Optimising MRI Sampling with Bi-Level Learning

Matthias J. Ehrhardt

Institute for Mathematical Innovation, University of Bath, UK

May 29, 2020

Joint work with:
Sherry, Graves, Maierhofer, Williams, Schönlieb (all Cambridge, UK), Benning (Queen Mary, UK), De los Reyes (EPN, Ecuador)

The Leverhulme Trust

UKRI Engineering and Physical Sciences Research Council

THE FARADAY INSTITUTION

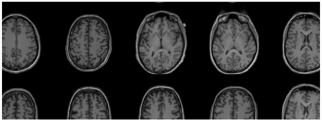
Outline

1) What are inverse problems?

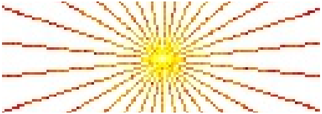
2) How to solve inverse problems?

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

3) Bi-level Learning

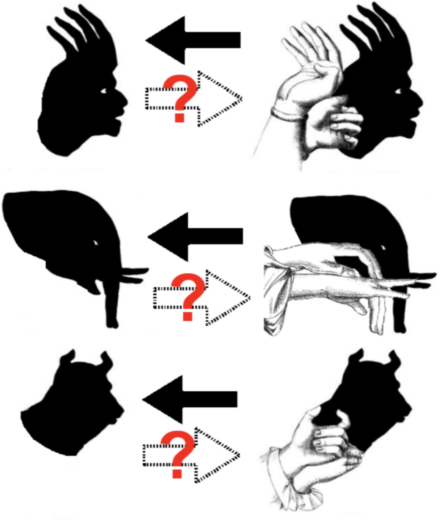


4) Learn sampling pattern in MRI



What are inverse problems?

What are inverse problems? Inverse to what?



Right to left:
forward problem (easy)

Left to right:
inverse problem (hard)

$$Ax = y$$

x : 3D image of hands

y : 2D shadow of hands

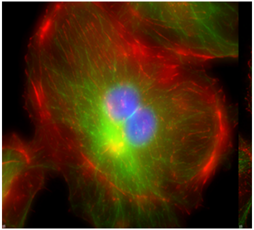
A : mathematical model

Goal: recover x given y

Example: Image Deblurring

traffic control

astronomy



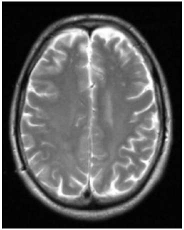
cell biology

Model: Convolution $Ax(t) = x * k(s) = \int_{\mathbb{R}^2} x(t)k(s-t)dt$

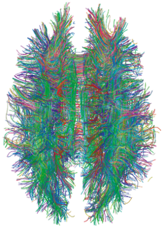
$\rightarrow$

Example: Magnetic Resonance Imaging (MRI)

clinical MRI scanner

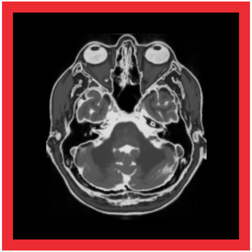


T_2^* weighted MRI

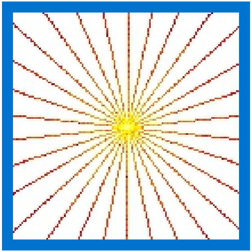


diffusion tensor imaging

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

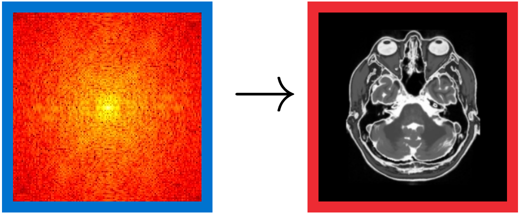


$\longrightarrow$



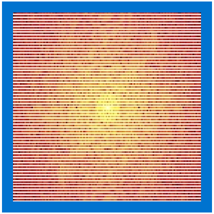
What is the problem with inverse problems?

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

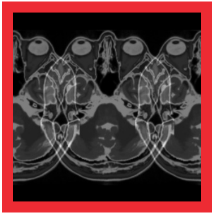


What is the problem with inverse problems?

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

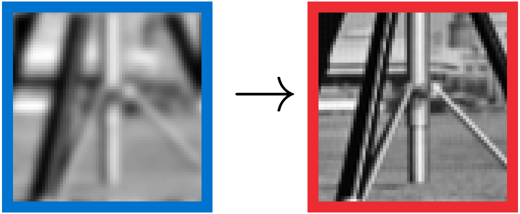


$\longrightarrow$



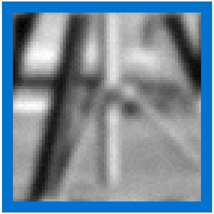
What is the problem with inverse problems?

Deblurring: $Ax = y \quad Ax(s) = \int_{\mathbb{R}^2} x(t)k(s-t)dt$

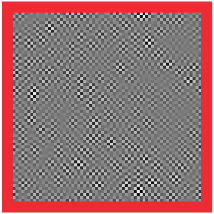


What is the problem with inverse problems?

Deblurring: $Ax = y \quad Ax(s) = \int_{\mathbb{R}^2} x(t)k(s-t)dt$

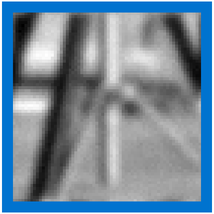
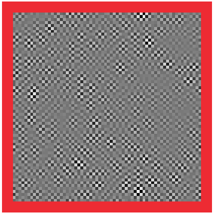


$\longrightarrow$



What is the problem with inverse problems?

Deblurring: $Ax = y \quad Ax(s) = \int_{\mathbb{R}^2} x(t)k(s-t)dt$

 $\longrightarrow$ 

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

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

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

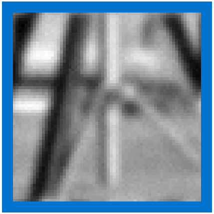
(3) x^* depends **continuously** on data y.

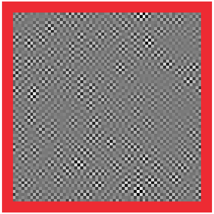
Otherwise, it is called **ill-posed**.

Jacques Hadamard

What is the problem with inverse problems?

Deblurring: $Ax = y \quad Ax(s) = \int_{\mathbb{R}^2} x(t)k(s-t)dt$





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?

How to solve inverse problems?

Variational regularization ($\sim$2000)
Approximate a solution x^* of $Ax = y$ via

$$\hat{x} \in \arg\min_x \left\{ \frac{1}{2}\|Ax - y\|_2^2 + \lambda \mathcal{R}(x) \right\}$$

$\mathcal{R}$ **regularizer**: penalizes unwanted features and ensures stability

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 $\mathcal{R}$.

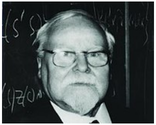
textbooks: Scherzer et al. 2008, Ito and Jin 2015, Benning and Burger 2018

Example: Regularizers

Tikhonov regularization ($\sim$1960):

$\mathcal{R}(x) = \frac{1}{2}\|x\|_2^2$

$$\hat{x} = \arg\min_{x} \left\{ \frac{1}{2}\|Ax - y\|_2^2 + \frac{\lambda}{2}\|x\|_2^2 \right\}$$



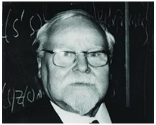
Andrey Tikhonov

Example: Regularizers

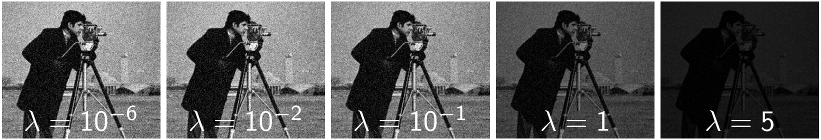
Tikhonov regularization ($\sim$1960):

$\mathcal{R}(x) = \frac{1}{2}\|x\|_2^2$

$$\hat{x} = \arg\min_{x} \left\{ \frac{1}{2}\|Ax - y\|_2^2 + \frac{\lambda}{2}\|x\|_2^2 \right\}$$



Andrey Tikhonov

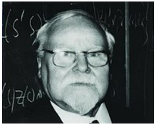


Example: Regularizers

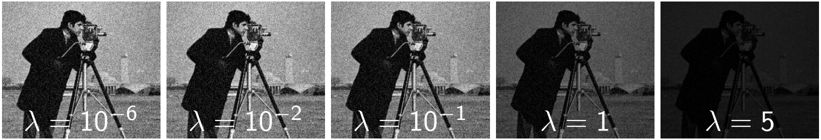
Tikhonov regularization ($\sim$1960):

$\mathcal{R}(x) = \frac{1}{2}\|x\|_2^2$

$$\hat{x} = \arg\min_{x} \left\{ \frac{1}{2}\|Ax - y\|_2^2 + \frac{\lambda}{2}\|x\|_2^2 \right\}$$



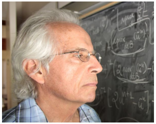
Andrey Tikhonov



Total Variation regularization:

$\mathcal{R}(x) = \|\nabla x\|_1$ Rudin, Osher, Fatemi 1992

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



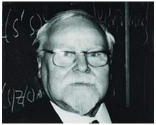
Stanley Osher

Example: Regularizers

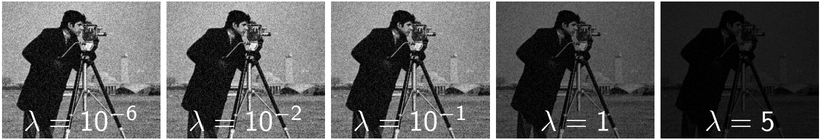
Tikhonov regularization ($\sim$1960):

$\mathcal{R}(x) = \frac{1}{2}\|x\|_2^2$

$$\hat{x} = \arg\min_x \left\{ \frac{1}{2}\|Ax - y\|_2^2 + \frac{\lambda}{2}\|x\|_2^2 \right\}$$



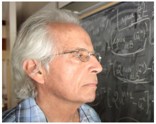
Andrey Tikhonov



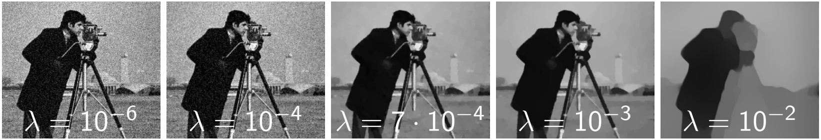
Total Variation regularization:

$\mathcal{R}(x) = \|\nabla x\|_1$ Rudin, Osher, Fatemi 1992

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



Stanley Osher



Example: Regularizers

Tikhonov ($\sim$1960)

$$\mathcal{R}(x) = \frac{1}{2}\|x\|_2^2$$

Total Variation Rudin, Osher, Fatemi 1992

$$\mathcal{R}(x) = \|\nabla x\|_1$$

Example: Regularizers

Tikhonov ($\sim$1960)

$$\mathcal{R}(x) = \frac{1}{2}\|x\|_2^2$$

Total Variation Rudin, Osher, Fatemi 1992

$$\mathcal{R}(x) = \|\nabla x\|_1$$

H^1 ($\sim$1960-1990?)

$$\mathcal{R}(x) = \frac{1}{2}\|\nabla x\|_2^2$$

Wavelet sparsity ($\sim$1990)

$$\mathcal{R}(x) = \|Wx\|_1$$

Total Generalized Variation: Bredies, Kunisch, Pock 2010

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

Connection to PDEs

Total Variation regularization:

$\mathcal{R}(x) = \|\nabla x\|_1$ Rudin, Osher, Fatemi 1992

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

"Smooth" Total Variation regularization:

$$\hat{x} = \arg\min_x \left\{ \frac{1}{2}\|Ax - y\|_2^2 + \lambda \int \rho(\nabla x(s))ds + \frac{\varepsilon}{2}\|x\|_2^2 \right\}$$

- $\rho(t) = \|t\|_2^2$
- $\rho(t) = \sqrt{\|t\|_2^2 + \gamma^2}$ or Huber loss
- strongly convex and smooth optimization problem

Connection to PDEs

Total Variation regularization:

$\mathcal{R}(x) = \|\nabla x\|_1$ Rudin, Osher, Fatemi 1992

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

"Smooth" Total Variation regularization:

$$\hat{x} = \arg\min_x \left\{ \frac{1}{2}\|Ax - y\|_2^2 + \lambda \int \rho(\nabla x(s))ds + \frac{\varepsilon}{2}\|x\|_2^2 \right\}$$

$$\Leftrightarrow \quad (A^*A + \varepsilon I)\hat{x} - \lambda \operatorname{div} \rho'(\nabla \hat{x}) = A^* y$$

- $\rho(t) = \|t\|_2^2 \quad \Rightarrow$ linear PDE
- $\rho(t) = \sqrt{\|t\|_2^2 + \gamma^2}$ or Huber loss $\quad \Rightarrow$ nonlinear PDE
- strongly convex and smooth optimization problem

Example: MRI reconstruction

Compressed Sensing MRI:

$A = S_\Omega \circ F$ Lustig, Donoho, Pauly 2007

Fourier transform F, sampling $S_\Omega w = w|_\Omega$

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

Miki Lustig

Example: MRI reconstruction

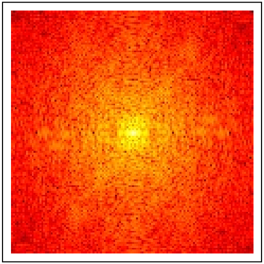
Compressed Sensing MRI:

$A = S_\Omega \circ F$ Lustig, Donoho, Pauly 2007

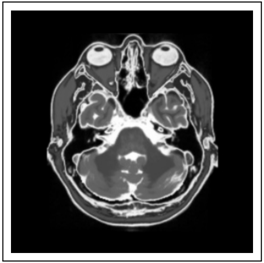
Fourier transform F, sampling $S_\Omega w = w|_\Omega$

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

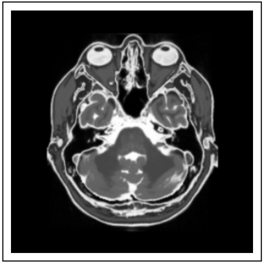
Miki Lustig



sampling $S_\Omega^* y$



$\lambda = 0$



$\lambda = 1$

Example: MRI reconstruction

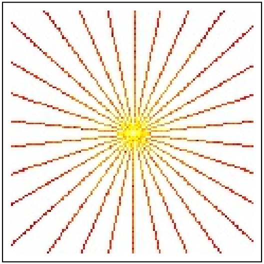
Compressed Sensing MRI:

$A = S_\Omega \circ F$ Lustig, Donoho, Pauly 2007

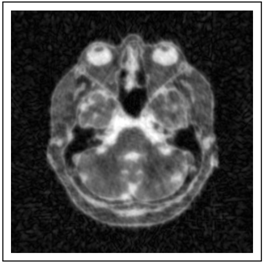
Fourier transform F, sampling $S_\Omega w = w|_\Omega$

$$\hat{x} \in \operatorname*{arg\,min}_x \left\{ \frac{1}{2} \|S_\Omega F x - y\|_2^2 + \lambda \|\nabla x\|_1 \right\}$$

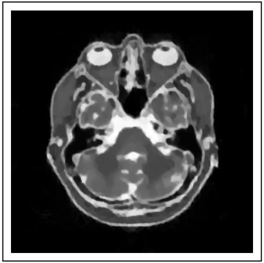
Miki Lustig



sampling $S_\Omega^* y$



$\lambda = 0$



$\lambda = 10^{-4}$

Example: MRI reconstruction

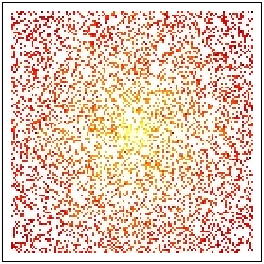
Compressed Sensing MRI:

$A = S_\Omega \circ F$ Lustig, Donoho, Pauly 2007

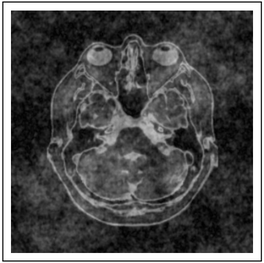
Fourier transform F, sampling $S_\Omega w = w|_\Omega$

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

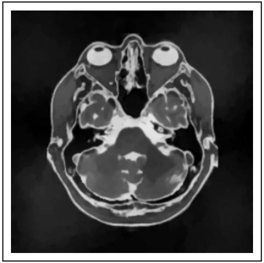
Miki Lustig



sampling $S_\Omega^* y$



$\lambda = 0$



$\lambda = 10^{-4}$

Example: MRI reconstruction

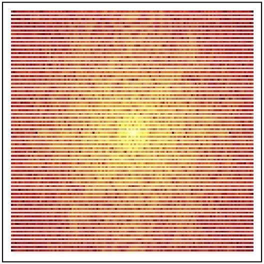
Compressed Sensing MRI:

$A = S_\Omega \circ F$ Lustig, Donoho, Pauly 2007

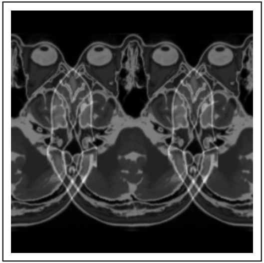
Fourier transform F, sampling $S_\Omega w = w|_\Omega$

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

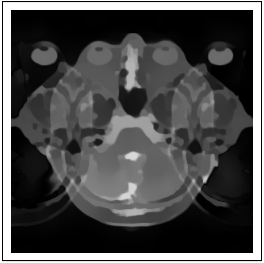
Miki Lustig



sampling $S_\Omega^* y$



$\lambda = 0$



$\lambda = 10^{-3}$

Example: MRI reconstruction

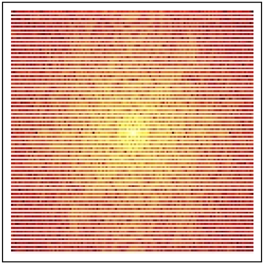
Compressed Sensing MRI:

$A = S_\Omega \circ F$ Lustig, Donoho, Pauly 2007

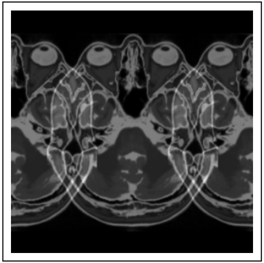
Fourier transform F, sampling $S_\Omega w = w|_\Omega$

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

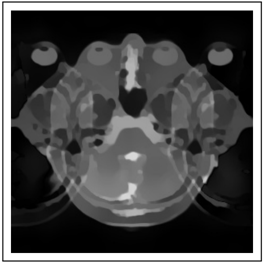
Miki Lustig



sampling $S_\Omega^* y$



$\lambda = 0$



$\lambda = 10^{-3}$

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

Example: MRI reconstruction

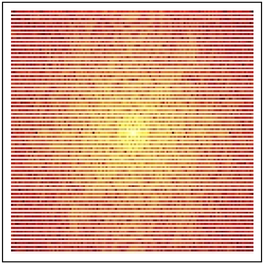
Compressed Sensing MRI:

$A = S_\Omega \circ F$ Lustig, Donoho, Pauly 2007

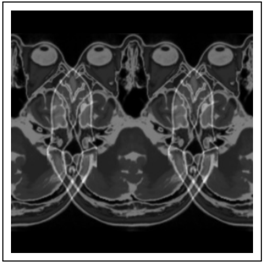
Fourier transform F, sampling $S_\Omega w = w|_\Omega$

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

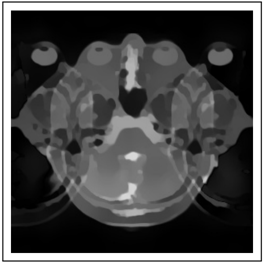
Miki Lustig



sampling $S_\Omega^* y$



$\lambda = 0$



$\lambda = 10^{-3}$

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

Does the optimal sampling depend on the regularizer $\mathcal{R}$?

Bi-level Learning

Bi-level learning for inverse problems

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

$\mathcal{R}$ smooth and strongly convex

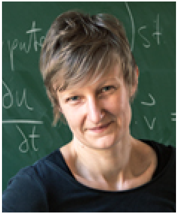
Bi-level learning for inverse problems

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):

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



Carola Schönlieb

$\mathcal{R}$ smooth and strongly convex

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

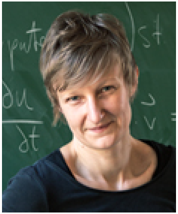
Bi-level learning for inverse problems

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):

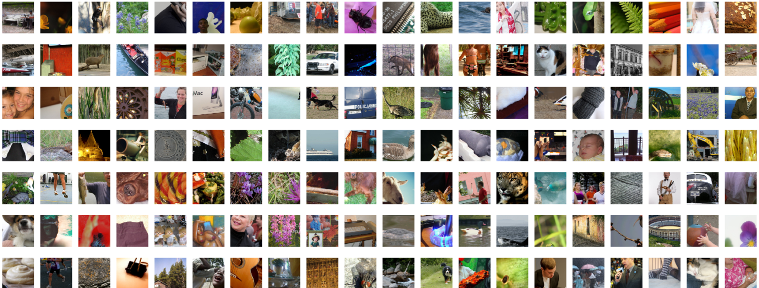
$$\hat{x} = \arg\min_x \left\{ \frac{1}{2}\|Ax - y\|_2^2 + \lambda \mathcal{R}(x) \right\}$$



Carola Schönlieb

$\mathcal{R}$ smooth and strongly convex

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



Bi-level learning for inverse problems

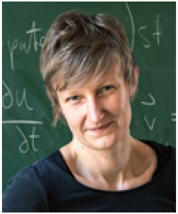
Upper level (learning):

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):

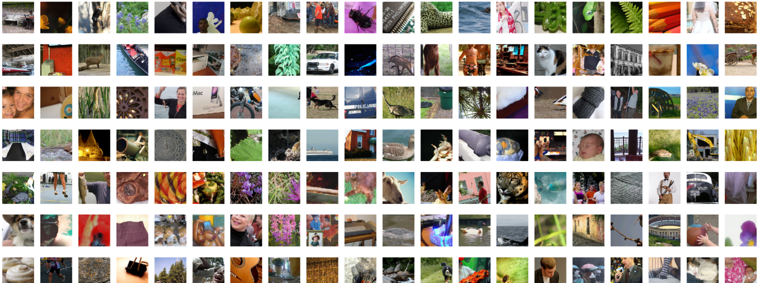
$$\hat{x}_i = \arg\min_x \left\{ \frac{1}{2}\|Ax - y_i\|_2^2 + \lambda \mathcal{R}(x) \right\}$$



Carola Schönlieb

$\mathcal{R}$ smooth and strongly convex

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



Bi-level learning for inverse problems: Reduced formulation

Upper level:

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

Lower level:

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

Bi-level learning for inverse problems: Reduced formulation

Upper level:

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

Lower level:

$$\hat{x} = \arg\min_{x} \left\{ \frac{1}{2} \|Ax - y\|_2^2 + \lambda \mathcal{R}(x) \right\}$$

Bi-level learning for inverse problems: Reduced formulation

Upper level:

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

Lower level:

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

Bi-level learning for inverse problems: 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:
$$\min_{\lambda \geq 0} U(x_\lambda) =: \tilde{U}(\lambda)$$

Bi-level learning for inverse problems: 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: $$\min_{\lambda \geq 0} U(x_\lambda) =: \tilde{U}(\lambda)$$

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

Bi-level learning for inverse problems: 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: $\displaystyle\min_{\lambda \geq 0} U(x_\lambda) =: \tilde{U}(\lambda)$

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

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

Bi-level learning for inverse problems: 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 Bi-level 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 $\nabla \tilde{U}(\lambda) = -A^* w$, $A := \partial_\theta \partial_x L(x_\lambda, \lambda)$

Learn sampling pattern in MRI

Learn sampling pattern in MRI

Upper level (learning):

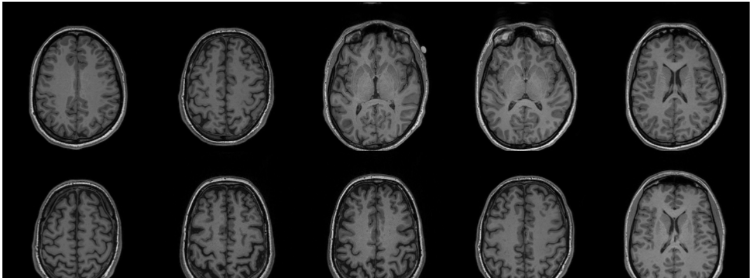
Given **training data** $(x_i^\dagger, y_i)_{i=1}^n$, solve

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

Lower level (MRI reconstruction):

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

Sherry et al. 2019, `https://arxiv.org/pdf/1906.08754.pdf`



Learn sampling pattern in MRI

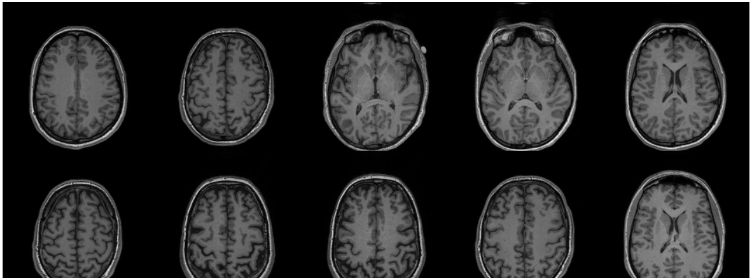
Upper level (learning):
Given **training data** $(x_i^\dagger, y_i)_{i=1}^n$, solve

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

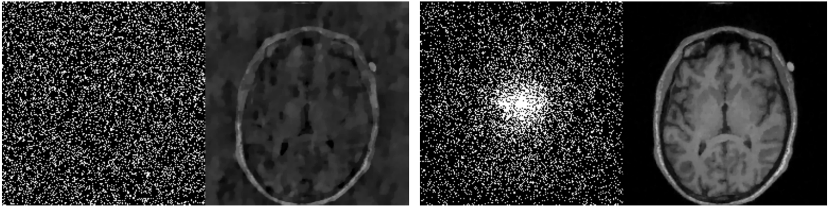
Lower level (MRI reconstruction):

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

Sherry et al. 2019, `https://arxiv.org/pdf/1906.08754.pdf`



Classical compressed sensing versus learned



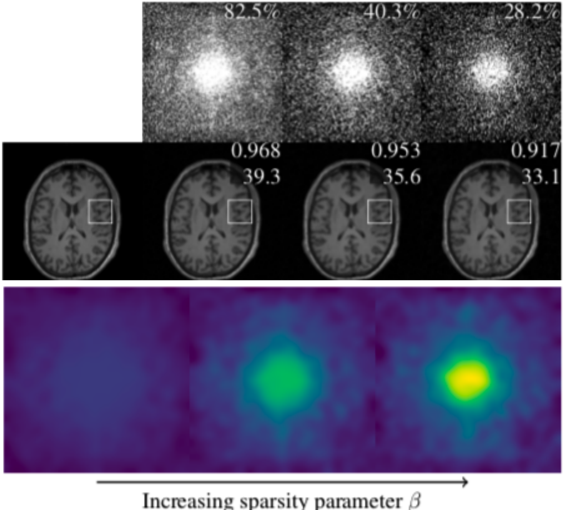
Uniform random Reconstruction Learned Reconstruction

Sherry et al. 2019, `https://arxiv.org/pdf/1906.08754.pdf`

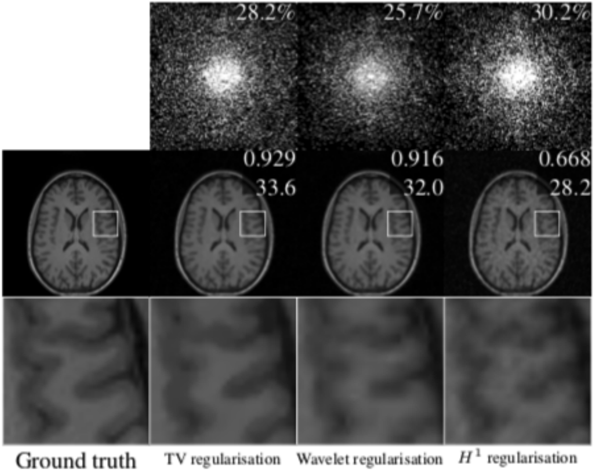
Increasing sparsity

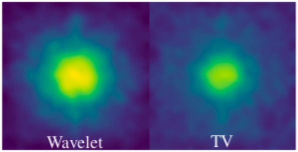
Reminder: **Upper level** (learning)

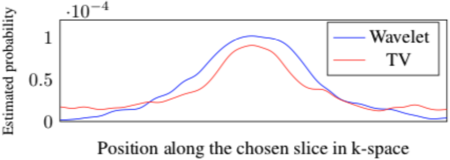
$$\min_{\lambda \geq 0, s \in [0,1]^m} \frac{1}{n} \sum_{i=1}^{n} \| R(\lambda, s, y_i) - x_i \|_2^2 + \beta_1 \| s \|_1 + \beta_2 \| s(1-s) \|_1$$



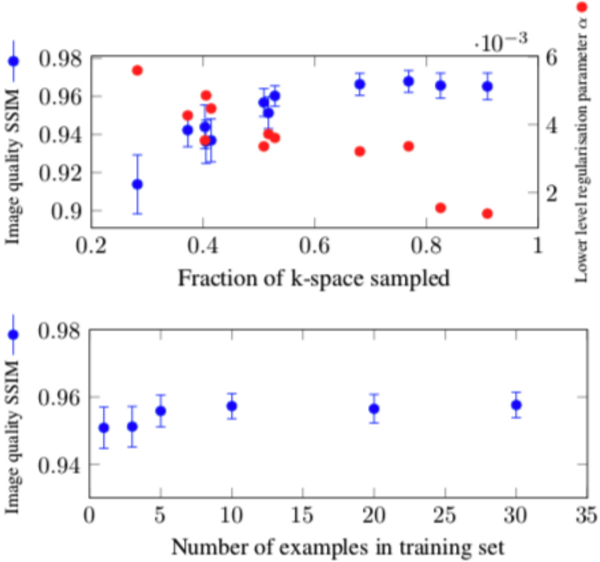
Compare regularizers





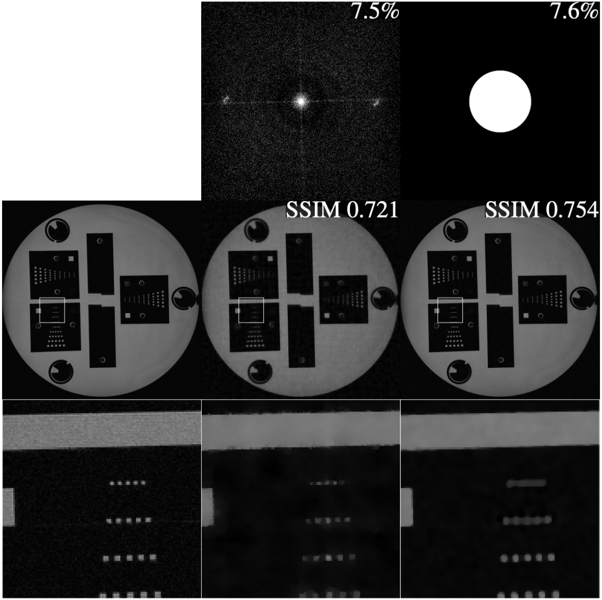


More insights: sampling and number of data



Sherry et al. 2019, `https://arxiv.org/pdf/1906.08754.pdf`

High resolution imaging: 1024^2



Sherry et al. 2019, `https://arxiv.org/pdf/1906.08754.pdf`

Conclusions and outlook

Conclusions

- Be aware of **ill-posedness**: regularization is needed!
- **Variational regularization**: Tikhonov, Total Variation
- Some parameters are **difficult** to choose: regularization parameter, sampling
- **Bi-level / machine learning** is a way out!

Outlook

- Investigate other **algorithms** tailored to problem
  - DFO with errors in objective (joint work with Lindon Roberts)
  - not based on reduced formulation, e.g. nonlinear ADMM
- **Unrolling**: replace lower level problem by algorithm
- **End-to-end learning**: learn reconstruction and sampling