A NOVEL ALGORITHM FOR ONLINE BILEVEL OPTIMIZATION FOR PRICE FORECASTING AROUND MARKET MOVING NEWS

Jason Bohne[1,2], David Rosenberg[2], Gary Kazantsev [2], Paweł Polak[1]

[1]Stony Brook University [2] Bloomberg

QR code for the full paper

Abstract

Bilevel optimization methods are increasingly relevant in machine learning, especially for tasks such as hyperparameter optimization and meta-learning. Compared to the offline setting, online bilevel optimization offers a more dynamic framework by accommodating time-varying functions and sequentially arriving data in the problem formulation. We introduce a novel **online Bregman bilevel optimizer (OBBO)** with an improved theoretical guarantee for regret minimization and an efficient computational implementation via PyTorch. Empirically, we apply OBBO against established online/offline bilevel benchmarks in an online hyperparameter optimization for financial time series and display the **superior performance of OBBO** in terms of forecasting loss on an independent test set.

Bilevel Optimization

Canonical hyperparameter optimization problem:
Optimization of a hyperparameter $\boldsymbol{\lambda}$ on a validation set for optimal parameters $\widehat{\beta}(\boldsymbol{\lambda})$ on a training set.

$$\arg\min_{\lambda \in \mathbb{R}^+} \frac{1}{2} \left\| \boldsymbol{y}^{val} - \boldsymbol{X}^{val} \widehat{\beta}(\lambda) \right\|_2^2, \qquad \text{(outer level)}$$

$$\widehat{\beta}(\lambda) \in \arg\min_{\beta \in \mathbb{R}^m} \frac{1}{2} \left\| \boldsymbol{y}^{train} - \boldsymbol{X}^{train} \boldsymbol{\beta} \right\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \qquad \text{(inner level)}$$

The above problem includes ridge regression, smoothing spline regression, and kernel ridge regression. Hyperparameter optimization is a special case of a **bilevel optimization**, where there is an outer level optimization parameterized by the optimal solution of an inner level optimization:

$$\arg\min_{\boldsymbol{\lambda} \in X \subseteq \mathbb{R}^{d_1}} \left\{ F(\boldsymbol{\lambda}) \triangleq f\left(\boldsymbol{\lambda}, \widehat{\boldsymbol{\beta}}(\boldsymbol{\lambda})\right) \right\}, \qquad \text{(outer level)}$$

$$\widehat{\boldsymbol{\beta}}(\boldsymbol{\lambda}) \in \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^{d_2}} g(\boldsymbol{\lambda}, \boldsymbol{\beta}) \qquad \text{(inner level)}$$

Other special cases include meta-learning, neural architecture search, dataset distillation, and RLHF.

The Hypergradient

One can differentiate through bilevel optimization problems under a few technical assumptions – leading to **bilevel specific gradient descent algorithms**. With the chain rule, the gradient of the outer level objective can be decomposed into a direct and indirect gradient term.

$$\nabla F(\boldsymbol{\lambda}) = \underbrace{\nabla_{\boldsymbol{\lambda}} f(\boldsymbol{\lambda}, \widehat{\boldsymbol{\beta}}(\boldsymbol{\lambda}))}_{(a)\text{ Hyperparameter Direct Gradient}} + \overbrace{\underbrace{\nabla \widehat{\boldsymbol{\beta}}(\boldsymbol{\lambda})}_{(b)\text{ Best-Response Jacobian}} \underbrace{\nabla_{\boldsymbol{\beta}} f(\boldsymbol{\lambda}, \widehat{\boldsymbol{\beta}}(\boldsymbol{\lambda}))}_{(c)\text{Parameter Direct Gradient}}}^{\text{Hyperparameter Indirect Gradient}}$$

Direct gradient terms (**a**,**c**) are computable if the objective is differentiable w.r.t. hyperparameters and parameters, e.g., neural networks. For any hyperparameter values $\boldsymbol{\lambda}$, the best-response Jacobian (**b**) is the typically **unknown** gradient of the corresponding optimal parameters. However with **implicit function theorem**, one can derive the best-response Jacobian in terms of computable gradients as

$$\nabla \widehat{\boldsymbol{\beta}}(\boldsymbol{\lambda}) = \underbrace{-\nabla^2_{\boldsymbol{\lambda},\boldsymbol{\beta}} g(\boldsymbol{\lambda}, \widehat{\boldsymbol{\beta}}(\boldsymbol{\lambda}))}_{\text{Training Partials}} \overbrace{\left(\nabla^2_{\boldsymbol{\beta},\boldsymbol{\beta}} g(\boldsymbol{\lambda}, \widehat{\boldsymbol{\beta}}(\boldsymbol{\lambda}))\right)^{-1}}^{\text{Training Hessian}}$$

An Improved Bilevel Optimizer

Our online Bregman bilevel optimizer (**OBBO**) generalizes the gradient step in known online bilevel optimizers through the application of a Bregman Divergence $\mathcal{D}_\phi(\cdot,\cdot)$. This offers a generalization from the squared Euclidean distance, as in Lin et al. 2024; Tarzanagh et al. 2024. Given a continuously differentiable ρ-strongly convex function $\phi(\boldsymbol{\lambda})$, a Bregman Divergence $\mathcal{D}_\phi(\cdot,\cdot)$ is defined for all $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2 \in \mathcal{X}$ as:

$$\mathcal{D}_\phi(\boldsymbol{\lambda}_2, \boldsymbol{\lambda}_1) := \phi(\boldsymbol{\lambda}_2) - \phi(\boldsymbol{\lambda}_1) - \langle \nabla\phi(\boldsymbol{\lambda}_1), \boldsymbol{\lambda}_2 - \boldsymbol{\lambda}_1 \rangle.$$

For a Bregman divergence $\mathcal{D}_\phi(\cdot,\cdot)$, our generalized gradient step then has the form of

$$\boldsymbol{\lambda}^+ = \arg\min_{\boldsymbol{\lambda} \in \mathcal{X}} \left\{ \langle \boldsymbol{q}, \boldsymbol{\lambda} \rangle + \frac{1}{\alpha} \mathcal{D}_\phi(\boldsymbol{\lambda}, \boldsymbol{u}) \right\},$$

where $\alpha > 0$ is a step size, and $\boldsymbol{q}, \boldsymbol{u} \in \mathbb{R}^{d_1}$. Our analysis shows **OBBO** achieves an improved sublinear rate of bilevel local regret– a measure of stationarity for online bilevel algorithms from Tarzanagh et al. 2024. For a window smoothing parameter $w \geq 1$, the bilevel local regret is defined for a smooth $F_t(\boldsymbol{\lambda})$ as

$$BLR_w(T) := \sum_{t=1}^{T} \left\| \nabla F_{t,w}(\boldsymbol{\lambda}_t) \right\|^2, \quad F_{t,w}(\boldsymbol{\lambda}_t) := \frac{1}{w} \sum_{i=0}^{w-1} F_{t-i}(\boldsymbol{\lambda}_{t-i}),$$

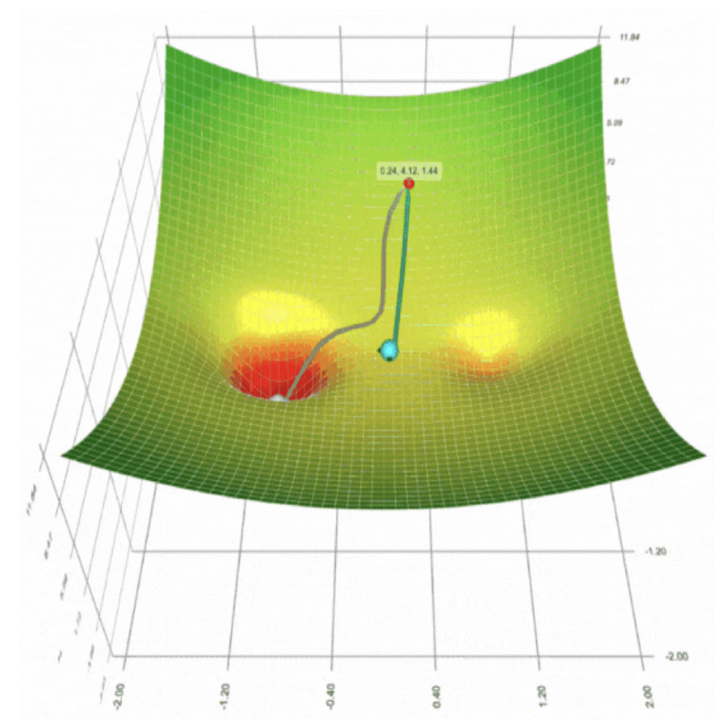
Specifically, the condition number of the inner objective $g(\boldsymbol{\lambda}, \boldsymbol{\beta})$ is $\kappa_g > 1$. **OBBO** achieves a κ_g^2 dependency, whereas benchmarks of SOBOW (Lin et al. 2024) and OAGD (Tarzanagh et al. 2024) only achieve κ_g^3 and κ_g^4 respectively.

- One special case of the generalized gradient step given adaptive matrix $\mathbf{H}_t$ is **Adagrad** – which can better capture the underlying geometry via use of adaptive learning rates.
- Another special case of the generalized gradient step is the reduction to gradient descent, when $\phi(\boldsymbol{\lambda}) = \frac{1}{2}\|\boldsymbol{\lambda}\|^2$ and $\mathcal{X} = \mathbb{R}^{d_1}$.
- Empirically, we update hyperparameters with an Adagrad step (2nd order information) vs. a gradient descent step (only 1st order).

Example	Bregman Function
Gradient Descent	$\frac{1}{2}\|\boldsymbol{\lambda}\|_2^2$
Adagrad	$\frac{1}{2}\boldsymbol{\lambda}^T H_t \boldsymbol{\lambda}$

Table 2: Example Bregman Functions

Figure 1: For local minimum in above, see the improvement of Adagrad (gray) relative to Gradient Descent (teal).



Full paper can be found on https://arxiv.org/abs/2409.10470

Application to Hyperparameter Optimization

Consider **hyperparameter optimization** where:

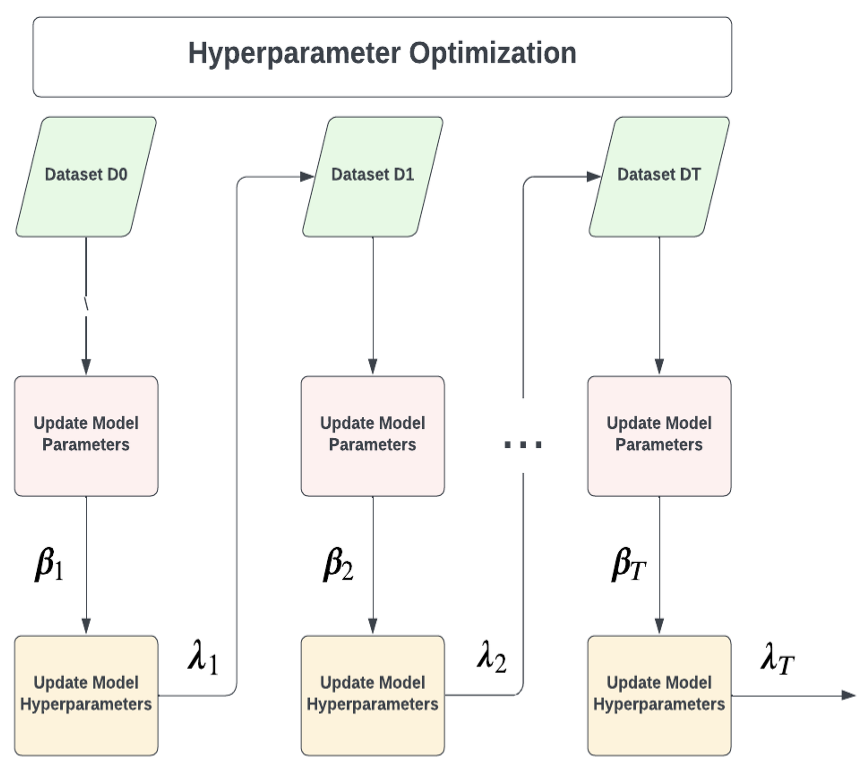
- New training-validation datasets sequentially arrive such that optimal values for (hyper)-parameters can vary over datasets.
- The goal is to update hyperparameters $\boldsymbol{\lambda}_t$ for optimal parameters $\widehat{\beta}_t(\boldsymbol{\lambda}_t)$ on dataset D_t.
- Ex: Linear smoothing spline model with B-spline coefficients as parameters β_t and positive regularization hyperparameter $\boldsymbol{\lambda}_t$.

For datasets $D_t := (\boldsymbol{X}_t^{train}, \boldsymbol{y}_t^{train}, \boldsymbol{X}_t^{val}, \boldsymbol{y}_t^{val})$, formulate our hyperparameter optimization $\forall t$ as

$$\arg\min_{\lambda \in \mathbb{R}^+} \frac{1}{2} \left\| \boldsymbol{y}_t^{val} - \boldsymbol{X}_t^{val} \widehat{\beta}_t(\lambda) \right\|_2^2,$$

$$\widehat{\beta}_t(\lambda) \in \arg\min_{\beta \in \mathbb{R}^m} \frac{1}{2} \left\| \boldsymbol{y}_t^{train} - \boldsymbol{X}_t^{train} \boldsymbol{\beta} \right\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2$$

Figure 2: Diagram of Hyperparameter Optimization



Price Forecasting around Market Moving News

Market Moving News Dataset:

- Significant price events and news stories annotated via segmentation algorithm.
- Subset of the RAY index (U.S. equities) between January 1, 2021 and June 1, 2022.

Experiment Setup:

- 440 samples of equity time series partitioned into a rolling window of training-validation data.
- Separate test set post annotated price event to evaluate forecasting mean-squared error.
- Model choice of linear smoothing spline with B-spline coefficients as parameters β_t and regularization hyperparameter $\boldsymbol{\lambda}_t$ results in forecasts satisfying assumptions of annotation pipeline.

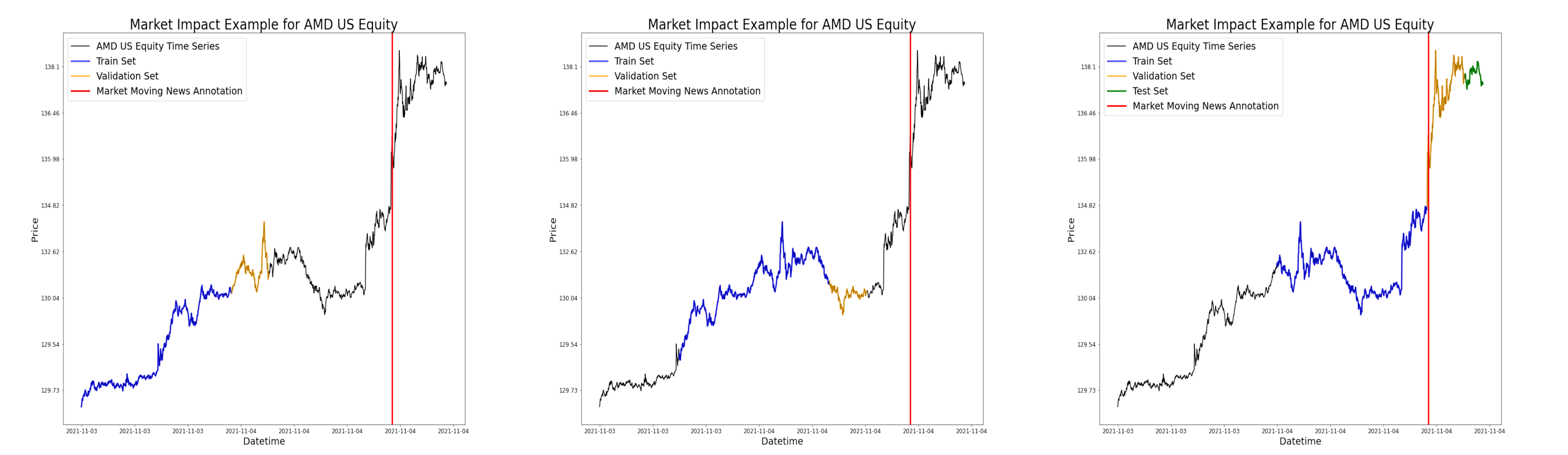


Figure 3: Sample training-validation subsets for AMD U.S. Equity and market event on 11-08-2021.

Benchmark algorithms include gradient bilevel algorithms OAGD (Tarzanagh et al. 2024) and SOBOW (Lin et al. 2024), limited to a gradient descent step. Further benchmarked by general purpose optimizers; ADAM (Kingma 2014), SGDM.

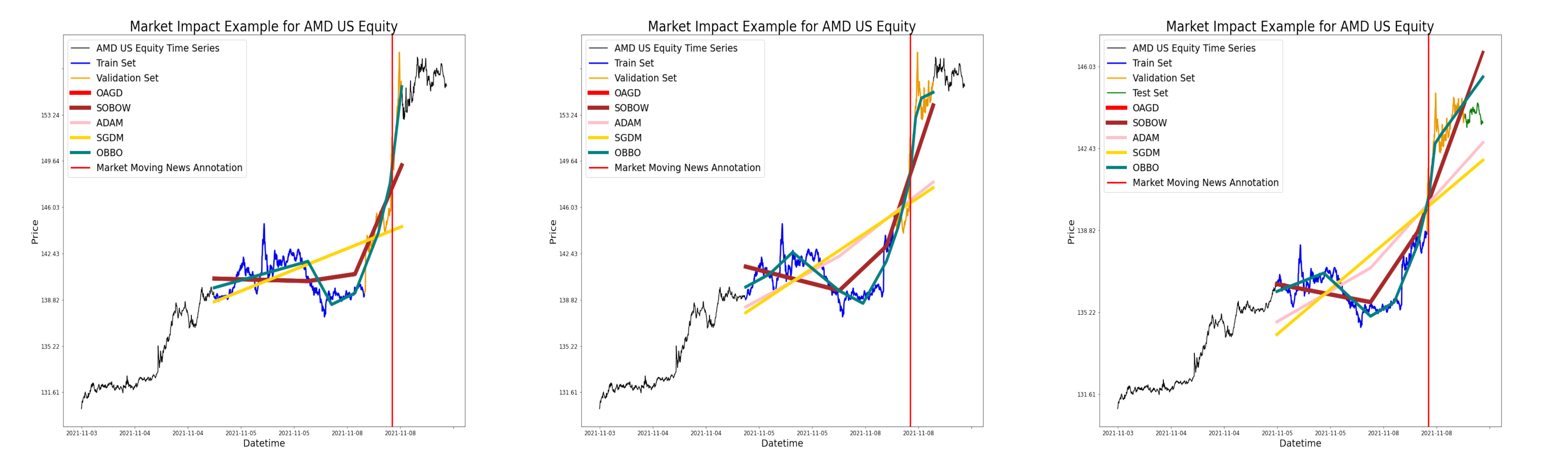


Figure 4: Example forecasts generated with **OBBO** vs. benchmark algorithms.

Better convergence of **OBBO** results in quicker adaptability of our underlying model to annotated event and improved forecasting MSE on a test set post market event— statistics provided below.

	Forecasting Loss across U.S. Markets	
Algorithm	Mean Loss ± Standard Error	Median Loss ± Median Absolute Deviation
OBBO	**0.661** ± 0.055	**0.205** ± 0.150
OAGD	0.707 ± 0.053	0.265 ± 0.209
SOBOW	0.689 ± 0.053	0.273 ± 0.215
Adam	1.265 ± 0.176	0.267 ±0.230
SGDM	0.872 ± 0.078	0.401 ± 0.286

Table 3: Statistics of forecasting mean-squared error across U.S. markets.

Conclusions and Extensions

- Hyperparameter optimization (HO) is actually a special case of bilevel optimization.
- Provide an **improved algorithm for general bilevel optimization** problems.
- Empirically show benefit of our improved algorithm in HO for financial time series forecasting.
- Determine if our algorithm offers an empirical improvement in **other special cases**, e.g., RLHF.

References:

Kingma, Diederik P (2014). "Adam: A Method for Stochastic Optimization". In: *arXiv preprint arXiv:1412.6980.*

Lin, Sen et al. (2024). "Non-Convex Bilevel Optimization with Time-Varying Objective Functions". In: *Advances in Neural Information Processing Systems* 36.

Tarzanagh, Davoud Ataee et al. (2024). "Online Bilevel Optimization: Regret Analysis of Online Alternating Gradient Methods". In: *International Conference on Artificial Intelligence and Statistics.*