Inexact Algorithms for Bilevel Learning

Matthias J. Ehrhardt

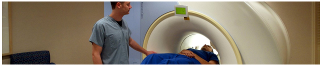
Department of Mathematical Sciences, University of Bath, UK

30 June, 2026

Joint work with: M. S. Salehi (Bath), S. Mukherjee (Kharagpur), L. Roberts (Sydney)

Outline

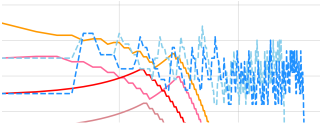
1) Bilevel learning of a regularizer



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

2) Inexact learning strategy

Salehi et al. SIMODS ’25



3) Numerical results

Inverse Problems

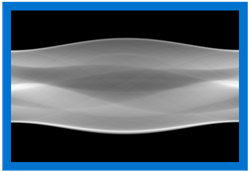
desired solution

mathematical model

observed data

$$Au = b$$

Forward problem: compute b given u



Inverse problem: recover u given b

Inverse Problems

mathematical model desired solution observed data

$$Au = b$$

Variational regularization

Approximate a solution u^* of $Au = b$ via

$$\hat{u} \in \arg\min_u \left\{ \mathcal{D}(Au, b) + \lambda \mathcal{R}(u) \right\}$$

$\mathcal{D}$ **data fidelity**: related to noise statistics

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

$\lambda \geq 0$ **regularization parameter**: weights data and regularizer

Examples of Regularisers

Total Variation (TV) Denoising Rudin et al. 1992

$$\min_{u}\left\{\|u-b\|_2^2+\lambda\mathcal{R}(u)\right\},\quad \mathcal{R}(u)=\int\|\nabla u(x)\|_2 dx$$

noisy

TV regularized

- Denoising without oversmoothing edges
- Results cartoonish
- 1 parameter to tune

Examples of Regularisers

Total Variation (TV) Denoising Rudin et al. 1992 1 parameter

$$\min_{u}\left\{\|u-b\|_2^2+\lambda\mathcal{R}(u)\right\},\quad \mathcal{R}(u)=\int\|\nabla u(x)\|_2 dx$$

Smooth TV 2 parameters

$$\mathcal{R}(u)=\int\phi_\gamma(\nabla u(x))dx,\quad \phi_\gamma(z):=\sqrt{\|z\|_2^2+\gamma^2}$$

Examples of Regularisers

Total Variation (TV) Denoising Rudin et al. 1992 1 parameter

$$\min_u \left\{ \|u - b\|_2^2 + \lambda \mathcal{R}(u) \right\}, \quad \mathcal{R}(u) = \int \|\nabla u(x)\|_2 dx$$

Smooth TV 2 parameters

$$\mathcal{R}(u) = \int \phi_\gamma(\nabla u(x)) dx, \quad \phi_\gamma(z) := \sqrt{\|z\|_2^2 + \gamma^2}$$

Fields-of-Experts (FoE) Roth and Black '05

$$\mathcal{R}(u) = \sum_{k=1}^{K} \lambda_k \int \phi_{\gamma_k}((\kappa_k * u)(x)) dx$$

E.g., 48 kernels 7×7 = 2448 parameters

Other options: ICNN, CRR ... Amos et al. '17, Goujon et al. '22, Mukherjee et al. '24, TDV, wCRR, wICNN, IDCNN ... Kobler et al. '21, Goujon et al. '24, Shumaylov et al. '24, Zhang and Leong '25; Review and comparison Hertrich et al. '25

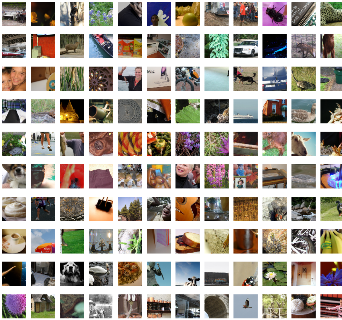
How to Train a Regularizer? Bilevel learning

Upper level (learning):
Given $(u_i, b_i)_{i=1}^n, b_i \approx Au_i$, solve

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \| \hat{u}_i(\theta) - u_i \|_2^2$$

Lower level (solve inverse problem):

$$\hat{u}_i(\theta) = \arg\min_u \{\mathcal{D}(Au, b_i) + \mathcal{R}_\theta(u)\}$$



von Stackelberg 1934, Haber and Tenorio '03, Kunisch and Pock '13, De los Reyes and Schönlieb '13, Crocket and Fessler '22, De los Reyes and Villacis '23

Other options: contrastive learning Hinton '02, fitting prior distribution Roth and Black '05, adversarial training Arjovsky et al. '17, adversarial regularization Lunz et al. '18 ...

How to solve Bilevel Learning Problems: An Inexact Learning Strategy

Salehi et al. SIMODS ’25

Exact Approaches for Bilevel learning

Upper level: $\min_\theta f(\theta) := g(\hat{u}(\theta))$

Lower level: $\hat{u}(\theta) := \arg\min_u h(u, \theta)$

Access to **gradients**: with chain rule $\nabla f(\theta) = (\hat{u}'(\theta))^* \nabla g(\hat{u}(\theta))$ and differentiate optimality condition:

$$0 = \partial_\theta[\partial_u h(\hat{u}(\theta), \theta)] = \partial_u^2 h(\hat{u}(\theta), \theta)\hat{u}'(\theta) + \partial_\theta \partial_u h(\hat{u}(\theta), \theta)$$

1) Compute $\hat{u}(\theta)$
2) Solve $Bw = b, \quad B = \partial_u^2 h(\hat{u}(\theta), \theta), \quad b = \nabla g(\hat{u}(\theta))$
3) Compute $\nabla f(\theta) = -A^* w, \quad A = \partial_\theta \partial_u h(\hat{u}(\theta), \theta)$

Exact Approaches for Bilevel learning

Upper level: $$\min_\theta f(\theta) := g(\hat{u}(\theta))$$

Lower level: $$\hat{u}(\theta) := \arg\min_u h(u, \theta)$$

Access to **gradients**: with chain rule $\nabla f(\theta) = (\hat{u}'(\theta))^* \nabla g(\hat{u}(\theta))$ and differentiate optimality condition:

$$0 = \partial_\theta[\partial_u h(\hat{u}(\theta), \theta)] = \partial_u^2 h(\hat{u}(\theta), \theta)\hat{u}'(\theta) + \partial_\theta \partial_u h(\hat{u}(\theta), \theta)$$

1) Compute $\hat{u}(\theta)$
2) Solve $Bw = b, \quad B = \partial_u^2 h(\hat{u}(\theta), \theta), \quad b = \nabla g(\hat{u}(\theta))$
3) Compute $\nabla f(\theta) = -A^* w, \quad A = \partial_\theta \partial_u h(\hat{u}(\theta), \theta)$

This strategy has a number of problems:

- $\hat{u}(\theta)$ has to be computed
- Derivative assumes $\hat{u}(\theta)$ is exact minimizer
- Large system of linear equations has to be solved

Inexact Approaches for Bilevel learning

Upper level: $$\min_\theta f(\theta) := g(\hat{u}(\theta))$$

Lower level: $$\hat{u}(\theta) := \arg\min_u h(u,\theta)$$

Approximate gradients $z(\theta) \approx \nabla f(\theta)$:

1) Compute $\hat{u}_\varepsilon(\theta)$ to accuracy ε :

$$\|\hat{u}_\varepsilon(\theta) - \hat{u}(\theta)\| < \varepsilon$$

2) Solve $B_\varepsilon w = b_\varepsilon$ to accuracy δ :

$$\|B_\varepsilon w_{\varepsilon,\delta} - b_\varepsilon\| < \delta,$$

with $B_\varepsilon = \partial_u^2 h(\hat{u}_\varepsilon(\theta), \theta)$, $b_\varepsilon = \nabla g(\hat{u}_\varepsilon(\theta))$

3) Compute $z(\theta) = -A_\varepsilon^* w_{\varepsilon,\delta}, \quad A_\varepsilon = \partial_\theta \partial_u h(\hat{u}_\varepsilon(\theta), \theta)$

Construction of Inexact Algorithms

1) Ignore inaccuracy: unrolling, Jacobian-free backprop ...

   Ochs et al. '16, Shaban et al. '19, Fung et al. '22, Bolte et al. '23
2) Zero-order: DFO-LS Ehrhardt and Roberts '21
   - adaptive accuracy using recent research in DFO Cartis et al. '19
   - does not scale well due to lack of gradients
3) First-order: HOAG Pedregosa '16

Compute $z_k = z(\theta_k)$ with accuracies ε_k, δ_k

$$\theta_{k+1} = \theta_k - \alpha_k z_k$$

   - A-prior chosen accuracies ε_k, δ_k
   - Convergence with stepsize $\alpha_k = 1/L$

Construction of Inexact Algorithms

1) Ignore inaccuracy: unrolling, Jacobian-free backprop ...
   Ochs et al. '16, Shaban et al. '19, Fung et al. '22, Bolte et al. '23
2) Zero-order: DFO-LS Ehrhardt and Roberts '21
   - adaptive accuracy using recent research in DFO Cartis et al. '19
   - does not scale well due to lack of gradients
3) First-order: HOAG Pedregosa '16

   Compute $z_k = z(\theta_k)$ with accuracies ε_k, δ_k
   $$\theta_{k+1} = \theta_k - \alpha_k z_k$$

   - A-prior chosen accuracies ε_k, δ_k
   - Convergence with stepsize $\alpha_k = 1/L$

Wish list:

- use "first-order" information: $z(\theta)$
- adaptive accuracy: as low as possible as high as necessary, **minimize compute**
- adaptive step-sizes: as large as possible as small as necessary, **maximize progress**

Ingredient 1: Inexact Gradient as a Descent Direction

Q: How to get descent with $z_k = z(\theta_k)$ for accuracies ε_k, δ_k?

Assumption: all functions sufficiently smooth, h strongly convex

- $h(u, \theta)$ is strongly convex in u
- h is twice differentiable and $\partial_u h(u, \theta), \partial_u^2 h(u, \theta)$ and $\partial_{u\theta}^2 h(u, \theta)$ are Lipschitz in u
- g and f are L_g-smooth and L_f-smooth, respectively

Ingredient 1: Inexact Gradient as a Descent Direction

Q: How to get descent with $z_k = z(\theta_k)$ for accuracies ε_k, δ_k?

Assumption: all functions sufficiently smooth, h strongly convex

- $h(u, \theta)$ is strongly convex in u
- h is twice differentiable and $\partial_u h(u, \theta), \partial_u^2 h(u, \theta)$ and $\partial_{u\theta}^2 h(u, \theta)$ are Lipschitz in u
- g and f are L_g-smooth and L_f-smooth, respectively

Lem: If $\|z_k - \nabla f(\theta_k)\| < \|z_k\|$, then $-z_k$ is a descent direction for f at θ_k.

Lem: Ehrhardt and Roberts, IMA AM '24 There exists computable ω_k (dep. on $\hat{u}_k := \hat{u}_{\varepsilon_k}(\theta_k), \varepsilon_k, \delta_k$) such that $\|z_k - \nabla f(\theta_k)\| \leq \omega_k$.

1) Given ε_k, δ_k, compute $\hat{u}_k, z_k$ and ω_k
2) If $\omega_k \geq \|z_k\|$, go to step 1) with smaller ε_k, δ_k

Thm: If $\nabla f(\theta_k) \neq 0$, then $-z_k$ is a descent direction for all sufficiently small ε_k, δ_k.

Ingredient 2: Sufficient Decrease with Inexact Gradients

Q: How to choose α_k to get sufficient decrease?

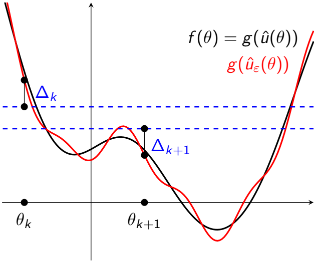
$$f(\theta_{k+1}) + \eta\alpha_k\|z_k\|^2 \leq f(\theta_k)$$

Ingredient 2: Sufficient Decrease with Inexact Gradients

Q: How to choose α_k to get sufficient decrease?

$$f(\theta_{k+1}) + \eta\alpha_k \|z_k\|^2 \leq f(\theta_k)$$

- $g(\hat{u}_{k+1}) + \Delta_{k+1} \geq f(\theta_{k+1})$
- $g(\hat{u}_k) - \Delta_k \leq f(\theta_k)$
- $\Delta_k := \|\nabla g(\hat{u}_k)\|\varepsilon_k + \frac{L_{\nabla g}}{2}\varepsilon_k^2$



Thm: Let $\nabla f(\theta_k) \neq 0$ and $\varepsilon_k, \varepsilon_{k+1} > 0$ be small enough. Then there exists $\alpha_k > 0$, such that

$$g(\hat{u}_{k+1}) + \Delta_k + \Delta_{k+1} + \eta\alpha_k \|z_k\|^2 \leq g(\hat{u}_k),$$

which implies sufficient decrease.

Method of Adaptive Inexact Descent (MAID)

One iteration:

1) Compute inexact gradient z_k (possibly reducing ε_k, δ_k)
2) Attempt backtracking to compute α_k; if failed, go to step 1) with smaller ε_k, δ_k
3) Update estimate: $\theta_{k+1} = \theta_k - \alpha_k z_k$
4) Increase accuracies $\varepsilon_{k+1}, \delta_{k+1}$ and initial step size α_{k+1}

Thm: If $\nabla f(\theta_k) \neq 0$, then MAID updates θ_k in finite time.

Thm: Let f be bounded below. Then MAID's iterates θ_k satisfy

$$\|\nabla f(\theta_k)\| \to 0.$$

Numerical Results

TV denoising: MAID vs DFO-LS (2 parameters)

$$h(u,\theta) = \frac{1}{2}\|u - b\|^2 + \underbrace{e^{\theta[1]} \sum_i \sqrt{|\nabla_1 u_i|^2 + |\nabla_2 u_i|^2 + (e^{\theta[2]})^2}}_{\text{smoothed TV}}$$

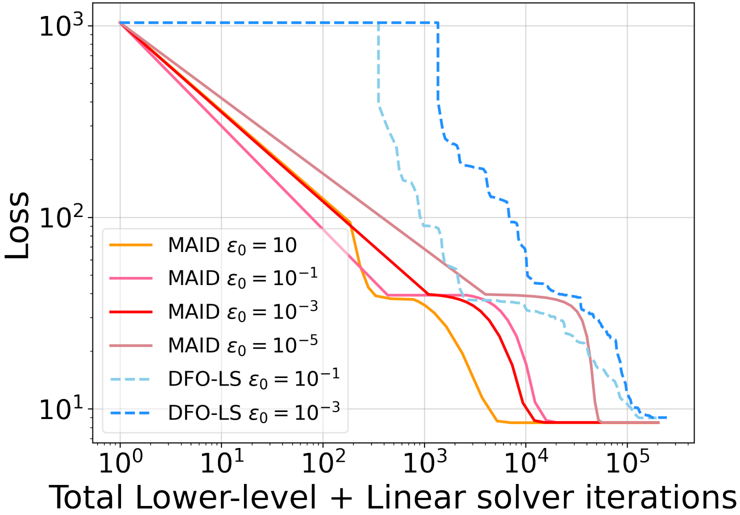
Noisy, PSNR=20.0

DFO-LS, 26.7

MAID, 26.9

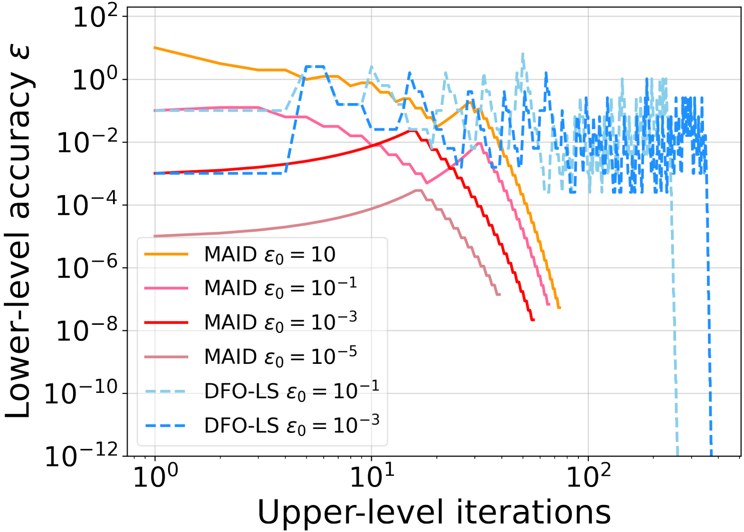
- similar image quality

TV denoising: MAID vs DFO-LS (2 parameters)



- Robustness to initial accuracy ε_0
- MAID particularly initially faster

TV denoising: MAID vs DFO-LS (2 parameters)

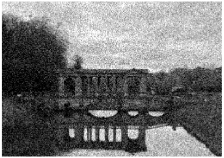


- MAID adapts accuracy, converge to same values in similar trend

FoE Denoising: MAID ($\approx$ 2.5k parameters)

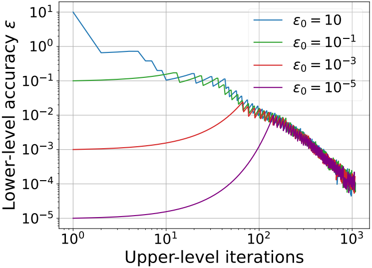
$$h(u,\theta) = \frac{1}{2}\|u-b\|^2 + \mathcal{R}(u)$$

$$\mathcal{R}(u) = \sum_{k=1}^{K} \lambda_k \phi_{\gamma_k}(\kappa_k * u)$$



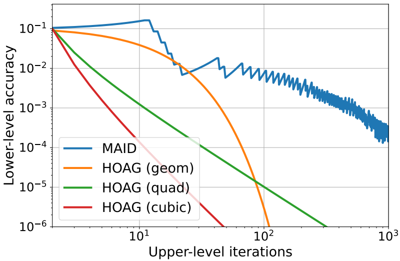
Noisy, PSNR=20.3

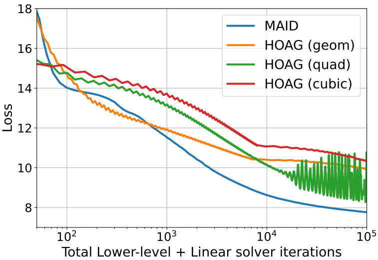
MAID, 29.7



- "It works": learns denoising
- MAID automatically tunes best accuracy schedule

FoE Denoising: MAID vs HOAG





- accuracy schedule important; here slower decay better
- faster convergence, robust

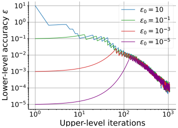
HOAG[2], 28.8

MAID, 29.7

Conclusions & Future Work

Conclusions

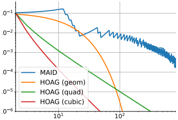
- Bilevel learning: supervised learning for variational regularization; computationally very hard
- Accuracy in the optimization algorithm is important; stability and efficiency
- MAID is a first-order algorithm with adaptive accuracies for descent and backtracking
- High-dimensional parametrizations can be learned; e.g., FoE, ICNN (a few thousand parameters)



Future work

- Other models, e.g., inexact forward operator
- Smart accuracy schedule; disentangle accuracies ε, δ and step size α
- Stochastic variants for training from large data

Salehi et al. '25



Inexact Primal-Dual

Bogensperger et al. JMIV '25

Inexact Primal-Dual for Bilevel learning

Upper level: $\min_{\theta}\{\mathcal{L}(\theta) := \ell_1(\hat{x}(\theta)) + \ell_2(\hat{y}(\theta))\}$

Lower level: $\hat{x}(\theta), \hat{y}(\theta) := \arg\min_{x}\max_{y}\{\langle\theta x, y\rangle + g(x) - f^*(y)\}$

If g and f^* are regular enough, gradients can be computed via

$$\nabla\mathcal{L}(\theta) = \hat{y}(\theta) \otimes \hat{X}(\theta) + \hat{Y}(\theta) \otimes \hat{x}(\theta)$$

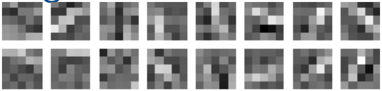
where $\hat{X}(\theta)$, $\hat{Y}(\theta)$ solve another saddle-point problem (this time quadratic!) involving $\nabla^2 g(\hat{x}(\theta))$, $\nabla^2 f^*(\hat{y}(\theta))$, $\nabla\ell_1(\hat{x}(\theta))$ and $\nabla\ell_2(\hat{x}(\theta))$

Idea: this is of the same form as for MAID.

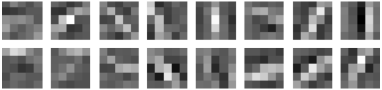
Problems of this form:

- learning discretisations of TV Chambolle and Pock '21
- training ICNNs after primal-dual reformulation Wong et al. '24

Learning TV discretisations



non-adaptive



adaptive

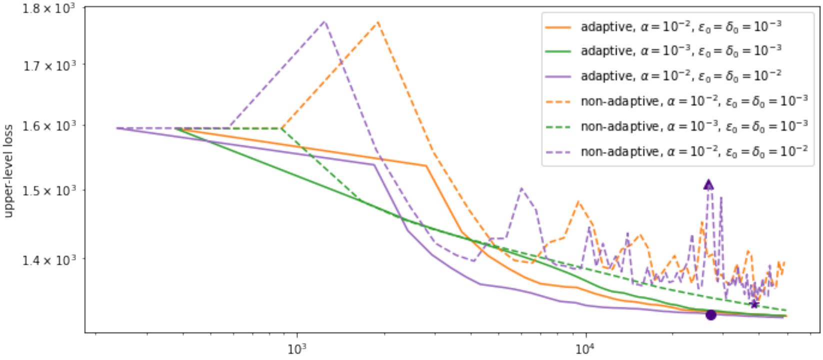
standard TV
PSNR = 25.82 dB

non-adaptive
PSNR = 26.63 dB

adaptive
PSNR = 26.90 dB

- similar reconstructions

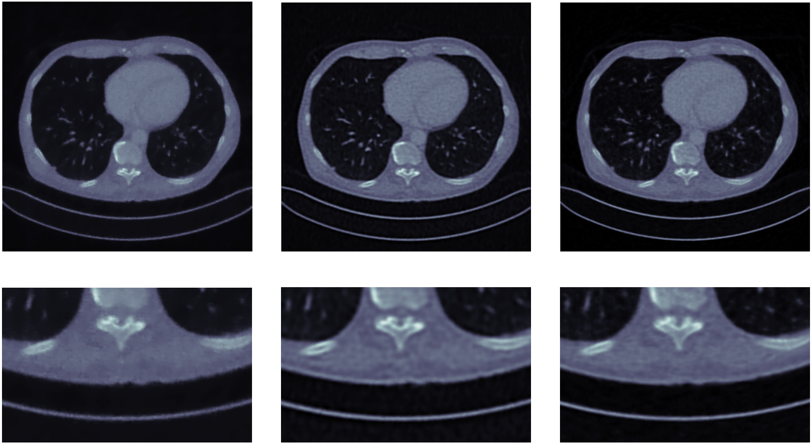
Learning TV discretisations II



compute

- results still depend on parameters
- sensitivity much reduced

CT Reconstruction



ICNN-AR, PSNR=29.3 | ICNN-Bilevel, 31.4 | LPD, 34.2

- much better performance with end-to-end learning

Mukherjee et al. '24, Adler and Öktem '18