# Functional Bilevel Optimization for Machine Learning

Julien Mairal

Univ. Grenoble-Alpes, Inria

# Collaborators

- I. Petrulionyte, J. Mairal and M. Arbel. Functional Bilevel Optimization for Machine Learning. *arXiv:2403.20233*. 2024.

Ieva Petrulionyte

Michael Arbel

# Bilevel optimization problems

$$\min_{\omega \in \Omega} L_{\mathsf{outer}}(\omega, \theta^\star_\omega) \qquad \text{s.t.} \qquad \theta^\star_\omega = \arg\min_{\theta \in \Theta} L_{\mathsf{inner}}(\omega, \theta).$$

- Introduced in game theory by von Stackelberg, 1934. Obviously, such a definition requires a **unique inner solution** for all outer parameter $\omega$ (to be discussed later).

# Bilevel optimization problems

$$\min_{\omega \in \Omega} L_{\mathsf{outer}}(\omega, \theta^\star_\omega) \qquad \text{s.t.} \qquad \theta^\star_\omega = \arg\min_{\theta \in \Theta} L_{\mathsf{inner}}(\omega, \theta).$$

- Introduced in game theory by von Stackelberg, 1934. Obviously, such a definition requires a **unique inner solution** for all outer parameter $\omega$ (to be discussed later).

A very natural formulation for model selection in machine learning, where

- $\theta$ represents **model parameters**, and $\omega$ **hyper-parameters**.
- $L_{\mathsf{inner}}$ is a regularized empirical risk on training data, whereas $L_{\mathsf{outer}}$ measures the fit of model $\theta^\star_\omega$ on validation data.

# Early occurences in machine learning

$$\min_{\omega \in \Omega} L_{\text{outer}}(\omega, \theta^\star_\omega) \qquad \text{s.t.} \qquad \theta^\star_\omega = \arg\min_{\theta \in \Theta} L_{\text{inner}}(\omega, \theta).$$

- Introduced in machine learning by Bennett et al. [2006]:

## Model Selection via Bilevel Optimization

Kristin P. Bennett, Jing Hu, Xiaoyun Ji, Gautam Kunapuli, and Jong-Shi Pang

***Abstract*— A key step in many statistical learning methods used in machine learning involves solving a convex optimization problem containing one or more hyper-parameters that must be selected by the users. While cross validation is a commonly employed and widely accepted method for selecting these parameters, its implementation by a grid-search procedure in the parameter space effectively limits the desirable number**

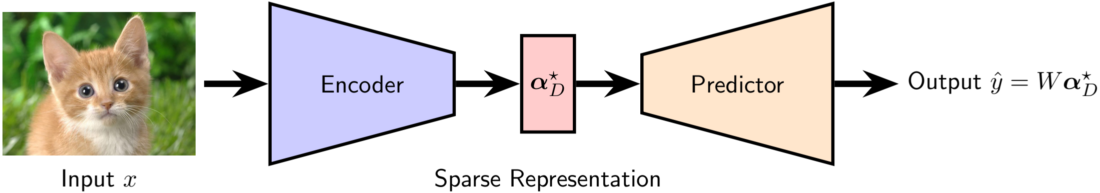
are pervasive in data analysis, e.g., they arise frequently in feature selection [16], [2], kernel construction [19], [22], and multitask learning [4], [10]. For such high-dimensional problems, greedy strategies such as stepwise regression, backward elimination, filter methods, or genetic algorithms are used. Yet, these heuristic methods, including grid search,

# Early occurences in machine learning: self-advertisement

Task-driven dictionary learning formulation [Mairal et al., 2010]:

$$\min_{W,D} \mathbb{E}_{(y,x)}[\ell(y, W\boldsymbol{\alpha}^\star_D(x))]$$

$$\text{s.t.} \quad \boldsymbol{\alpha}^\star_D(x) = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\|x - D\boldsymbol{\alpha}\|^2 + \lambda\|\boldsymbol{\alpha}\|_1 + \frac{\gamma}{2}\|\boldsymbol{\alpha}\|^2.$$



# Early occurences in machine learning: self-advertisement

Task-driven dictionary learning formulation [Mairal et al., 2010]:

$$\min_{W,D} \mathbb{E}_{(y,x)}[\ell(y, W\boldsymbol{\alpha}^\star_D(x))]$$

$$\text{s.t.} \quad \boldsymbol{\alpha}^\star_D(x) = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\|x - D\boldsymbol{\alpha}\|^2 + \lambda\|\boldsymbol{\alpha}\|_1 + \frac{\gamma}{2}\|\boldsymbol{\alpha}\|^2.$$

- derives implicit differentiation for the Lasso/Elastic-Net problem.
- can be seen as **backpropagation rules for sparse coding**.
- operates at the patch level.

# More recent instances in machine learning

Since 2019, more and more applications:

- **hyper-parameter tuning** [Feurer and Hutter, 2019, Lorraine et al., 2019, Franceschi et al., 2017];
- **meta-learning** [Bertinetto et al., 2019];
- **reinforcement learning** [Hong et al., 2023, Liu et al., 2021, Nikishin et al., 2022];
- **inverse problems** (see previous talk), [Holler et al., 2018];
- **invariant risk minimization** [Arjovsky et al., 2019, Ahuja et al., 2020].
- **automatic data augmentation** [Li et al., 2020, Marrie et al., 2023].
- . . . .

## Basic theory
## from the "well-defined" (strongly convex) world

# The workhorse: implicit differentiation

$$\min_{\omega \in \Omega} \mathcal{L}(\omega) := L_{\text{outer}}(\omega, \theta^\star_\omega) \qquad \text{s.t.} \qquad \theta^\star_\omega = \arg\min_{\theta \in \Theta} L_{\text{inner}}(\omega, \theta).$$

**Assumptions**:

- $\Theta = \mathbb{R}^p$ and $\Omega = \mathbb{R}^q$.
- $L_{\text{inner}}$ is twice differentiable and strongly convex with respect to $\theta$.
- $L_{\text{outer}}$ is differentiable.

**Computing the derivative of $\mathcal{L}$:**

$$\nabla \mathcal{L}(\omega) = \partial_\omega L_{\text{outer}}(\omega, \theta^\star_\omega) + [\partial_\omega \theta^\star_\omega]^\top \partial_\theta L_{\text{outer}}(\omega, \theta^\star_\omega),$$

with

$$\partial_\theta L_{\text{inner}}(\omega, \theta^\star_\omega) = 0.$$

# The workhorse: implicit differentiation

$$\min_{\omega\in\Omega} \mathcal{L}(\omega) := L_{\text{outer}}(\omega, \theta^\star_\omega) \qquad \text{s.t.} \qquad \theta^\star_\omega = \arg\min_{\theta\in\Theta} L_{\text{inner}}(\omega, \theta).$$

**Assumptions**:

- $\Theta = \mathbb{R}^p$ and $\Omega = \mathbb{R}^q$.
- $L_{\text{inner}}$ is twice differentiable and strongly convex with respect to $\theta$.
- $L_{\text{outer}}$ is differentiable.

**Computing the derivative of $\mathcal{L}$:**

$$\nabla\mathcal{L}(\omega) = \partial_\omega L_{\text{outer}}(\omega, \theta^\star_\omega) + [\partial_\omega \theta^\star_\omega]^\top \partial_\theta L_{\text{outer}}(\omega, \theta^\star_\omega),$$

with

$$\partial_{\omega,\theta} L_{\text{inner}}(\omega, \theta^\star_\omega) + [\partial_\omega \theta^\star_\omega]^\top \partial^2_\theta L_{\text{inner}}(\omega, \theta^\star_\omega) = 0.$$

# The workhorse: implicit differentiation

$$\min_{\omega\in\Omega} \mathcal{L}(\omega) := L_{\text{outer}}(\omega, \theta^\star_\omega) \qquad \text{s.t.} \qquad \theta^\star_\omega = \arg\min_{\theta\in\Theta} L_{\text{inner}}(\omega, \theta).$$

**Assumptions**:

- $\Theta = \mathbb{R}^p$ and $\Omega = \mathbb{R}^q$.
- $L_{\text{inner}}$ is twice differentiable and strongly convex with respect to $\theta$.
- $L_{\text{outer}}$ is differentiable.

**Computing the derivative of $\mathcal{L}$:**

$$\nabla\mathcal{L}(\omega) = \partial_\omega L_{\text{outer}}(\omega, \theta^\star_\omega) + \partial_{\omega,\theta} L_{\text{inner}}(\omega, \theta^\star_\omega) a^\star_\omega$$
$$\text{where} \qquad a^\star_\omega = -\partial^2_\theta L_{\text{inner}}(\omega, \theta^\star_\omega)^{-1} \partial_\theta L_{\text{outer}}(\omega, \theta^\star_\omega).$$

# Recap

There are **three** actors:

- An **inner-loop**:

$$\theta^\star_\omega = \arg\min_{\theta\in\Theta} L_{\mathsf{inner}}(\omega, \theta).$$

- An **outer-loop**:

$$\min_{\omega\in\Omega} \mathcal{L}(\omega) = L_{\mathsf{outer}}(\omega, \theta^\star_\omega).$$

- A **linear system**: find $a^\star_\omega$ such that

$$\partial_\theta^2 L_{\mathsf{inner}}(\omega, \theta^\star_\omega) a^\star_\omega + \partial_\theta L_{\mathsf{outer}}(\omega, \theta^\star_\omega) = 0,$$

# Recap

There are **three** actors:

- An **inner-loop**:

$$\theta^\star_\omega = \arg\min_{\theta\in\Theta} L_{\text{inner}}(\omega, \theta).$$

- An **outer-loop**:

$$\min_{\omega\in\Omega} \mathcal{L}(\omega) = L_{\text{outer}}(\omega, \theta^\star_\omega).$$

- A **linear system**: find $a^\star_\omega$ such that

$$\partial^2_\theta L_{\text{inner}}(\omega, \theta^\star_\omega) a^\star_\omega + \partial_\theta L_{\text{outer}}(\omega, \theta^\star_\omega) = 0,$$

and the gradient is:

$$\nabla\mathcal{L}(\omega) = \partial_\omega L_{\text{outer}}(\omega, \theta^\star_\omega) + \partial_{\omega,\theta} L_{\text{inner}}(\omega, \theta^\star_\omega) a^\star_\omega.$$

# Questions/Topics

## Inexact gradients

- Controlling the approximation error, designing approximations: [Ablin et al., 2020, Blondel et al., 2022]. . .

## Dealing with stochastic objectives

- algorithm design and optimal rates: [Ghadimi and Wang, 2018, Yang et al., 2021, Arbel and Mairal, 2022a]. . .
- variance reduction for deterministic finite sums: [Dagréou et al., 2022].

## Exotic implicit differentiation

- non-smooth implicit differentiation [Bolte et al., 2021].

# Dealing with non-convex inner problems

# An ambiguous definition

$$\min_{\omega \in \Omega} L_{\mathsf{outer}}(\omega, \theta^\star_\omega) \qquad \text{s.t.} \qquad \theta^\star_\omega \in \operatorname*{arg\,min}_{\theta \in \Theta} L_{\mathsf{inner}}(\omega, \theta).$$

# An ambiguous definition

$$\min_{\omega\in\Omega} L_{\text{outer}}(\omega, \theta^\star_\omega) \qquad \text{s.t.} \qquad \theta^\star_\omega \in \operatorname*{arg\,min}_{\theta\in\Theta} L_{\text{inner}}(\omega, \theta).$$

We need a mechanism for **selecting** $\theta^\star_\omega$. For example,

## Optimistic formulation

$$\min_{\omega\in\Omega}\min_{\theta\in\Theta} L_{\text{outer}}(\omega, \theta) \qquad \text{s.t.} \qquad \theta \in \operatorname*{arg\,min}_{\theta\in\Theta} L_{\text{inner}}(\omega, \theta).$$

## Pessimistic formulation

$$\min_{\omega\in\Omega}\max_{\theta\in\Theta} L_{\text{outer}}(\omega, \theta) \qquad \text{s.t.} \qquad \theta \in \operatorname*{arg\,min}_{\theta\in\Theta} L_{\text{inner}}(\omega, \theta).$$

# An ambiguous definition

$$\min_{\omega\in\Omega} L_{\text{outer}}(\omega, \theta^\star_\omega) \qquad \text{s.t.} \qquad \theta^\star_\omega \in \operatorname*{arg\,min}_{\theta\in\Theta} L_{\text{inner}}(\omega, \theta).$$

We need a mechanism for **selecting** $\theta^\star_\omega$. For example,

## Optimistic formulation

$$\min_{\omega\in\Omega} \min_{\theta\in\Theta} L_{\text{outer}}(\omega, \theta) \qquad \text{s.t.} \qquad \theta \in \operatorname*{arg\,min}_{\theta\in\Theta} L_{\text{inner}}(\omega, \theta).$$

## Pessimistic formulation

$$\min_{\omega\in\Omega} \max_{\theta\in\Theta} L_{\text{outer}}(\omega, \theta) \qquad \text{s.t.} \qquad \theta \in \operatorname*{arg\,min}_{\theta\in\Theta} L_{\text{inner}}(\omega, \theta).$$

**Problems:** may be meaningless for model selection in machine learning, especially with overparametrized deep networks.

# A first solution: Bilevel Games with Selection [Arbel and Mairal, 2022b]

$$\min_{\omega\in\Omega} \mathcal{L}_{\varphi}(\omega,\theta) := L_{\text{outer}}(\omega,\varphi(\omega,\theta)), \qquad \min_{\theta\in\Theta} L_{\text{inner}}(\omega,\theta).$$

**Definition of selection maps $\varphi$:**

- **Criticality:** $\varphi(\omega,\theta)$ is a critical point of $L_{\text{inner}}(\omega,.)$.
- **Consistency:** if $\theta$ is a critical point of $L_{\text{inner}}(\omega,.)$, $\varphi(\omega,\theta)=\theta$.

**Goal:** Finding an equilibrium point $(\omega^\star,\theta^\star)$ such that

$$\partial_\omega \mathcal{L}_{\varphi}(\omega^\star,\theta^\star)=0 \quad \text{and} \quad \partial_\theta L_{\text{inner}}(\omega^\star,\theta^\star)=0.$$

# A first solution: Bilevel Games with Selection [Arbel and Mairal, 2022b]

$$\min_{\omega\in\Omega} \mathcal{L}_{\varphi}(\omega,\theta) := L_{\text{outer}}(\omega,\varphi(\omega,\theta)), \qquad \min_{\theta\in\Theta} L_{\text{inner}}(\omega,\theta).$$

**Definition of selection maps $\varphi$:**

- **Criticality:** $\varphi(\omega,\theta)$ is a critical point of $L_{\text{inner}}(\omega,.)$.
- **Consistency:** if $\theta$ is a critical point of $L_{\text{inner}}(\omega,.)$, $\varphi(\omega,\theta)=\theta$.

**Example:**

- if strongly-convex, $\varphi(\omega,\theta)=\theta^{\star}_{\omega}$ (classical bilevel).
- more interesting: **limit of a gradient flow, initialized at $\theta$**, under (rather strong) geometric assumptions called **parametric Morse-Bott**.

# A first solution: Bilevel Games with Selection [Arbel and Mairal, 2022b]

$$\min_{\omega\in\Omega} \mathcal{L}_\varphi(\omega,\theta) := L_{\text{outer}}(\omega, \varphi(\omega,\theta)), \qquad \min_{\theta\in\Theta} L_{\text{inner}}(\omega,\theta).$$

**Definition of selection maps $\varphi$:**

- **Criticality:** $\varphi(\omega,\theta)$ is a critical point of $L_{\text{inner}}(\omega,.)$.
- **Consistency:** if $\theta$ is a critical point of $L_{\text{inner}}(\omega,.)$, $\varphi(\omega,\theta)=\theta$.

**Example:**

- if strongly-convex, $\varphi(\omega,\theta)=\theta^\star_\omega$ (classical bilevel).
- more interesting: **limit of a gradient flow, initialized at $\theta$**, under (rather strong) geometric assumptions called **parametric Morse-Bott**.

**Consequences:**

- justify iterative differentiation in the non-convex setting with degenerate critical points. Provides a correction for better gradient approximation.

**Go functional!**
**[Petrulionyte, Mairal, and Arbel, 2024]**

# A different point of view, specific to machine learning

$$\min_{\theta \in \Theta} \mathbb{E}[\ell_{\text{inner}}(\omega, h_\theta(x), y)].$$

- A typical inner-loop problem, where $h_\theta$ is a neural network with parameters $\theta$.
- $(y, x)$ represent data pairs in supervised learning.
- $\ell_{\text{inner}}$ is a classical convex loss function including a regularization term.

# A different point of view, specific to machine learning

$$\min_{\theta\in\Theta} \mathbb{E}[\ell_{\mathsf{inner}}(\omega, h_\theta(x), y)].$$

- A typical inner-loop problem, where $h_\theta$ is a neural network with parameters $\theta$.
- $(y, x)$ represent data pairs in supervised learning.
- $\ell_{\mathsf{inner}}$ is a classical convex loss function including a regularization term.

**Functional point of view:** this is an approximate solution of a more general one

$$\min_{h\in\mathcal{H}} \mathbb{E}[\ell_{\mathsf{inner}}(\omega, h(x), y)],$$

where $\mathcal{H}$ is a Hilbert space such as $L^2$. Ex:

$$\min_{h\in\mathcal{H}} \mathbb{E}\left[\|y - h(x)\|^2\right] + \omega\|h\|^2_{\mathcal{H}}.$$

# Why do we care?

$$\min_{\omega \in \Omega} \mathbb{E}[\ell_{\text{outer}}(\omega, h^\star_\omega(x'), y')] \qquad \text{s.t.} \qquad h^\star_\omega = \arg\min_{h \in \mathcal{H}} \mathbb{E}[\ell_{\text{inner}}(\omega, h(x), y)]. \qquad \text{(FBO)}$$

- Strong convexity with respect to $h$ is a **mild assumption**.
- **No ambiguity** to define $h^\star_\omega$.
- Compatible with deep neural networks used for **function approximation**.

# Why do we care?

$$\min_{\omega \in \Omega} \mathbb{E}[\ell_{\text{outer}}(\omega, h^\star_\omega(x'), y')] \qquad \text{s.t.} \qquad h^\star_\omega = \arg\min_{h \in \mathcal{H}} \mathbb{E}[\ell_{\text{inner}}(\omega, h(x), y)]. \qquad \text{(FBO)}$$

- Strong convexity with respect to $h$ is a **mild assumption**.
- **No ambiguity** to define $h^\star_\omega$.
- Compatible with deep neural networks used for **function approximation**.

What is the price to pay?

- Need to develop theory and algorithms for (FBO).
- Differentiability in infinite dimension is . . . tricky.

# Parenthesis: differentiability in infinite dimension

**Fréchet derivative:** Given $F : U \to Y$ where $X, Y$ are Banach spaces and $U$ is an open subset, $F$ is differentiable at $h \in U$ if there exists a bounded linear operator $A : X \to Y$ such that

$$F(h+\varepsilon) = F(h) + A.\varepsilon + o(\varepsilon).$$

# Parenthesis: differentiability in infinite dimension

**Fréchet derivative:** Given $F : U \to Y$ where $X, Y$ are Banach spaces and $U$ is an open subset, $F$ is differentiable at $h \in U$ if there exists a bounded linear operator $A : X \to Y$ such that

$$F(h + \varepsilon) = F(h) + A.\varepsilon + o(\varepsilon).$$

**Good news**: implicit differentiation works for twice Fréchet differentiable functionsm

# Parenthesis: differentiability in infinite dimension

**Fréchet derivative:** Given $F: U \to Y$ where $X, Y$ are Banach spaces and $U$ is an open subset, $F$ is differentiable at $h \in U$ if there exists a bounded linear operator $A: X \to Y$ such that

$$F(h+\varepsilon) = F(h) + A.\varepsilon + o(\varepsilon).$$

**Good news**: implicit differentiation works for twice Fréchet differentiable functionsm

**Is it such a good news?**

# Parenthesis: differentiability in infinite dimension

Consider an objective $F : L^2[0,1] \to \mathbb{R}$ of the form

$$F(h) = \int \ell(h(x)),$$

where $h$ is in $L^2([0,1])$ and assume that $\ell(u) = \sum_{i=0}^{n} a_i u^i$ is a polynomial function with $a_n \neq 0$ and $n > 2$.

# Parenthesis: differentiability in infinite dimension

Consider an objective $F : L^2[0,1] \to \mathbb{R}$ of the form

$$F(h) = \int \ell(h(x)),$$

where $h$ is in $L^2([0,1])$ and assume that $\ell(u) = \sum_{i=0}^n a_i u^i$ is a polynomial function with $a_n \neq 0$ and $n > 2$. Consider $\varepsilon$ in $L^2[0,1]$ such that $\varepsilon(x) = \frac{1}{x^{1/3}}$ (not in $L^3$)

$$\begin{aligned} F(\varepsilon) &= \int_{x=0}^{1} \sum_{i=0}^{n} a_i \frac{1}{x^{i/3}} \\ &= a_0 + \frac{3a_1}{2} + 3a_2 + \left[ a_3 \log(x) + \sum_{i=4}^{n} a_i \frac{3}{(3-i)x^{i/3-1}} \right]_{x=0}^{1} = \operatorname{sign}(a_n)\infty. \end{aligned}$$

**$\ell$ needs to be quadratic!**

# Parenthesis: differentiability in infinite dimension

Intuition why twice Fréchet differentiability is a **very strong** assumption in $L^2$:
Assuming it is the case for $F$ below and $\ell$ is in $C^3$ (not necessarily polynomial).

$$F(h) = \int \ell(h(x)).$$

Then, for **any** $h, \varepsilon$ in $L^2$ (not necessarily in $L^3$)

$$F(h+\varepsilon) = F(h) + \langle \ell' \circ h, \varepsilon \rangle + \frac{1}{2}\langle (\ell'' \circ h)\varepsilon, \varepsilon \rangle + \int_x \frac{1}{2}\int_0^1 (1-t)^2 \ell'''(h(x) + t\varepsilon(x))\varepsilon(x)^3.$$

# Parenthesis: differentiability in infinite dimension

Intuition why twice Fréchet differentiability is a **very strong** assumption in $L^2$:
Assuming it is the case for $F$ below and $\ell$ is in $C^3$ (not necessarily polynomial).

$$F(h) = \int \ell(h(x)).$$

Then, for **any** $h, \varepsilon$ in $L^2$ (not necessarily in $L^3$)

$$F(h+\varepsilon) = F(h) + \langle \ell' \circ h, \varepsilon \rangle + \frac{1}{2}\langle (\ell'' \circ h)\varepsilon, \varepsilon \rangle + \int_x \frac{1}{2}\int_0^1 (1-t)^2 \ell'''(h(x)+t\varepsilon(x))\varepsilon(x)^3.$$

Hard to ensure that the last term is finite for any $h, \varepsilon$, unless $\ell$ is quadratic.

# Parenthesis: differentiability in infinite dimension

Intuition why twice Fréchet differentiability is a **very strong** assumption in $L^2$:
Assuming it is the case for $F$ below and $\ell$ is in $C^3$ (not necessarily polynomial).

$$F(h) = \int \ell(h(x)).$$

Then, for **any** $h, \varepsilon$ in $L^2$ (not necessarily in $L^3$)

$$F(h+\varepsilon) = F(h) + \langle \ell' \circ h, \varepsilon \rangle + \frac{1}{2}\langle (\ell'' \circ h)\varepsilon, \varepsilon \rangle + \int_x \frac{1}{2}\int_0^1 (1-t)^2 \ell'''(h(x) + t\varepsilon(x))\varepsilon(x)^3.$$

Hard to ensure that the last term is finite for any $h, \varepsilon$, unless $\ell$ is quadratic.

**Exercise for Gabriel:** Does twice Fréchet differentiable implies quadratic here? Which assumptions are needed for that to be true? (see Nemirovski and Semenov, 1973).

# Parenthesis: differentiability in infinite dimension

**Fréchet is too strong for the second derivative**, because $L^2$ may contain sequences of "nasty perturbations" (unit ball is not compact).

# Parenthesis: differentiability in infinite dimension

**Fréchet is too strong for the second derivative**, because $L^2$ may contain sequences of "nasty perturbations" (unit ball is not compact).

- **Gâteaux?**: perturbations along fixed directions: not strong enough!

# Parenthesis: differentiability in infinite dimension

**Fréchet is too strong for the second derivative**, because $L^2$ may contain sequences of “nasty perturbations” (unit ball is not compact).

- **Gâteaux?**: perturbations along fixed directions: not strong enough!
- **The solution: Hadamard!** ($\approx$ perturbations in compact sets).
  **Sufficient to derive an implicit differentiation theorem.**

# Computing the gradient

Consider the problem

$$\min_{\omega \in \Omega} \mathcal{L}(\omega) := L_{\text{outer}}(\omega, h^\star_\omega) \qquad \text{s.t.} \qquad h^\star_\omega = \arg\min_{h \in \mathcal{H}} L_{\text{inner}}(\omega, h).$$

Assume

- $L_{\text{outer}}$ is Fréchet differentiable.
- $L_{\text{inner}}$ is $\mu$-strongly convex w.r.t. $h$ and Fréchet differentiable w.r.t. $\omega$.
- $\partial_h L_{\text{inner}}$ is Hadamard differentiable.

Then, $\mathcal{L}$ is differentiable and

$$\nabla \mathcal{L}(\omega) = \nabla_\omega L_{\text{outer}}(\omega, h^\star_\omega) + \nabla_{\omega,h} L_{\text{inner}}(\omega, h^\star_\omega) a^\star_\omega,$$

where

$$a^\star_\omega = \arg\min_{a \in \mathcal{H}} L_{\text{adj}}(\omega, a) := \frac{1}{2} \langle a, \nabla_h^2 L_{\text{inner}}(\omega, h^\star_\omega) a \rangle_{\mathcal{H}} + \langle a, \nabla_h L_{\text{outer}}(\omega, h^\star_\omega) \rangle_{\mathcal{H}}.$$

# 1st ingredient: stochastic approximations

Consider $\mathcal{H}$ to be an $L^2$ space with the previous machine learning objectives, and $\Omega = \mathbb{R}^p$. We still have **three** actors:

- An **inner-loop**:

$$h^\star_\omega = \arg\min_{h \in \mathcal{H}} L_{\mathsf{inner}}(\omega, h).$$

- An **outer-loop**:

$$\min_{\omega \in \Omega} L_{\mathsf{outer}}(\omega, h^\star_\omega).$$

- A **linear system** (quadratic objective in $\mathcal{H}$):

$$a^\star_\omega = \arg\min_{a \in \mathcal{H}} L_{\mathsf{adj}}(\omega, a).$$

# 1st ingredient: stochastic approximations

Consider $\mathcal{H}$ to be an $L^2$ space with the previous machine learning objectives, and $\Omega = \mathbb{R}^p$. We still have **three** actors:

- An **inner-loop**:

$$h_\omega^\star = \arg\min_{h \in \mathcal{H}} \mathbb{E}[\ell_{\text{inner}}(\omega, h(x), y)].$$

- An **outer-loop**:

$$\min_{\omega \in \Omega} \mathbb{E}[\ell_{\text{outer}}(\omega, h_\omega^\star(x'), y')].$$

- A **linear system** (quadratic objective in $\mathcal{H}$):

$$a_\omega^\star = \arg\min_{a \in \mathcal{H}} \frac{1}{2}\mathbb{E}\left[a(x)\partial_2^2 \ell_{\text{inner}}(\omega, h_\omega^\star(x), y)a(x)\right] + \mathbb{E}\left[a(x)\partial_2 \ell_{\text{outer}}(\omega, h_\omega^\star(x'), y')\right].$$

# 1st ingredient: stochastic approximations

Consider $\mathcal{H}$ to be an $L^2$ space with the previous machine learning objectives, and $\Omega = \mathbb{R}^p$. We still have **three** actors:

- An **inner-loop**:

$$h^\star_\omega = \arg\min_{h\in\mathcal{H}} \mathbb{E}[\ell_{\text{inner}}(\omega, h(x), y)].$$

- An **outer-loop**:

$$\min_{\omega\in\Omega} \mathbb{E}[\ell_{\text{outer}}(\omega, h^\star_\omega(x'), y')].$$

- A **linear system** (quadratic objective in $\mathcal{H}$):

$$a^\star_\omega = \arg\min_{a\in\mathcal{H}} \frac{1}{2}\mathbb{E}\left[a(x)\partial_2^2\ell_{\text{inner}}(\omega, h^\star_\omega(x), y)a(x)\right] + \mathbb{E}\left[a(x)\partial_2\ell_{\text{outer}}(\omega, h^\star_\omega(x'), y')\right].$$

**The first ingredient is naturally the use of stochastic approximations.**

# 2nd ingredient: function approximation

Since directly optimizing over $\mathcal{H}$ is too difficult (unless it is an RKHS), we consider a map $\theta : \Theta \to \mathcal{H}$ (*e.g.*, a deep neural network) and optimize over $\Theta$.

- We do that both for $L_{\text{inner}}$ and $L_{\text{adj}}$.
- Optimizing w.r.t. $\theta$ may yield multiple solutions (not a problem).
- Overall algorithm can be seen as **SGD with inexact gradients**.
- The larger the neural network, the better the approximation of the functional bilevel formulation (use **overparametrized** deep neural networks).

# The algorithm

**Algorithm 1** ***FuncID***

**Input:** initial outer, inner, and adjoint parameter $\omega_0$, $\theta_0$, $\xi_0$; warm-start option `WS`.
**for** $n = 0, \ldots, N-1$ **do**
  *# Optional warm-start*
  **if** WS=True **then** $(\theta_0, \xi_0) \leftarrow (\theta_n, \xi_n)$ **end if**
  *# Inner-level optimization*
  $\hat{h}_{\omega_n}, \theta_{n+1} \leftarrow \texttt{InnerOpt}(\omega_n, \theta_0, \mathcal{D}_{in})$
  *# Adjoint optimization*
  $\hat{a}_{\omega_n}, \xi_{n+1} \leftarrow \texttt{AdjointOpt}(\omega_n, \xi_0, \hat{h}_{\omega_n}, \mathcal{D})$
  *# Outer gradient estimation*
  Sample a mini-batch $\mathcal{B} = (\mathcal{B}_{out}, \mathcal{B}_{in})$ from $\mathcal{D} = (\mathcal{D}_{out}, \mathcal{D}_{in})$
  $g_{out} \leftarrow \texttt{TotalGrad}(\omega_n, \hat{h}_{\omega_n}, \hat{a}_{\omega_n}, \mathcal{B})$
  $\omega_{n+1} \leftarrow$ update $\omega_n$ using $g_{out}$;
**end for**

# Applications and experiments

1. instrumental variable regression.
2. model-based reinforcement learning.

# Instrumental variable regression (IV)

## Example courtesy of Arthur Gretton, from his AISTATS'23 keynote

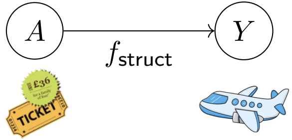
Price tickets $A$; Seats sold $Y$.

$$A \xrightarrow[f_{\text{struct}}]{} Y$$

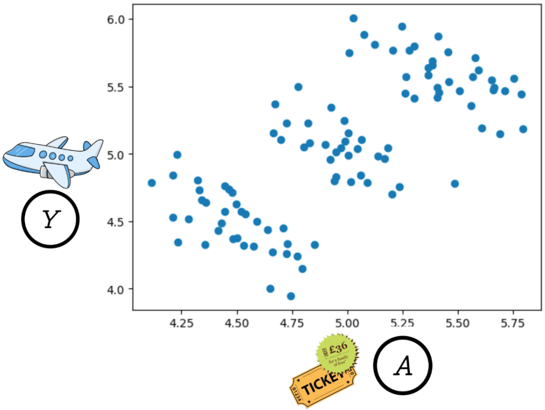

# Instrumental variable regression (IV)

## Example courtesy of Arthur Gretton, from his AISTATS'23 keynote

Price tickets $A$; Seats sold $Y$.

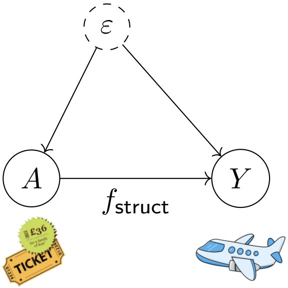



What we observe



# Instrumental variable regression (IV)

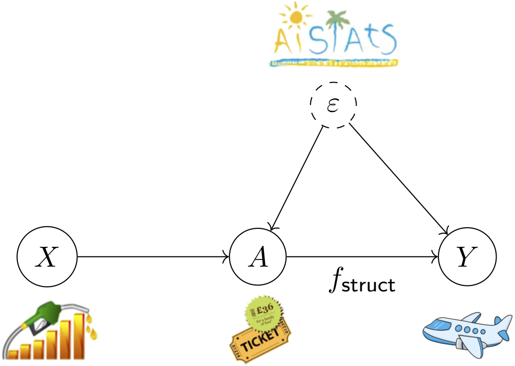
## Example courtesy of Arthur Gretton, from his AISTATS'23 keynote




- We assume $Y = f_{\text{struct}}(A) + \varepsilon$ with $\mathbb{E}[\varepsilon] = 0$ and we want to recover $f_{\text{struct}}$.
- An unobserved counfounder $\varepsilon$ affects both $Y, A$ making direct regression vacuous.

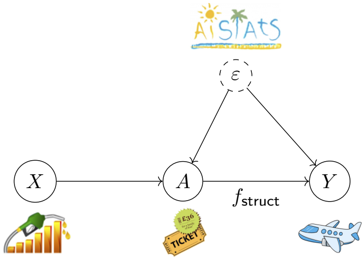
# Instrumental variable regression (IV)

## Example courtesy of Arthur Gretton, from his AISTATS'23 keynote



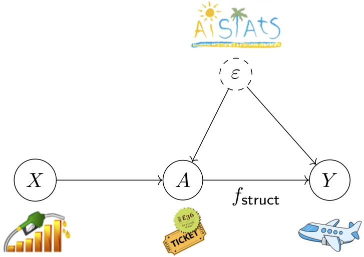

- $X$ is an observed **instrumental variable**, independent of $\varepsilon$, that affects $Y$ through $A$.

# Two-stage least squares regression (2SLS)




- **Instrumental Variable regression** exploits the problem structure to learn $f_{\text{struct}}$.
- Classical approach in econometrics and recent interest in ML [Singh et al., 2019, Xu et al., 2021] with bilevel formulations.
- In practice, we need to find an instrumental variable $X$ that strongly influences $A$ without being affected by $\varepsilon$ (this is hard).

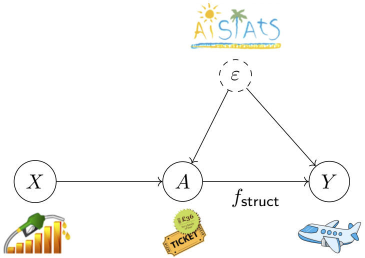
# Two-stage least squares regression (2SLS)




- Given the model $Y = f_{\text{struct}}(A) + \varepsilon$, we have $\mathbb{E}[f_{\text{struct}}(A)|X] = \mathbb{E}[Y|X]$.
- This suggests the regression problem:

$$\min_{\omega \in \Omega} \mathbb{E}\left[\|Y - \mathbb{E}[f_\omega(A)|X]\|^2\right].$$

# Two-stage least squares regression (2SLS)




- This suggests the regression problem:

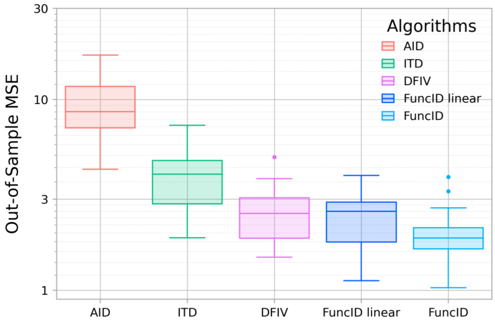
$$\min_{\omega\in\Omega} \mathbb{E}\left[\|Y - \mathbb{E}[f_\omega(A)|X]\|^2\right].$$

- but note that $\mathbb{E}[f_\omega(A)|X]$ is the optimal least-square estimator, which suggests

$$\min_{\omega\in\Omega} \mathbb{E}\left[\|Y - h^\star_\omega(X)\|^2\right] \quad \text{with} \quad h^\star_\omega = \arg\min_{h\in\mathcal{H}} \mathbb{E}\left[\|h(X) - f_\omega(A)\|^2\right].$$

# Two-stage least squares regression (2SLS)

Experiment on the dpsrite dataset from Xu et al. [2021]:




- advantage over AID/ITD (no conditioning problem due to degenerate Hessians).
- close to DFIV (same perf with different sample size).

# Model-based reinforcement learning

We rely on the bilevel RL formulation of Nikishin et al. [2022]. Consider a Markov decision process (MDP):

- $x = (s, a)$ represents a **state** $s$ and an **action** $a$ taken by an agent.
- current state/action $x = (s, a)$ yields a **future reward** $r'$ and **next state** $s'$, modeled by the joint probability distribution $(x, r', s') \sim \mathbb{P}$.

# Model-based reinforcement learning

We rely on the bilevel RL formulation of Nikishin et al. [2022]. Consider a Markov decision process (MDP):

- $x = (s, a)$ represents a **state** $s$ and an **action** $a$ taken by an agent.
- current state/action $x = (s, a)$ yields a **future reward** $r'$ and **next state** $s'$, modeled by the joint probability distribution $(x, r', s') \sim \mathbb{P}$.
- We need to learn a **model** with parameters $\omega$ that can predict the next state $s_\omega(x)$ and reward $r_\omega(x)$ given $x$.

# Model-based reinforcement learning

We rely on the bilevel RL formulation of Nikishin et al. [2022]. Consider a Markov decision process (MDP):

- $x = (s, a)$ represents a **state** $s$ and an **action** $a$ taken by an agent.
- current state/action $x = (s, a)$ yields a **future reward** $r'$ and **next state** $s'$, modeled by the joint probability distribution $(x, r', s') \sim \mathbb{P}$.
- We need to learn a **model** with parameters $\omega$ that can predict the next state $s_\omega(x)$ and reward $r_\omega(x)$ given $x$.
- We also need to learn an **action-value function** $h^\star_\omega$ that estimates the **expected cumulative reward** given a action/state pair $x = (s, a)$.

$$h^\star_\omega = \underset{h \in \mathcal{H}}{\arg\min}\, \mathbb{E}_x[\ell(h(x), r_\omega(x), s_\omega(x))],$$

where $\ell$ is the Bellman error (lots of details hidden under the carpet).

# Model-based reinforcement learning

- We need to learn an **action-value function** $h^\star_\omega$ that estimates the **expected cumulative reward** given a action/state pair $x = (s, a)$.

$$h^\star_\omega = \arg\min_{h \in \mathcal{H}} \mathbb{E}_x[\ell(h(x), r_\omega(x), s_\omega(x))],$$

where $\ell$ is the Bellman error (lots of details hidden under the carpet).

- The parameters of the MDP model are learned by also minimizing the Bellman error, with true samples from $\mathbb{P}$ this time:

$$\min_{\omega \in \Omega} \mathbb{E}_{x,r',s'}[\ell(h^\star_\omega(x), r', s')].$$

# Model-based reinforcement learning

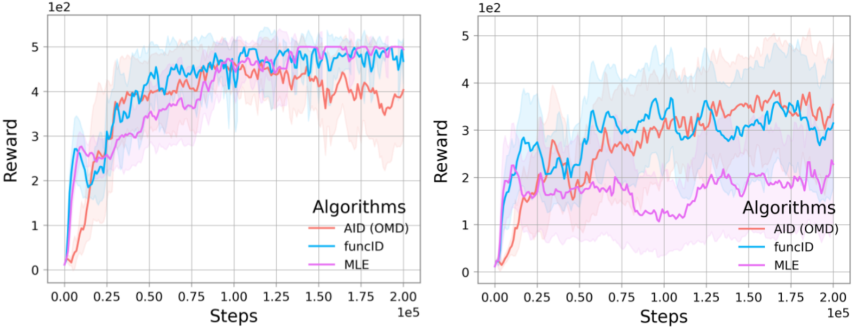



Figure 2: Average reward on an evaluation environment vs. training iterations on the *CartPole* task. (**Left**) Well-specified model. (**Right**) Misspecified model with 3 hidden units. Both plots show mean reward over 10 runs where the shaded region is the 95% confidence interval.

# Conclusion

- The functional point of view **solves many conceptual issues** for bilevel optimization in machine learning.
- It is fully **compatible with deep neural networks**.
- Despite the infinite dimension, it comes with **concrete algorithms** with reasonable complexity.

**We are just scratching the surface.**
**This is perhaps a new playground for machine learners/optimizers!**

# References I


Pierre Ablin, Gabriel Peyré, and Thomas Moreau. Super-efficiency of automatic differentiation for functions defined as a minimum. In *International Conference on Machine Learning (ICML)*, 2020.

Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. *International Conference on Machine Learning (ICML)*, 2020.

Michael Arbel and Julien Mairal. Amortized implicit differentiation for stochastic bilevel optimization. *International Conference on Learning Representations (ICLR)*, 2022a.

Michael Arbel and Julien Mairal. Non-convex bilevel games with critical point selection maps. *Advances in Neural Information Processing Systems*, 35:8013–8026, 2022b.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint 1907.02893*, 2019.

Kristin P. Bennett, Jing Hu, Xiaoyun Ji, Gautam Kunapuli, and Jong-Shi Pang. Model selection via bilevel optimization. *IEEE International Joint Conference on Neural Network Proceedings*, 2006.

# References II


Luca Bertinetto, João F. Henriques, Philip H.S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *International Conference on Learning Representations (ICLR)*, 2019.

Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Jérôme Bolte, Tam Le, Edouard Pauwels, and Tony Silveti-Falls. Nonsmooth implicit differentiation for machine-learning and optimization. *Advances in neural information processing systems*, 34:13537–13549, 2021.

Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. *Advances in Neural Information Processing Systems*, 35, 2022.

Matthias Feurer and Frank Hutter. *Hyperparameter optimization*. Springer International Publishing, 2019.

# References III


Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. *International Conference on Machine Learning (ICML)*, 2017.

Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. *Optimization and Control*, 2018.

Gernot Holler, Karl Kunisch, and Richard C. Barnard. A bilevel approach for parameter learning in inverse problems. *Inverse Problems*, 34(11):115012, 2018.

Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33(1):147–180, 2023.

Yonggang Li, Guosheng Hu, Yongtao Wang, Timothy Hospedales, Neil M Robertson, and Yongxin Yang. Differentiable automatic data augmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 580–595. Springer, 2020.

# References IV


Risheng Liu, Xuan Liu, Shangzhi Zeng, Jin Zhang, and Yixuan Zhang. Value-function-based sequential minimization for bi-level optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 45:15930–15948, 2021.

Jonathan Lorraine, Paul Vicol, and David Kristjanson Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Juliette Marrie, Michael Arbel, Diane Larlus, and Julien Mairal. SLACK: Stable learning of augmentations with cold-start and KL regularization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

A.S. Nemirovski and S.M. Semenov. On polynomial approximation of functions on hilbert space. *Mathematics of the USSR-Sbornik*, 21(2):255, 1973.

Evgenii Nikishin, Romina Abachi, Rishabh Agarwal, and Pierre-Luc Bacon. Control-oriented model-based reinforcement learning with implicit differentiation. *AAAI Conference on Artificial Intelligence*, 2022.

# References V


Ieva Petrulionyte, Julien Mairal, and Michael Arbel. Functional bilevel optimization for machine learning. *arXiv preprint arXiv:2403.20233*, 2024.

Rahul Singh, Maneesh Sahani, and Arthur Gretton. Kernel instrumental variable regression. *Advances in Neural Information Processing Systems*, 32, 2019.

Liyuan Xu, Heishiro Kanagawa, and Arthur Gretton. Deep proxy causal learning and its application to confounded bandit policy evaluation. *Advances in Neural Information Processing Systems*, 34:26264–26275, 2021.

Junjie Yang, Kaiyi Ji, and Yingbin Liang. Provably faster algorithms for bilevel optimization. *Advances in Neural Information Processing Systems*, 34:13670–13682, 2021.