

Chapter 3

Bilevel Optimization

Convex Optimization for Machine Learning & Computer Vision
SS 2017



Tao Wu
Thomas Möllenhoff
Emanuel Laude

Computer Vision Group
Department of Informatics
TU München



- Arise of hyperparameters in variational methods.
- Bilevel optimization.
- Implicit function from lower-level problem.
- Adjoint-based solver.

Training regularization parameters for denoising

A denoising oracle $z \mapsto u$:

$$u \in \arg \min_{\hat{u}: \Omega \rightarrow \mathbb{R}} \sum_{l=1}^L \alpha_l \|K_l \hat{u}\|_1 + \frac{1}{2} \|\hat{u} - z\|^2.$$

- $\{K_l\}$ are given analysis filters (typically overcomplete).
- The regularization weights $\{\alpha_l\}$ are hyperparameters.



Figure: BSDS500 dataset. Left: clean data $\{\bar{u}_i\}$; Right: noisy data $\{z_i\}$.

$$\min_{\{\alpha_l\}} J(\{u_i\}) = \frac{1}{N} \sum_{i=1}^N \|u_i - \bar{u}_i\|^2$$

$$\text{s.t. } u_i \in \arg \min_{\hat{u}} \sum_{l=1}^L \alpha_l \|K_l \hat{u}\|_1 + \frac{1}{2} \|\hat{u} - z_i\|^2.$$

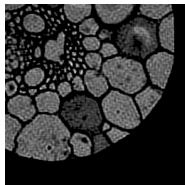


Calibrating blurring kernel

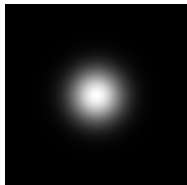
A deblurring oracle $z \mapsto u$:

$$u \in \arg \min_{\hat{u}: \Omega \rightarrow \mathbb{R}} \alpha \|\nabla \hat{u}\|_1 + \frac{1}{2} \|h \star \hat{u} - z\|^2.$$

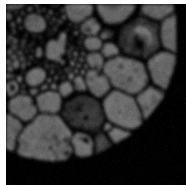
- For simplicity, α and ∇ are fixed.
- \star denotes 2D convolution: $(h \star u)_{i,j} = \sum_{i',j'} h_{i-i',j-j'} u_{i',j'}$.
- The blurring kernel h is a hyperparameter.



clean image \bar{u}_i



blurring kernel h



blurred image z_i

$$\min_{\{\alpha_j\}} J(\{u_i\}, h) = \frac{1}{N} \sum_{i=1}^N \|u_i - \bar{u}_i\|^2 + R(h)$$

$$\text{s.t. } u_i \in \arg \min_{\hat{u}} \alpha \|\nabla \hat{u}\|_1 + \frac{1}{2} \|h \star \hat{u} - z_i\|^2.$$

- The regularization term R enforces structures on h .

- A generic **bilevel optimization** model:

$$\begin{aligned} \min_{u, h} J(u, h) \\ \text{s.t. } u \in \arg \min_{\hat{u}} H(\hat{u}, h). \end{aligned}$$

- The **lower-level** problem is an oracle / classifier / predictor (on u), which is dependent on hyperparameter h .
- The **upper-level** problem is a supervisor, which trains the lower-level model (i.e. hyperparameter h).





- A generic **bilevel optimization** model:

$$\begin{aligned} \min_{u, h} J(u, h) \\ \text{s.t. } u \in \arg \min_{\hat{u}} H(\hat{u}, h). \end{aligned}$$

- The **lower-level** problem is an oracle / classifier / predictor (on u), which is dependent on hyperparameter h .
- The **upper-level** problem is a supervisor, which trains the lower-level model (i.e. hyperparameter h).
- Hyperparameter h may arise from:
 - Parameters / freedoms in the oracle (e.g. regularization weights).
 - Unknowns / uncertainties from modeling (e.g. blurring kernel).



- A generic **bilevel optimization** model:

$$\begin{aligned} \min_{u, h} J(u, h) \\ \text{s.t. } u \in \arg \min_{\hat{u}} H(\hat{u}, h). \end{aligned}$$

- The **lower-level** problem is an oracle / classifier / predictor (on u), which is dependent on hyperparameter h .
- The **upper-level** problem is a supervisor, which trains the lower-level model (i.e. hyperparameter h).
- Hyperparameter h may arise from:
 - Parameters / freedoms in the oracle (e.g. regularization weights).
 - Unknowns / uncertainties from modeling (e.g. blurring kernel).
- An **implicit function approach**:
 - Ideally, lower-level problem \Leftrightarrow **implicit function** $h \mapsto u$.
 - In this scenario, bilevel problem \rightsquigarrow reduced problem:

$$\min_h J(u(h), h).$$



$$\begin{aligned} \min_{u \in \mathbb{R}^n, h \in \mathbb{R}^m} \quad & J(u, h) \\ \text{s.t.} \quad & u \in \arg \min_{\hat{u}} f(\hat{u}, h) + g(\hat{u}), \end{aligned}$$

- The upper-level objective $J : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ is continuously differentiable.
- The lower-level objective is the sum of a differentiable $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ and a non-differentiable function $g : \mathbb{R}^n \rightarrow \mathbb{R}$.
- Assume f and g are proper, convex, lsc in u for any $h \Rightarrow$

$$\begin{aligned} \min_{u \in \mathbb{R}^n, h \in \mathbb{R}^m} \quad & J(u, h) \\ \text{s.t.} \quad & 0 \in F(u, h) + G(u) \end{aligned}$$

with $F(u, h) = \nabla_u f(u, h)$, $G(u) = \partial g(u)$.

- Further assume $F(\cdot, h)$ is μ -strongly monotone for any $h \in \mathbb{R}^m$, i.e.

$$\langle u - v, F(u, h) - F(v, h) \rangle \geq \mu \|u - v\|^2, \quad \forall u, v \in \mathbb{R}^n.$$

Theorem (implicit function under strong monotonicity)

There exists a (single-valued) Lipschitz continuous implicit function $h \mapsto u$ s.t. (u, h) satisfies $0 \in F(u, h) + G(u)$.

Proof: on board.





Implicit function

Theorem (implicit function under strong monotonicity)

There exists a (single-valued) Lipschitz continuous implicit function $h \mapsto u$ s.t. (u, h) satisfies $0 \in F(u, h) + G(u)$.

Proof: on board.

Remarks

Implicit function can be obtained without assuming strong monotonicity of $F(\cdot, h)$.

Theorem (generalized implicit function theorem)

Let (u_0, h_0) be a point that satisfies $0 \in F(u, h) + G(u)$, where F is continuously differentiable and G is a *closed* set-valued map. Assume \exists neighborhoods $U_\xi \ni 0$, $U_u \ni u_0$ s.t.

$$\xi \in U_\xi \mapsto \left\{ u : \xi \in F(u_0, h_0) + \nabla_u F(u_0, h_0)^\top (u - u_0) + G(u) \right\} \cap U_u$$

is a Lipschitz function. Then \exists neighborhoods $V_h \ni h_0$, $V_u \ni u_0$ s.t. $h \in V_h \mapsto u \in V_u$ with $0 \in F(u, h) + G(u)$ is a Lipschitz implicit function.

Sensitivity analysis



- How to perform gradient descent on bilevel optimization?
- Solve the **linearized bilevel problem** for $(\delta u, \delta h)$:

$$\min_{\delta u, \delta h} \nabla_u J(u, h)^\top \delta u + \nabla_h J(u, h)^\top \delta h$$

$$\text{s.t. } (\delta u, -\nabla_u F(u, h)^\top \delta u - \nabla_h F(u, h)^\top \delta h) \in T_{\text{gph } G}(u, -F(u, h)).$$

- The constraint on $(\delta u, \delta h)$ is called the **sensitivity equation**, which is motivated by the characterization:

$$0 \in F(u, h) + G(u) \Leftrightarrow (u, -F(u, h)) \in \text{gph } G.$$

- $\text{gph } G$ is the **graph** of G , i.e.

$$\text{gph } G = \{(u, v) \in \mathbb{R}^n \times \mathbb{R}^n : v \in G(u)\}.$$

- $T_C(\xi)$ is the **tangent cone** of a closed set C at $\xi \in C$, i.e.

$$T_C(\xi) = \left\{ \eta : \exists \{t^k\} \subset \mathbb{R}_+, \{\xi^k\} \subset C \text{ s.t. } t^k \rightarrow 0^+, \frac{\xi^k - \xi}{t^k} \rightarrow \eta \right\}.$$



Sensitivity analysis

- How to perform gradient descent on bilevel optimization?
- Solve the **linearized bilevel problem** for $(\delta u, \delta h)$:

$$\min_{\delta u, \delta h} \nabla_u J(u, h)^\top \delta u + \nabla_h J(u, h)^\top \delta h$$

$$\text{s.t. } (\delta u, -\nabla_u F(u, h)^\top \delta u - \nabla_h F(u, h)^\top \delta h) \in T_{\text{gph } G}(u, -F(u, h)).$$

- The constraint on $(\delta u, \delta h)$ is called the **sensitivity equation**, which is motivated by the characterization:

$$0 \in F(u, h) + G(u) \Leftrightarrow (u, -F(u, h)) \in \text{gph } G.$$

- $\text{gph } G$ is the **graph** of G , i.e.

$$\text{gph } G = \{(u, v) \in \mathbb{R}^n \times \mathbb{R}^n : v \in G(u)\}.$$

- $T_C(\xi)$ is the **tangent cone** of a closed set C at $\xi \in C$, i.e.

$$T_C(\xi) = \left\{ \eta : \exists \{t^k\} \subset \mathbb{R}_+, \{\xi^k\} \subset C \text{ s.t. } t^k \rightarrow 0^+, \frac{\xi^k - \xi}{t^k} \rightarrow \eta \right\}.$$

- Case study: $G(\cdot) = \partial \|\cdot\|_1$ and $F(u, h) = (u - h)/\alpha$.

Adjoint-based solver

- Further assume G is (single-valued) differentiable function.
- The linearized bilevel problem reduces to:

$$\min_{\delta u, \delta h} \nabla_u J(u, h)^\top \delta u + \nabla_h J(u, h)^\top \delta h$$

$$\text{s.t. } 0 = \nabla_u F(u, h)^\top \delta u + \nabla_h F(u, h)^\top \delta h + \nabla G(u)^\top \delta u.$$





- Further assume G is (single-valued) differentiable function.
- The linearized bilevel problem reduces to:

$$\min_{\delta u, \delta h} \nabla_u J(u, h)^\top \delta u + \nabla_h J(u, h)^\top \delta h$$

$$\text{s.t. } 0 = \nabla_u F(u, h)^\top \delta u + \nabla_h F(u, h)^\top \delta h + \nabla G(u)^\top \delta u.$$

- Hence, its solution provides a gradient descent direction:

$$\delta h = -\nabla_h J(u, h) + \nabla_h F(u, h) [\nabla_u F(u, h) + \nabla G(u)]^{-1} \nabla_u J(u, h).$$

Adjoint-based solver



- Further assume G is (single-valued) differentiable function.
- The linearized bilevel problem reduces to:

$$\begin{aligned} \min_{\delta u, \delta h} \quad & \nabla_u J(u, h)^\top \delta u + \nabla_h J(u, h)^\top \delta h \\ \text{s.t.} \quad & 0 = \nabla_u F(u, h)^\top \delta u + \nabla_h F(u, h)^\top \delta h + \nabla G(u)^\top \delta u. \end{aligned}$$

- Hence, its solution provides a gradient descent direction:

$$\delta h = -\nabla_h J(u, h) + \nabla_h F(u, h) [\nabla_u F(u, h) + \nabla G(u)]^{-1} \nabla_u J(u, h).$$

- Let $\frac{\partial u}{\partial h}$ be the Jacobian of the implicit function $h \mapsto u$. Then

$$-[\nabla_u F(u, h) + \nabla G(u)]^{-1} = \left(\frac{\partial u}{\partial h} \right)^\top.$$

For this reason, $-\left[\nabla_u F(u, h) + \nabla G(u)\right]^{-1} \nabla_u J(u, h)$ is called the **adjoint state**.



- Further assume G is (single-valued) differentiable function.
- The linearized bilevel problem reduces to:

$$\min_{\delta u, \delta h} \nabla_u J(u, h)^\top \delta u + \nabla_h J(u, h)^\top \delta h$$

$$\text{s.t. } 0 = \nabla_u F(u, h)^\top \delta u + \nabla_h F(u, h)^\top \delta h + \nabla G(u)^\top \delta u.$$

- Hence, its solution provides a gradient descent direction:

$$\delta h = -\nabla_h J(u, h) + \nabla_h F(u, h) [\nabla_u F(u, h) + \nabla G(u)]^{-1} \nabla_u J(u, h).$$

- Let $\frac{\partial u}{\partial h}$ be the Jacobian of the implicit function $h \mapsto u$. Then

$$-[\nabla_u F(u, h) + \nabla G(u)]^{-1} = \left(\frac{\partial u}{\partial h} \right)^\top.$$

For this reason, $-[\nabla_u F(u, h) + \nabla G(u)]^{-1} \nabla_u J(u, h)$ is called the **adjoint state**.

- To complete one iteration, $h \leftarrow h + \tau \delta h$ for properly chosen step size τ ; update u by solving the lower-level problem.