-4}$

Example: MRI reconstruction

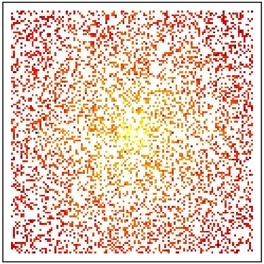
Compressed Sensing MRI:

$A = S \circ F$ Lustig, Donoho, Pauly 2007

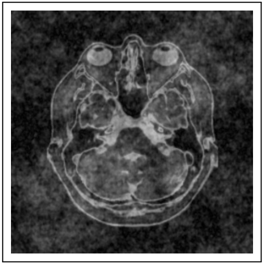
Fourier transform F, sampling $Sw = (w_i)_{i \in \Omega}$

$$\hat{x} \in \arg\min_x \left\{ \frac{1}{2} \|SFx - y\|_2^2 + \lambda \|\nabla x\|_1 \right\}$$

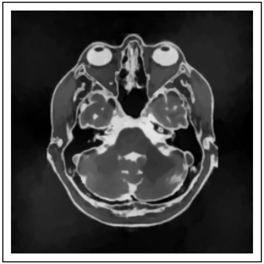
Miki Lustig



sampling $S^* y$



$\lambda = 0$



$\lambda = 10^{-4}$

Example: MRI reconstruction

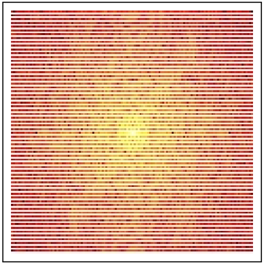
Compressed Sensing MRI:

$A = S \circ F$ Lustig, Donoho, Pauly 2007

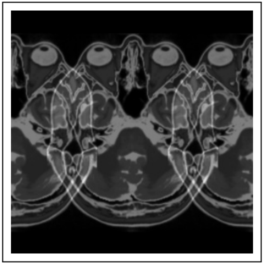
Fourier transform F, sampling $Sw = (w_i)_{i \in \Omega}$

$$\hat{x} \in \arg\min_x \left\{ \frac{1}{2} \|SFx - y\|_2^2 + \lambda \|\nabla x\|_1 \right\}$$

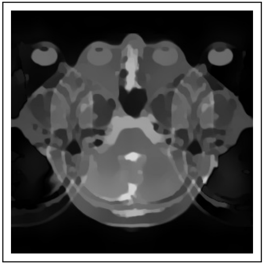
Miki Lustig



sampling $S^* y$



$\lambda = 0$



$\lambda = 10^{-3}$

How to choose the sampling S? Is there an optimal sampling?

Does a good sampling depend on $\mathcal{R}$ and λ?

Bilevel Learning

Bilevel learning for inverse problems

$$\hat{x} \in \arg\min_{x} \{\mathcal{D}(Ax, y) + \lambda \mathcal{R}(x)\}$$

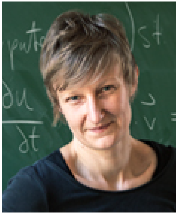
Bilevel learning for inverse problems

Upper level (learning):
Given $(x^\dagger, y), y = Ax^\dagger + \varepsilon$, solve

$$\min_{\lambda \geq 0, \hat{x}} \|\hat{x} - x^\dagger\|_2^2$$

Lower level (solve inverse problem):

$$\hat{x} \in \arg\min_x \{\mathcal{D}(Ax, y) + \lambda \mathcal{R}(x)\}$$



Carola Schönlieb

von Stackelberg 1934, Kunisch and Pock 2013, De los Reyes and Schönlieb 2013

Bilevel learning for inverse problems

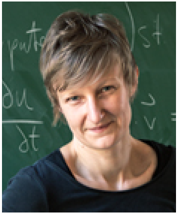
Upper level (learning):

Given $(x_i^\dagger, y_i)_{i=1}^n, y_i = Ax_i^\dagger + \varepsilon_i$, solve

$$\min_{\lambda \geq 0, \hat{x}_i} \frac{1}{n} \sum_{i=1}^n \|\hat{x}_i - x_i^\dagger\|_2^2$$

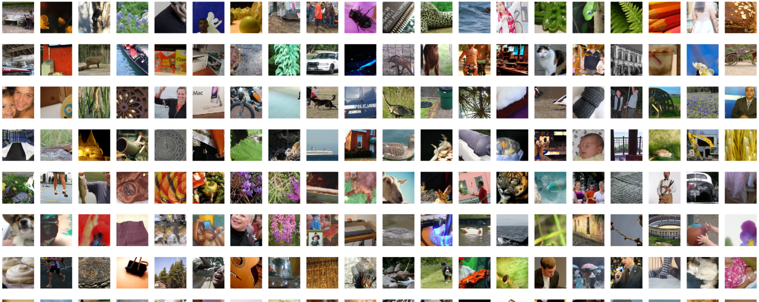
Lower level (solve inverse problem):

$$\hat{x}_i \in \arg\min_x \{\mathcal{D}(Ax, y_i) + \lambda \mathcal{R}(x)\}$$



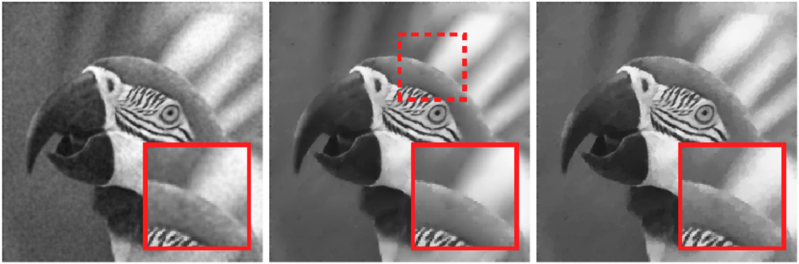
Carola Schönlieb

von Stackelberg 1934, Kunisch and Pock 2013, De los Reyes and Schönlieb 2013



Denoising: Learning two TGV parameters.

$$\mathcal{R}(x) = \inf_v \|\nabla x - v\|_1 + \beta \|\nabla v\|_1$$



(a) Too low β / High oscillation

(b) Optimal β

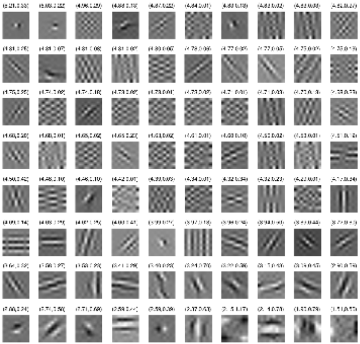
(c) Too high β / almost TV

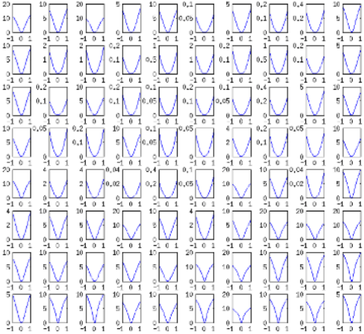
De los Reyes, Schönlieb, Valkonen 2017

Denoising: fields of experts regularisation

Learning filters K_k and potential functions ρ_k for fields of experts regularisation

$$\mathcal{R}(x) = \sum_{k=1}^{M} \sum_{i,j} \rho_k((K_k x)_{i,j})$$





Chen, Ranftl, Pock 2014

Learn sampling pattern in MRI

Some important works on sampling for MRI

Uninformed

- Cartesian, radial, variable density ... e.g. Lustig et al. 2007
  - ✓ simple to implement
  - ✗ not tailored to application or reconstruction method
- compressed sensing: random sampling e.g. Candes and Romberg 2007
  - ✓ mathematical guarantees
  - ✗ limited to sparse signals and sparsity promoting regularizers

Some important works on sampling for MRI

Uninformed

- Cartesian, radial, variable density ... e.g. Lustig et al. 2007
  - ✓ simple to implement
  - ✗ not tailored to application or reconstruction method
- compressed sensing: random sampling e.g. Candes and Romberg 2007
  - ✓ mathematical guarantees
  - ✗ limited to sparse signals and sparsity promoting regularizers

Learned

- **Largest Fourier coefficients** of training set Knoll et al. 2011
  - ✓ simple to implement, computationally light
  - ✗ not tailored to reconstruction method
- **greedy**: iteratively select "best" sample e.g. Gözcü et al. 2018
  - ✓ adaptive to dataset, reconstruction method
  - ✗ only discrete values; computationally heavy
- **Deep learning**: e.g. specify sampling as continuous parameters in network Wang et al. 2021
  - ✓ realistic and easy to implement sampling patterns
  - ✓ end-to-end
  - ✗ limited to neural network reconstruction

Learn sampling pattern in MRI

Lower level (MRI reconstruction):

$$R(\lambda, s, y) = \arg\min_x \left\{ \frac{1}{2}\|S(Fx - y)\|_2^2 + \lambda \mathcal{R}(x) \right\}$$

$$S = \mathrm{diag}(s), \quad s_i \in \{0, 1\}$$

Sherry et al. 2020

Learn sampling pattern in MRI

Upper level (learning):

Given **training data** $(x_i^\dagger, y_i)_{i=1}^n$, solve

$$\min_{\lambda \geq 0, s \in \{0,1\}^m} \frac{1}{n} \sum_{i=1}^{n} \|R(\lambda, s, y_i) - x_i^\dagger\|_2^2$$

Lower level (MRI reconstruction):

$$R(\lambda, s, y) = \arg\min_x \left\{ \frac{1}{2} \|S(Fx - y)\|_2^2 + \lambda \mathcal{R}(x) \right\}$$

$$S = \operatorname{diag}(s), \quad s_i \in \{0, 1\}$$

Sherry et al. 2020

Learn sampling pattern in MRI

Upper level (learning):
Given **training data** $(x_i^\dagger, y_i)_{i=1}^n$, solve

$$\min_{\lambda \geq 0, s \in [0,1]^m} \frac{1}{n} \sum_{i=1}^n \| R(\lambda, s, y_i) - x_i^\dagger \|_2^2$$

Lower level (MRI reconstruction):

$$R(\lambda, s, y) = \arg\min_x \left\{ \frac{1}{2} \| S(Fx - y) \|_2^2 + \lambda \mathcal{R}(x) \right\}$$

$$S = \mathrm{diag}(s), \quad s_i \in [0, 1]$$

Sherry et al. 2020

Learn sampling pattern in MRI

Upper level (learning):

Given **training data** $(x_i^\dagger, y_i)_{i=1}^n$, solve

$$\min_{\lambda \geq 0, s \in [0,1]^m} \frac{1}{n} \sum_{i=1}^n \|R(\lambda, s, y_i) - x_i^\dagger\|_2^2 + \beta_1 \|s\|_1 + \beta_2 \|s(1-s)\|_1$$

Lower level (MRI reconstruction):

$$R(\lambda, s, y) = \arg\min_x \left\{ \frac{1}{2} \|S(Fx - y)\|_2^2 + \lambda \mathcal{R}(x) \right\}$$

$$S = \mathrm{diag}(s), \quad s_i \in [0, 1]$$

Sherry et al. 2020

Warm up

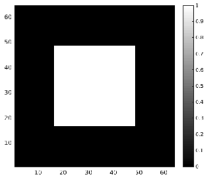
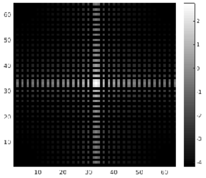
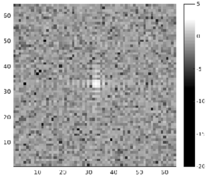


Figure: Discrete 2d bump



(a) Original data: $\log |y|$



(b) Noisy data: $\log |\tilde{y}|$

Warm up

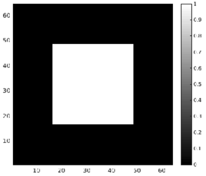
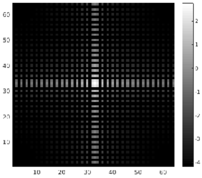
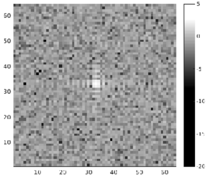


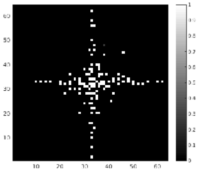
Figure: Discrete 2d bump



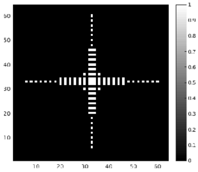
(a) Original data: $\log |y|$



(b) Noisy data: $\log |\tilde{y}|$



(c) Learned sampling pattern



(d) Largest 2.76% Fourier Coefficients

Warm up

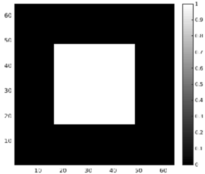
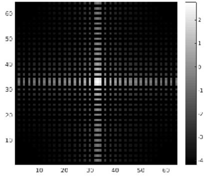
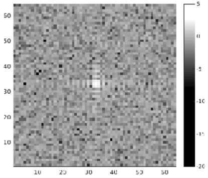


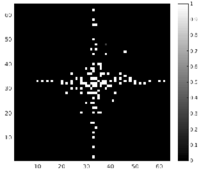
Figure: Discrete 2d bump



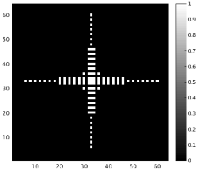
(a) Original data: $\log |y|$



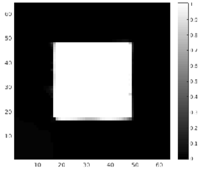
(b) Noisy data: $\log |\tilde{y}|$



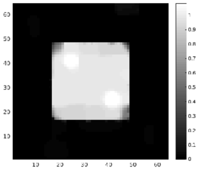
(c) Learned sampling pattern



(d) Largest 2.76% Fourier Coefficients

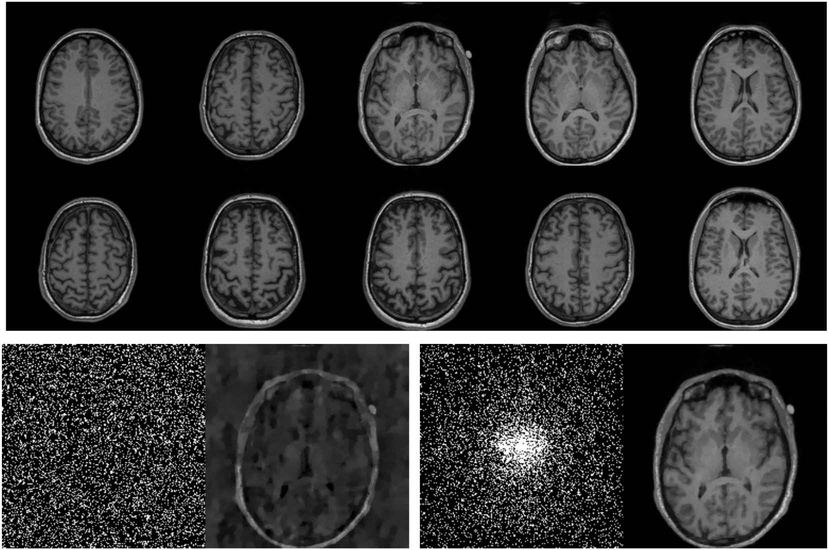


(e) Learned sampling pattern



(f) Largest 2.76% Fourier Coefficients

Classical compressed sensing versus learned Sherry et al. 2020



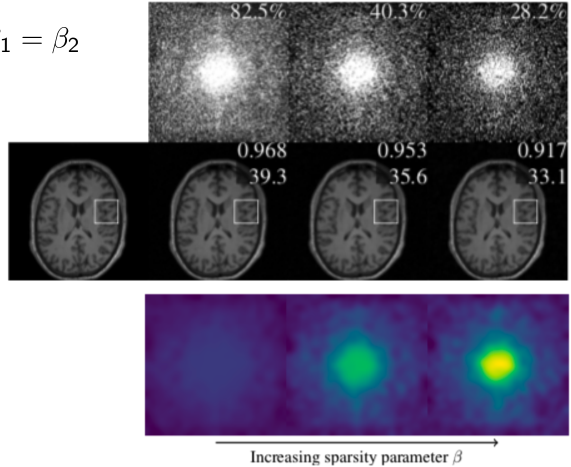
Uniform random | Reconstruction | Learned | Reconstruction

Increasing sparsity Sherry et al. 2020

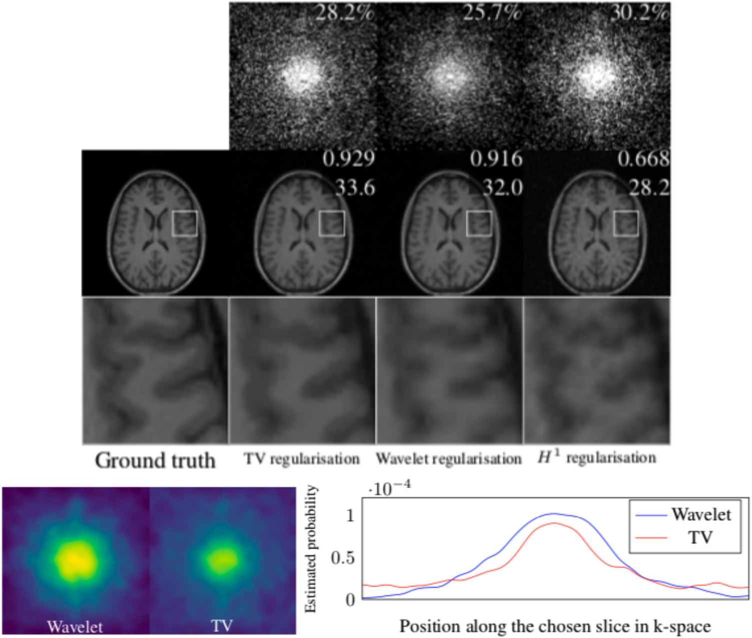
Reminder: **Upper level** (learning)

$$\min_{\lambda \geq 0, s \in [0,1]^m} \frac{1}{n} \sum_{i=1}^{n} \|R(\lambda, s, y_i) - x_i\|_2^2 + \beta_1 \|s\|_1 + \beta_2 \|s(1-s)\|_1$$

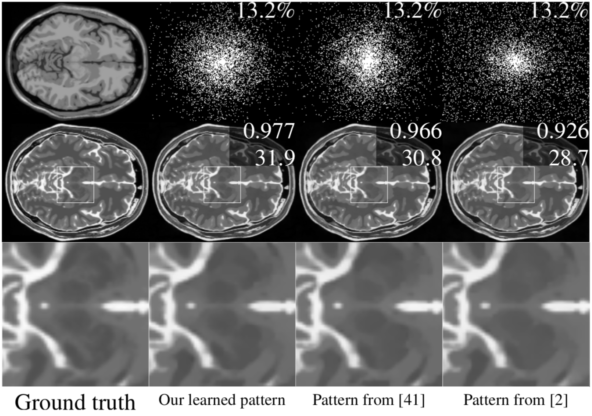
$$\beta = \beta_1 = \beta_2$$



Compare regularizers Sherry et al. 2020



Compare "free" samplings Sherry et al. 2020



	Pattern type	SSIM	PSNR
Training	Our method	0.977 ± 0.002	32.5 ± 0.2
	Data-adapted [41]	0.968 ± 0.002	31.1 ± 0.1
	Uninformed VDS [2]	0.925 ± 0.005	28.9 ± 0.1
Testing	Our method	0.975 ± 0.003	32.1 ± 0.2
	Data-adapted [41]	0.967 ± 0.003	31.1 ± 0.2
	Uninformed VDS [2]	0.924 ± 0.003	28.8 ± 0.1

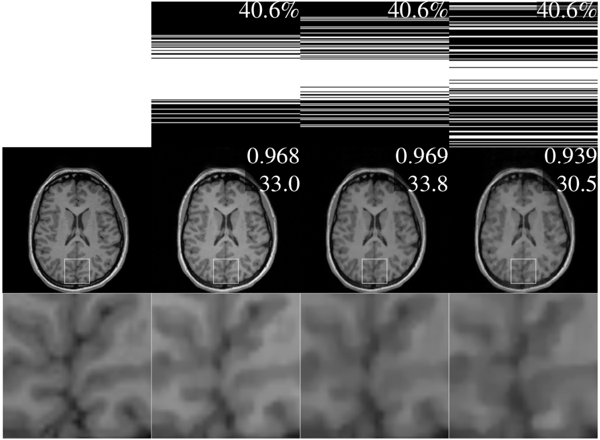
"ours" = Sherry et al. 2020

[41] = Knoll et al. 2011

[2] = Lustig et al. 2007

regularizer = dTV Ehrhardt and Betcke 2016

Compare Cartesian samplings Sherry et al. 2020



Ground truth | Our learned pattern | Pattern from [23] | Pattern from [2]

	Line sampling (40.6%)	Free pattern (34.7%)
Our method	4192	6494
The method from [23]	12087	$3.90 \cdot 10^8$

number of lower-level solves

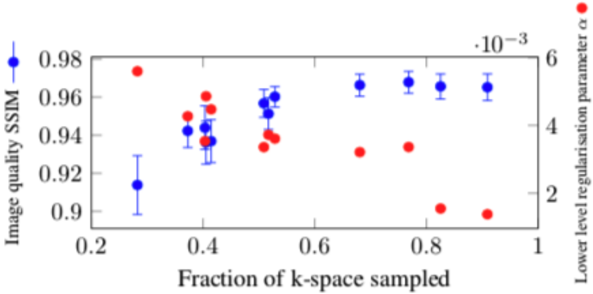
"ours" = Sherry et al. 2020

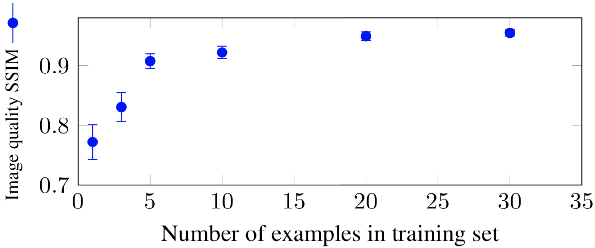
[23] = Gözcü et al. 2018

[2] = Lustig et al. 2007

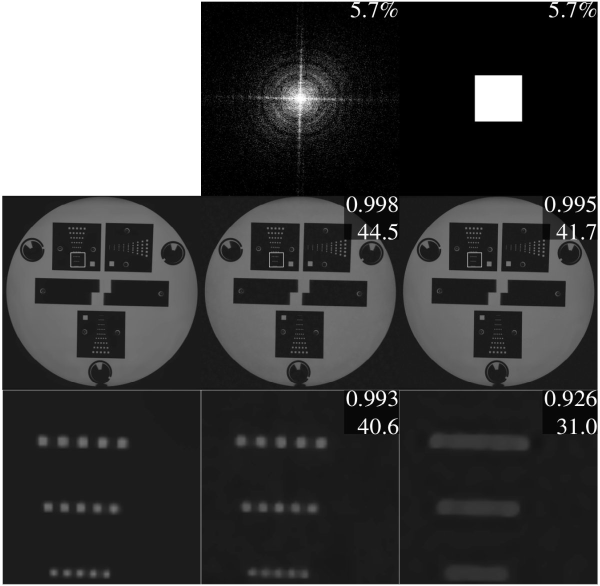
regularizer = TV

More insights: sampling and number of data Sherry et al. 2020





High resolution imaging: 1024^2 Sherry et al. 2020



Inexact Algorithms for Bilevel Learning

Bilevel learning: Reduced formulation

Upper level:

$$\min_{\lambda \geq 0, \hat{x}} \|\hat{x} - x^\dagger\|_2^2$$

Lower level:

$$\hat{x} = \arg\min_x \{\mathcal{D}(Ax, y) + \lambda \mathcal{R}(x)\}$$

Bilevel learning: Reduced formulation

Upper level:

$$\min_{\lambda \geq 0, \hat{x}} U(\hat{x})$$

Lower level:

$$\hat{x} = \arg\min_{x} \{\mathcal{D}(Ax, y) + \lambda \mathcal{R}(x)\}$$

Bilevel learning: Reduced formulation

Upper level:

$$\min_{\lambda \geq 0, \hat{x}} U(\hat{x})$$

Lower level:

$$\hat{x} = \arg\min_{x} L(x, \lambda)$$

Bilevel learning: Reduced formulation

Upper level:
$$\min_{\lambda \geq 0, \hat{x}} U(\hat{x})$$

Lower level:
$$x_\lambda := \hat{x} = \arg\min_x L(x, \lambda)$$

Reduced formulation: $\min_{\lambda \geq 0} U(x_\lambda) =: \tilde{U}(\lambda)$

Bilevel learning: Reduced formulation

Upper level: $\min_{\lambda \geq 0, \hat{x}} U(\hat{x})$

Lower level:

$$x_\lambda := \hat{x} = \arg\min_x L(x, \lambda) \quad \Leftrightarrow \quad \partial_x L(x_\lambda, \lambda) = 0$$

Reduced formulation: $\min_{\lambda \geq 0} U(x_\lambda) =: \tilde{U}(\lambda)$

Bilevel learning: Reduced formulation

Upper level: $$\min_{\lambda \geq 0, \hat{x}} U(\hat{x})$$

Lower level:

$$x_\lambda := \hat{x} = \arg\min_x L(x, \lambda) \quad \Leftrightarrow \quad \partial_x L(x_\lambda, \lambda) = 0$$

Reduced formulation: $$\min_{\lambda \geq 0} U(x_\lambda) =: \tilde{U}(\lambda)$$

$$0 = \partial_x^2 L(x_\lambda, \lambda) \partial_\lambda x_\lambda + \partial_\theta \partial_x L(x_\lambda, \lambda) \quad \Leftrightarrow \quad \partial_\lambda x_\lambda = -B^{-1} A$$

Bilevel learning: Reduced formulation

Upper level:
$$\min_{\lambda \geq 0, \hat{x}} U(\hat{x})$$

Lower level:
$$x_\lambda := \hat{x} = \arg\min_x L(x, \lambda) \quad \Leftrightarrow \quad \partial_x L(x_\lambda, \lambda) = 0$$

Reduced formulation: $\min_{\lambda \geq 0} U(x_\lambda) =: \tilde{U}(\lambda)$

$$0 = \partial_x^2 L(x_\lambda, \lambda) \partial_\lambda x_\lambda + \partial_\theta \partial_x L(x_\lambda, \lambda) \quad \Leftrightarrow \quad \partial_\lambda x_\lambda = -B^{-1} A$$

$$\nabla \tilde{U}(\lambda) = (\partial_\lambda x_\lambda)^* \nabla U(x_\lambda)$$

Bilevel learning: Reduced formulation

Upper level: $$\min_{\lambda \geq 0, \hat{x}} U(\hat{x})$$

Lower level:
$$x_\lambda := \hat{x} = \arg\min_x L(x, \lambda) \quad \Leftrightarrow \quad \partial_x L(x_\lambda, \lambda) = 0$$

Reduced formulation: $$\min_{\lambda \geq 0} U(x_\lambda) =: \tilde{U}(\lambda)$$

$$0 = \partial_x^2 L(x_\lambda, \lambda) \partial_\lambda x_\lambda + \partial_\theta \partial_x L(x_\lambda, \lambda) \quad \Leftrightarrow \quad \partial_\lambda x_\lambda = -B^{-1} A$$

$$\begin{aligned} \nabla \tilde{U}(\lambda) &= (\partial_\lambda x_\lambda)^* \nabla U(x_\lambda) \\ &= -A^* B^{-1} \nabla U(x_\lambda) = -A^* w \end{aligned}$$

where w solves $Bw = \nabla U(x_\lambda)$.

Algorithm for Bilevel learning

Upper level: $\min_{\lambda \geq 0, \hat{x}} U(\hat{x})$

Lower level: $x_\lambda := \arg\min_x L(x, \lambda)$

Reduced formulation: $\min_{\lambda \geq 0} U(x_\lambda) =: \tilde{U}(\lambda)$

- Solve reduced formulation via L-BFGS-B Nocedal and Wright 2000
- Compute gradients: Given λ
  (1) Compute x_λ, e.g. via PDHG Chambolle and Pock 2011
  (2) Solve $Bw = \nabla U(x_\lambda)$, $B := \partial_x^2 L(x_\lambda, \lambda)$ e.g. via CG
  (3) Compute $\nabla \tilde{U}(\lambda) = -A^* w$, $A := \partial_\theta \partial_x L(x_\lambda, \lambda)$

Algorithm for Bilevel learning

Upper level: $\min_{\lambda \geq 0, \hat{x}} U(\hat{x})$

Lower level: $x_\lambda := \arg\min_x L(x, \lambda)$

Reduced formulation: $\min_{\lambda \geq 0} U(x_\lambda) =: \tilde{U}(\lambda)$

- Solve reduced formulation via L-BFGS-B Nocedal and Wright 2000
- Compute gradients: Given λ
  (1) Compute x_λ, e.g. via PDHG Chambolle and Pock 2011
  (2) Solve $Bw = \nabla U(x_\lambda)$, $B := \partial_x^2 L(x_\lambda, \lambda)$ e.g. via CG
  (3) Compute $\nabla \tilde{U}(\lambda) = -A^* w$, $A := \partial_\theta \partial_x L(x_\lambda, \lambda)$

This approach has a number of problems:

- x_λ has to be computed
- Derivative assumes x_λ is exact minimizer
- Large system of linear equations has to be solved

How to solve Bilevel Problem?

- Most people: Ignore "problems", just compute it. e.g. Sherry et al. 2020
- Semi-smooth Newton: similar fundamental problems Kunisch and Pock 2013
- Replace lower level problem by finite number of iterations of algorithms: not bilevel anymore Ochs et al. 2015
- Use algorithm that does not need x_λ, gradients etc Ehrhardt and Roberts 2020

Dynamic Accuracy Derivative Free Optimization

$$\min_{\theta} f(\theta)$$

Key idea: make use of $g(\theta, \epsilon)$

$$|f(\theta) - g(\theta, \epsilon)| < \epsilon$$

inexact minimisation of f early, only ask for high accuracy when needed

If $g(\theta^{k+1}, \epsilon) < g(\theta^k, \epsilon) - 2\epsilon$, then $f(\theta^{k+1}) < f(\theta^k)$.

Dynamic Accuracy Derivative Free Optimization

$$\min_{\theta} f(\theta)$$

Key idea: make use of $g(\theta, \epsilon)$

$$|f(\theta) - g(\theta, \epsilon)| < \epsilon$$

inexact minimisation of f early, only ask for high accuracy when needed

If $g(\theta^{k+1}, \epsilon) < g(\theta^k, \epsilon) - 2\epsilon$, then $f(\theta^{k+1}) < f(\theta^k)$.

For $k = 0, 1, 2, \ldots$

1) Sample f in a neighbourhood of θ_k
2) Build model $m_k(\theta) \approx f$
3) Minimise m_k in a neighbourhood of θ_k to get θ_{k+1}

Algorithm 1 Dynamic accuracy DFO algorithm for (22).

Inputs: Starting point $\theta^0 \in \mathbb{R}^n$, initial trust-region radius $0 < \Delta^0 \le \Delta_{\max}$.

Parameters: strictly positive values $\Delta_{\max}, \gamma_{\text{dec}}, \gamma_{\text{inc}}, \eta_1, \eta_2, \eta_1', \epsilon$ satisfying $\gamma_{\text{dec}} < 1 < \gamma_{\text{inc}}$, $\eta_1 \le \eta_2 < 1$, and $\eta_1' < \min(\eta_1, 1 - \eta_2)/2$.

Select an arbitrary interpolation set and construct m^0 (26).
for $k = 0, 1, 2, \ldots$ **do**
  repeat
    Evaluate $\widetilde{f}(\theta^k)$ to sufficient accuracy that (32) holds with η_1' (using s^k from the previous iteration of this inner repeat/until loop). Do nothing in the first iteration of this repeat/until loop.
    if $\|g^k\| \le \epsilon$ **then**
      By replacing Δ^k with $\gamma_{\text{dec}}^i \Delta^k$ for $i = 0, 1, 2, \ldots$, find m^k and Δ^k such that m^k is fully linear in $B(\theta^k, \Delta^k)$ and $\Delta^k \le \|g^k\|$. *[criticality phase]*
    end if
    Calculate s^k by (approximately) solving (27).
  until the accuracy in the evaluation of $\widetilde{f}(\theta^k)$ satisfies (32) with η_1' *[accuracy phase]*
  Evaluate $\widetilde{f}(\theta^k + s^k)$ so that (32) is satisfied with η_1' for $\widetilde{f}(\theta^k + s^k)$, and calculate $\widetilde{\rho}^k$ (29).
  Set θ^{k+1} and Δ^{k+1} as:

$$\theta^{k+1} = \begin{cases} \theta^k + s^k, & \widetilde{\rho}^k \ge \eta_2, \text{ or } \widetilde{\rho}^k \ge \eta_1 \text{ and } m^k \\ & \text{fully linear in } B(\theta^k, \Delta^k), \\ \theta^k, & \text{otherwise,} \end{cases} \tag{33}$$

and

$$\Delta^{k+1} = \begin{cases} \min(\gamma_{\text{inc}} \Delta^k, \Delta_{\max}), & \widetilde{\rho}^k \ge \eta_2, \\ \Delta^k, & \widetilde{\rho}^k < \eta_2 \text{ and } m^k \text{ not} \\ & \text{fully linear in } B(\theta^k, \Delta^k), \\ \gamma_{\text{dec}} \Delta^k, & \text{otherwise.} \end{cases} \tag{34}$$

  If $\theta^{k+1} = \theta^k + s^k$, then build m^{k+1} by adding θ^{k+1} to the interpolation set (removing an existing point). Otherwise, set $m^{k+1} = m^k$ if m^k is fully linear in $B(\theta^k, \Delta^k)$, or form m^{k+1} by making m^k fully linear in $B(\theta^{k+1}, \Delta^{k+1})$.
end for

Theoretical Guarantees

Algorithm converges with inexact evaluations of $\hat{x}_i(\theta)$:

Theorem Ehrhardt and Roberts 2020

If f is sufficiently smooth and bounded below, then:

- The Dynamic Accuracy DFO algorithm is globally convergent in the sense that $\lim_{k\to\infty} \|\nabla f(\theta_k)\| = 0$.
- All evaluations of $\hat{x}_i(\theta)$ together require at most $\mathcal{O}(\epsilon^{-2}|\log \epsilon|)$ iterations (of gradient descent, FISTA etc.)

Numerical Results

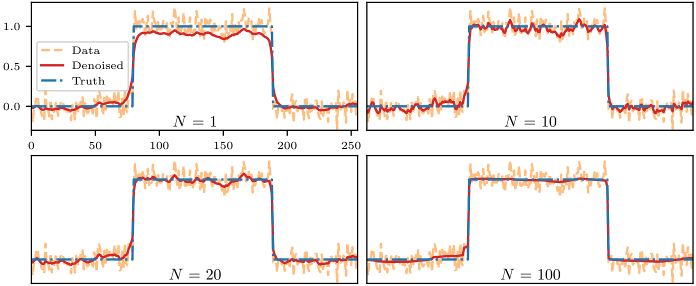
- Dynamic Accuracy DFO
  github.com/lindonroberts/inexact_dfo_bilevel_learning
- Use gradient descent & FISTA to calculate $\hat{x}_i(\theta) = \min_x L_i(x, \theta)$
  - Using known Lipschitz and strong convexity constants (depending on θ)
  - Allow arbitrary accuracy in $\hat{x}_i(\theta)$: terminate when $\|\nabla_x L_i\|$ sufficiently small
  - A priori linear convergence bounds too conservative in practice
- Compare to regular DFO with "fixed accuracy" lower-level solutions (constant # iterations of GD/FISTA)
  - In practice, have to guess appropriate # iterations
- Measure decrease in $f(\theta)$ as function of total GD/FISTA iterations

1D Denoising Problem (learn α, ν and ξ)

$$\min_{\theta} \left\{ f(\theta) = \frac{1}{2} \sum_i \|x_i(\theta) - x_i\|_2^2 + \beta \left(\frac{L(\theta)}{\kappa(\theta)} \right)^2 \right\}$$

$$x_i(\theta) = \arg\min_x \frac{1}{2}\|x - y_i\|_2^2 + \alpha \left(\sum_j \sqrt{\|(\nabla x)_j\|_2^2 + \nu^2} + \frac{\xi}{2}\|x\|_2^2 \right)$$

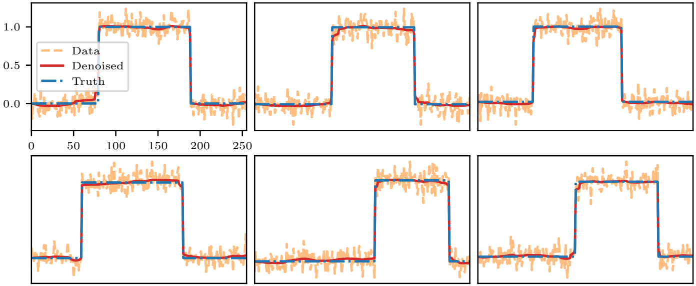
With more evaluations of $f(\theta)$, the parameter choices give better reconstructions:



Reconstruction of x_1 after N evaluations of $f(\theta)$

1D Denoising Problem (learn α, ν and ξ)

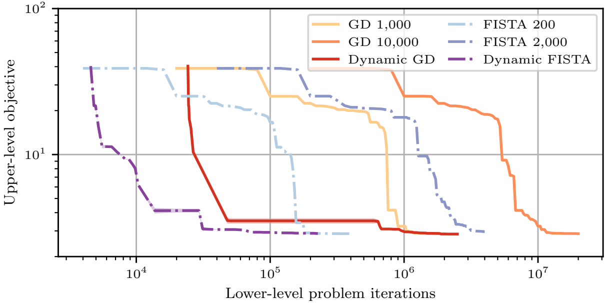
Final learned parameters give good reconstructions of all training data:



Final reconstructions after 100 **evaluations of** $f(\theta)$

1D Denoising Problem (learn α, ν and ξ)

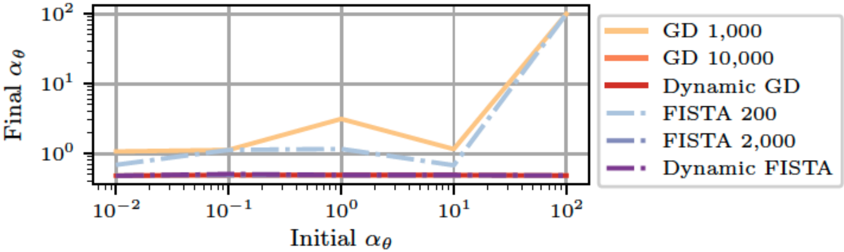
Dynamic accuracy is faster than “fixed accuracy” (at least 10x speedup):



Objective value $f(\theta)$ vs. computational effort

1D Denoising Problem

Always learns the same parameter for sufficient accuracy.



Robustness to initialization

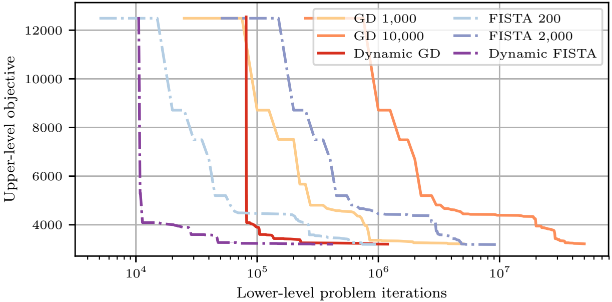
2D Denoising Problem (learn α, ν and ξ)

2D denoising — final learned parameters give good reconstructions...

Final reconstructions after 100 **evaluations of** $f(\theta)$

2D Denoising Problem (learn α, ν and ξ)

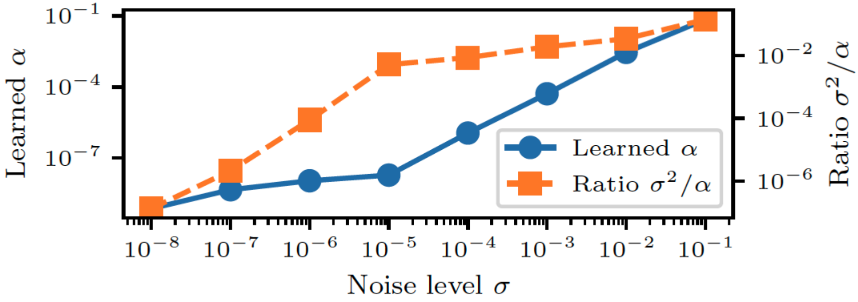
2D denoising — ... and dynamic accuracy is still 10x faster than fixed accuracy:



Objective value $f(\theta)$ vs. computational effort

2D Denoising Problem (learn α, ν and ξ)

Conjecture: Bilevel learning is a convergent regularization.



Convergent regularization?

MRI Sampling revisited

MRIs measure a subset of Fourier coefficients of an image: reconstruct using

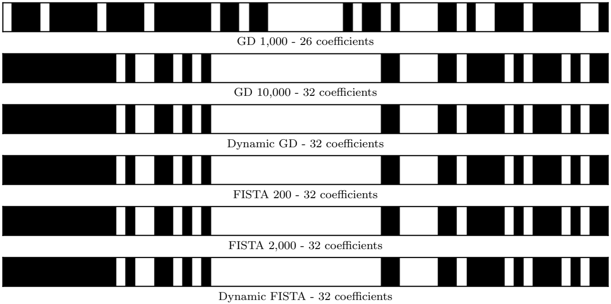
$$\min_x \frac{1}{2}\|S(Fx-y)\|^2 + \mathcal{R}(x)$$

where sampling pattern $S = \mathrm{diag}(s_1, \ldots, s_d)$.

- Use same smoothed TV regulariser $\mathcal{R}$ (with fixed α, ν, ξ)
- Learn $s_j(\theta) := \sqrt{\theta_j/(1-\theta_j)}$ Chen et al. 2014
- Promote sparsity: $\mathcal{J}(\theta) = \|\theta\|_1$.

Learning MRI Sampling Patterns

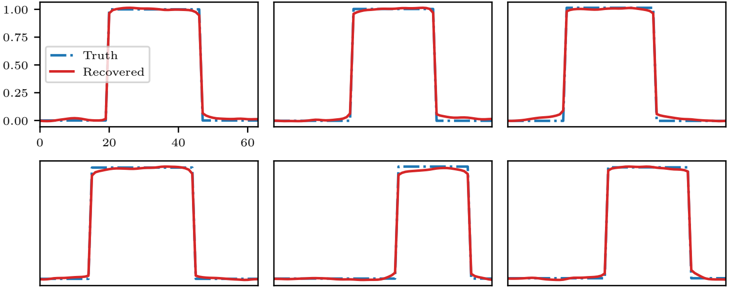
All variants learn 50% sparse sampling patterns:



Learned sampling patterns (white = active)

Learning MRI Sampling Patterns

Learned sampling patterns give good reconstructions:

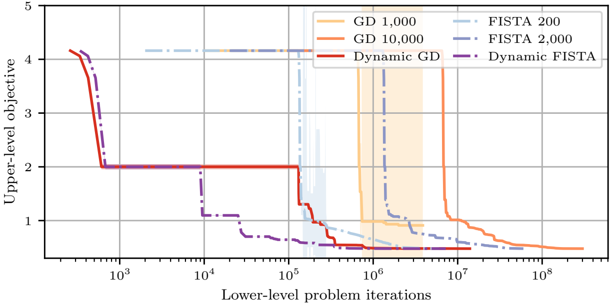


Final reconstructions after 3000 **evaluations of** $f(\theta)$

Robustness to lower-level solver with "enough" accuracy

Learning MRI Sampling Patterns

… and dynamic accuracy is still substantially faster than fixed accuracy:



Objective value $f(\theta)$ vs. computational effort

Conclusions and Outlook

Conclusions

- **Bilevel learning**: supervised learning framework to learn parameters in variational regularization
- **Learned sampling** better than generic sampling
  - "Optimal" sampling **depends on regularizer**
  - **Very little data** needed
- **Optimization** plays a key role in bilevel learning
  - **Dynamic accuracy**: no need to specify number of iterations
  - Improved algorithms **speed up** learning significantly
  - Make learning **surprisingly robust**

Future work

- **Stochastic** algorithms (like stochastic gradient descent etc)
- **Nonsmooth** or **nonconvex** lower-level problems
- **Inexact gradient** methods