S&DS 631 Final Project
Introduction to Bilevel Optimization

Keyi Li

2024.12.9

What’s Bilevel Optimization?

- Optimization in a hierarchical structure, where one optimization problem is nested within another

- Mathematical Formulation:

Upper-level problem (Leader):

$$\min F(x, y)$$
$$\text{s.t. } G(x, y) \geq 0,$$
$$y \in S(x) \quad \xrightarrow{S(x) \text{ is the set of optimal solutions}}$$

Lower-level problem (Follower):

$$\min f(x, y)$$
$$\text{s.t. } g(x, y) \geq 0.$$

(Beck and Schmidt, 2021)

Why Bilevel Optimization?

1. Prevalent in Daily Life

   - **Example**: market entry, toll setting
   - **Methods**: Value Function, Optimality condition

2. Application in Machine learning and Signal Processing

   - **Example**: Invariant Representation Learning
   - **Methods**: Implicit Gradient, Gradient Unrolling

Example 1: Market Entry

1. **Assumption on Demand and Price**

   Quantity: $Q = q_l + q_f$

   Price: $P(Q) = a - bQ$,

2. **Decision Sequence**

   Leader first announce the production quantity q_l

   The follower then decides q_f ,responding optimally

3. **Goal: Profit Maximization**

Leader: $$\max_{x_l} \pi_l = \max_{x_l} \left(a - b\left(q_l + q_f^*\right)\right) q_l - c_l \cdot q_l$$

Follower: $$q_f^* = \arg\max_{q_f} \pi_f = \arg\max_{q_f} \left(a - b\left(q_l + q_f\right)\right) q_f - c_f \cdot q_f$$

(Li, 2024).

Example 1: Market Entry - Continued

Notice that q^*_f can be solved in closed form

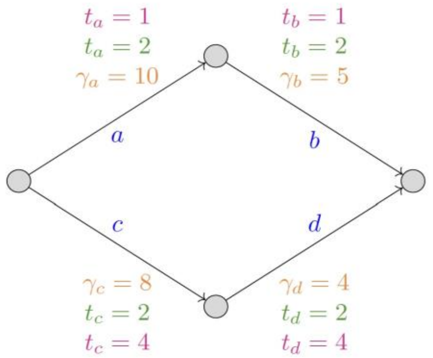
$$\begin{tikzcd}
\frac{\partial \pi_f}{\partial q_f} = 0 \arrow[r] & q_f^* = \frac{a - c_f - b \cdot q_l}{2b} \arrow[d, "\text{substitute in}"] \\
\text{Ordinary Optimization} & \max_{x_l} \pi_l = \max_{x_l} \left(a - b\left(q_l + q_f^*\right)\right) q_l - c_l \cdot q_l \arrow[l]
\end{tikzcd}$$

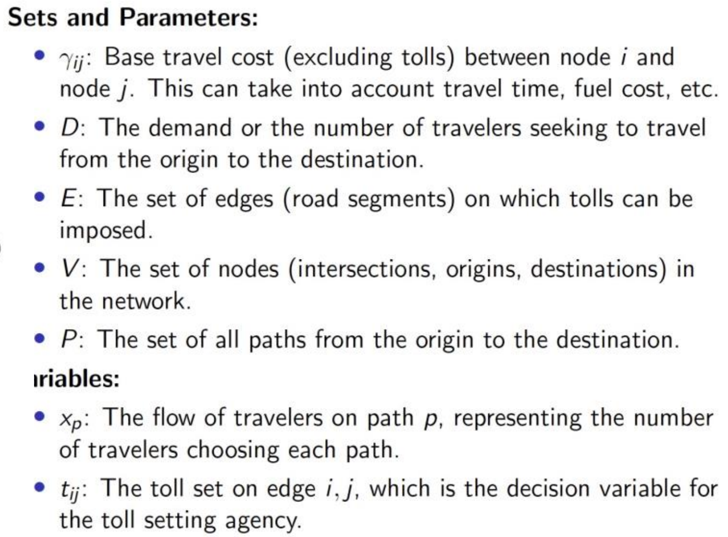
Idea:

- Reformulating Bilevel Optimization into single-level optimization
- Applying the toolkit of optimization principles to solve the problem

❌ solution to lower-level problem cannot be expressed in close form in most cases

Example 2: Toll Setting





Sets and Parameters:

- γ_{ij}: Base travel cost (excluding tolls) between node i and node j. This can take into account travel time, fuel cost, etc.
- D: The demand or the number of travelers seeking to travel from the origin to the destination.
- E: The set of edges (road segments) on which tolls can be imposed.
- V: The set of nodes (intersections, origins, destinations) in the network.
- P: The set of all paths from the origin to the destination.

ıriables:

- x_p: The flow of travelers on path p, representing the number of travelers choosing each path.
- t_{ij}: The toll set on edge i, j, which is the decision variable for the toll setting agency.

Traveler's Goal:
to reach their destination, starting from their origin at minimum costs. In this situation, costs can, e.g., be travel time, toll costs, or a combination of both.

Toll Agency's Goal:
maximize the revenues based on the tolls and the travelers, afterward, minimize traveling costs.

- The toll setting agency is the leader and the travelers are the followers in this setting.
- Leader anticipates the optimal reaction of the followers, whereas the followers' decisions obviously depend on the decision of the leader.

(Li, 2024) and (Beck and Schmidt, 2021).

Example 2: Toll Setting - continued

- Leader (Toll Agency) Objective

$$\max_{\mathbf{t}} \sum_{(i,j)\in E} t_{ij} \cdot \sum_{p\in P:(i,j)\in p} x_p^* \quad \text{s.t. } t_{ij} \geq 0, \quad \forall (i,j) \in \mathbf{E}$$

- Follower (Traveler) Objective

$$x^* = \arg\min_{x} \left\{ \sum_{p\in P} x_p \cdot \left(\sum_{(i,j)\in p} (\gamma_{ij} + t_{ij}) \right) \right\} \quad \text{s.t.} \sum_{p\in P} x_p = D \ , \ x_p \geq 0 \quad \forall p \in P$$

- **Optimistic formulation:** When multiple responses are optimal, the follower takes the decision that work best (e.g the route with the highest toll) for the leader. The reformulated problem is

$$\max_{\mathbf{t},\mathbf{x}^*\in\mathcal{S}(\mathbf{x})} \sum_{(i,j)\in E} t_{ij} \cdot \sum_{p\in P:(i,j)\in p} x_p^* \quad \text{s.t. } t_{ij} \geq 0, \quad \forall (i,j) \in \mathbf{E}$$

- **Pessimistic formulation:** When multiple responses are optimal, the follower takes the decision that work worst (e.g the route with the lowest toll) for the leader. The reformulated problem is

$$\max_{\mathbf{t}} \min_{\mathbf{x}^*\in\mathcal{S}(\mathbf{x})} \sum_{(i,j)\in E} t_{ij} \cdot \sum_{p\in P:(i,j)\in p} x_p^* \quad \text{s.t. } t_{ij} \geq 0, \quad \forall (i,j) \in \mathbf{E}$$

How if we can't write lower form solution in a closed form?

Reformulations based on Value Function

$$\begin{aligned} \min_{x\in X, y\in Y} \quad & F(x,y) \\ \text{s.t.} \quad & G(x,y) \le 0 \\ & y \in S(x) \end{aligned}$$

$$\Big\downarrow$$

$$\varphi(x) := \min_{y\in Y}\{f(x,y) : g(x,y) \le 0\}$$

$$\begin{aligned} \min_{x\in X, y\in Y} \quad & F(x,y) \\ \text{s.t.} \quad & G(x,y) \le 0 \\ & g(x,y) \le 0 \\ & f(x,y) \ge \varphi(x) \end{aligned}$$

However, still hard to optimize!

Reformulations based on Optimality conditions

Idea: Replace $y \in S(x)$ by the optimality conditions of the lower level problem

For LP (strong duality)

- primal feasibility
- dual feasibility
- primal and dual objectives are equal ($c^T x = p^T b$)

For convex NLP (KKT conditions)

- primal feasibility
- dual feasibility
- complementary slackness
- stationarity

(Li, 2024).

LP-LP

$$\begin{aligned} \min_{x,y} \quad & c_x^\top x + c_y^\top y \\ \text{s.t.} \quad & Ax + By \le a \\ & y \in \arg\min_{\bar{y}} \{ d^\top \bar{y} : Cx + D\bar{y} \le b \} \end{aligned}$$

$$\begin{aligned} D^\top \lambda = d, \quad \lambda \ge 0 \quad & \text{(dual feasibility)} \\ Cx + Dy \le b \quad & \text{(primal feasibility)} \\ \lambda_i (C_i x + D_i y - b_i) = 0 \quad \forall i = 1, \dots, \ell \quad & \text{(Complementary slackness)} \end{aligned}$$

General Convex Lower-Level Problems

$$
\begin{aligned}
\min_{x\in X, y\in Y} \quad & F(x,y) \\
\text{s.t.} \quad & G(x,y) \le 0 \\
& y \in \arg\min_{y}\{f(x,y) : g(x,y) \le 0\}
\end{aligned}
$$

$$L(x,y,\lambda) = f(x,y) - \sum_{i=1}^{\ell} \lambda_i g_i(x,y)$$

$$
\begin{aligned}
& \nabla_y L(x,y,\lambda) = 0 \\
& g(x,y) \le 0 \\
& \lambda \ge 0 \\
& \lambda^\top g(x,y) = 0
\end{aligned}
$$

Advanced Setting: Machine Learning

The discussion is limited to a subclass of bilevel optimization problem that follows the following constraints:

(a) the lower-level is unconstrained

(b) the optimal solution to the lower level is a singleton

(c) the objective functions are differentiable.

More specifically, the problem can be abstractly formulated as

$$\min_{\theta} f(\theta, \phi(\theta^*)) \quad \textbf{s.t.} \;\; \phi(\theta^*)) = \arg\min g(\theta, \phi)$$

With the assumption that the objective function is differentiable, we can have the following result by applying chain rule:

$$\frac{df(\theta, \phi^*(\theta))}{d\theta} = \nabla_\theta f(\theta, \phi^*(\theta)) + \frac{d\phi^*(\theta)}{d\theta}^\top \nabla_\phi f(\theta, \phi^*(\theta))$$

(Zhang et al., 2023a).

Implicit Gradient

Idea: Estimate. $\frac{d\phi^*(\theta)}{d\theta}$ using first-order stationary condition

Assuming that g is second-order differentiable, by taking derivative with respect to θ:

$$\nabla_\phi g(\theta, \phi^*(\theta)) = \mathbf{0}. \qquad \frac{d\phi^*(\theta)}{d\theta} = -\nabla^2_{\theta,\phi} g(\theta, \phi^*(\theta)) \boxed{\nabla^2_{\phi,\phi} g(\theta, \phi^*(\theta))^{-1}}$$

Computationally expensive! Needs approximation!

Conjugate Gradient Method

WoodFisher Approximation

Approximation Methods

Denote $\nabla^2_{\phi,\phi} g(\theta_t, \phi(\theta_t))^{-1} \nabla_\phi f(\theta_t, \phi(\theta_t))$ as $\mathbf{H}^{-1}\mathbf{g}$.

Conjugate Gradient Method:

$\mathbf{H}^{-1}\mathbf{g}$ is estimated by solving a quadratic program :

$$\min_x \frac{1}{2} x^T \mathbf{H} x - g^T x$$

- A typical first-order gradient descent algorithm can be used to solve for the estimate.
- However, the convergence rate of CG depends on the smallest singular value of Hessian. An ill-conditioned lower level problem can result in slow convergence.

Approximation Methods

Denote $\nabla^2_{\phi,\phi} g(\theta_t, \phi(\theta_t))^{-1} \nabla_\phi f(\theta_t, \phi(\theta_t))$ as $\mathbf{H}^{-1}\mathbf{g}$.

WoodFihser:

use low-rank approximation to simplify the estimation of the Hessian

$$\mathbf{H} \approx \mathbf{v}\mathbf{v}^T + \gamma \mathbf{I}$$

$$\downarrow$$

Using Woodfisher matrix identity, we have

$$\mathbf{H}^{-1}\mathbf{v} \approx \gamma^{-1}\mathbf{v} + \frac{\gamma^{-2}\mathbf{v}\mathbf{v}^\top}{1 + \gamma^{-1}\mathbf{v}^\top\mathbf{v}}$$

Gradient Unrolling

Unlike IF-based methods, GU-based methods uses an intermediary phase: an unrolled lower-level optimizer, to bridges the gap between the lower-level solution and upper-level optimization. Through this structure, automatic differentiation (AD) is used to calculate the gradients with respect to the upper-level optimization variable θ.

Specifically, the GU-based approach approximates the lower-level optimal solution $\phi^*(\theta)$ by running a given algorithm for a fixed number of iterations, and then inserting the entire trajectory into the upper-level objective.

As an abstract example, for the t^{th} iteration, the K step optimal solution is defined as $\widetilde{\phi}(\theta_t) := \phi_K = q(\theta_t, q(\theta_t, \cdots, q(\theta_t, \phi_0)))$, and will be passed to upper level objective. AD will then be used to calculate the gradient for upper-level problem

$$\widetilde{\nabla} f(\theta_t, \widetilde{\phi}(\theta_t)) := \frac{df(\theta_t, q(\theta_t, q(\theta_t, \cdots, q(\theta_t, \phi_0))))}{d\theta}$$

will be used by standard gradient descent algorithm to update θ

Application in Invariant Representation Learning

Bilevel optimization can be used to improve the domain generalizability of machine learning model. Training model using Invariant risk minimization (IRM) (Arjovsky et al., 2020) can improve model robustness to distribution shifts.

representation learning backbone

$$\min_\theta \sum_{i=1}^{E} \ell_i(\phi^*(\theta) \circ \theta) \text{ s.t. } \phi^*(\theta) \in \arg\min_\phi \ell_i(\phi \circ \theta)$$

prediction head

$$\Downarrow$$

$$\min_\theta \sum_{i=1}^{E} [\ell_i(\theta) + \gamma \|\nabla_{w|w=1.0} \ell_i(w \circ \theta)\|_2^2]$$

(Zhang et al., 2023a).

Invariant Representation Learning: Experiment

Setup:

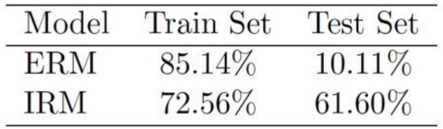
- The MNIST dataset was randomly partitioned into three equal-sized environments: two for training and one for testing.
- Each image underwent a systematic transformation process:
    - binarized the labels by assigning y = 0 to digits 0-4 and y = 1 to digits 5-9. These labels were subject to random noise, with a 25% probability of being flipped.
    - Based on the binary label, images were colored either red or green, depending on which environment they were in.
    - The correlation between colors and labels varied across environments through an environment-specific color-flipping probability e.g. 10% for the first training environment, 20% for the second training environment, and 90% for the test environment

This design created a challenging scenario where the model must learn to rely on genuine features rather than spurious color correlations to assign prediction

Invariant Representation Learning: Experiment

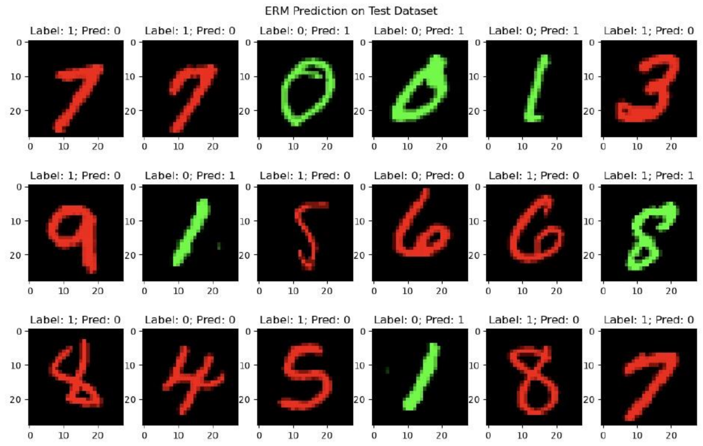
Model: 2 models, the same CNN structure, but trained with different loss functions

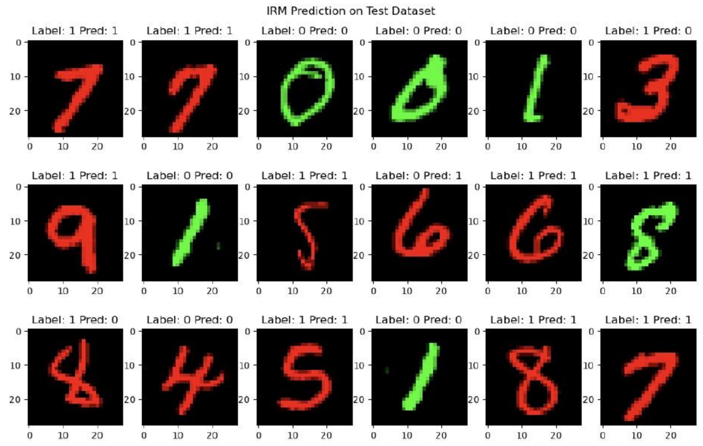
- ERM (standard loss)
- IRM (loss approximating a bilevel-optimization objective)



Model	Train Set	Test Set
ERM	85.14%	10.11%
IRM	72.56%	61.60%

Table 1: Performance comparison between ERM and IRM





Summary

Bilevel Optimization is a class of problems that performs optimization in a hierarchical structure, where one optimization problem is nested within another.

Common ways to solve bilevel optimization:

- **Classical setting**
  - Optimality Condition (e.g Duality, KKT condition)
- **Advanced setting (machine learning, large scale model)**
  - Implicit Function (e.g conjugate function method, WoodFisher methods)
  - Gradient Unrolling

Reference

1. Martin Arjovsky, LÅLeon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization, 2020

2. Yasmine Beck and Martin Schmidt. A gentle and incomplete introduction to bilevel optimization. Technical report, Trier University, 2021.

3. Reiichiro Kano. invariant-risk-minimization, 2019.

4. Can Li. Lecture 24: Bilevel optimization. Lecture Notes in ChE 597: Computational Optimization, 2024.

5. Yihua Zhang, Prashant Khanduri, Ioannis Tsaknakis, Yuguang Yao, Mingyi Hong, and Sijia Liu. An introduction to bi-level optimization: Foundations and applications in signal processing and machine learning. arXiv preprint arXiv:2308.00788, 2023a.