Expert-Calibrated Learning for Online Optimization with Switching Costs

Pengfei Li*, Jianyi Yang*, Shaolei Ren

University of California, Riverside

*Pengfei Li and Jianyi Yang contributed equally.
Acknowledgement: This work was supported in part by the U.S. NSF under grants CNS-1551661 and CNS-2007115.
Online optimization is everywhere…

Siemens Smart Infrastructure announced an open, modular suite of grid software to address the increasingly critical energy transition. The suite enables stakeholders to be fast, agile and flexible when coping with existing and future energy challenges, the company says.

The new grid software offering enables energy companies to easily and quickly implement smart technology for their grid planning and management. These IT upgrades will be faster and less costly than investments in hardware, according to Siemens. Through the modular

AWS launches Gateway Load Balancer for highly available and scalable network appliances

Robotic camera mimics human operators to anticipate basketball game action

Automated cameras make it possible to broadcast even minor events, but the result often looks...well, robotic. Now scientists at Disney Research have made it possible for robotic cameras to learn from human operators how to better frame shots of a basketball game.
Smoothed online convex optimization (SOCO)

Demand

Servers

$T = 0$
Smoothed online convex optimization (SOCO)

Demand

T = 1

Servers

4
Smoothed online convex optimization (SOCO)
Problem formulation

Goal

\[
\min_{x_1, \ldots, x_T} \sum_{t=1}^{T} f(x_t, y_t) + c(x_t, x_{t-1})
\]

Assumptions

1. Online information \( y_t \in \mathcal{Y}_t \) is revealed sequentially
2. Hitting cost \( f(x_t, y_t) \) is non-negative and \( m \)-strongly convex in \( x_t \)
3. Switching cost \( c(x_t, x_{t-1}) \) is \( \beta \)-smooth and \( \alpha \)-strongly convex in \( x_t \) and \( x_{t-1} \) (e.g. squared Mahalanobis distance)
Performance metrics

Average Cost

For contextual information \( s = (x_0, y) \sim \mathbb{P} \), the average cost of an algorithm \( \pi \) is defined as

\[
\text{AVG}(\pi) = \mathbb{E} \left[ \text{cost}(\pi, s) \right]
\]

with expectation taken over the joint distribution \( \mathbb{P} \).

Competitive ratio

The competitive ratio of an algorithm \( \pi \) is defined as

\[
\text{CR}(\pi) = \sup_{s \in S} \frac{\text{cost}(\pi, s)}{\text{cost}(\pi^*, s)}
\]
Why is SOCO challenging?

Hitting cost curve

\[ f(x, y_0), f(x, y_2), f(x, y_1) \]

\[ x_0, x_2, x_1 \]

**A simple policy:** Minimize hitting cost

\[
\text{cost}_1 = f(x_1, y_1) + c(x_1, x_0) \\
\text{cost}_2 = f(x_2, y_2) + c(x_2, x_1)
\]

The hitting cost is minimized, but we may pay too much switching cost.
The competitive ratio of ML is at least
\[ 1 + \frac{m + 2\alpha}{2} \rho \]
where \( \rho \) is the prediction error of ML (Lemma 3.1)

Related works

Performance

ML models

Expert algorithms

Robustness

R-OBD [1] with competitive ratio
\[ CR = \frac{1}{2} \left( \sqrt{1 + \frac{4\rho^2}{\alpha m}} + 1 \right) \]

[1] Goel, G., Lin, Y., Sun, H., & Wierman, A. (2019). Beyond online balanced descent: An optimal algorithm for smoothed online optimization. *Advances in Neural Information Processing Systems*, 32.
Related works

Key drawbacks:
1. Treat ML as a black box
2. Standalone ML training

ML-augmented algorithms

[1] Goel, G., Lin, Y., Sun, H., & Wierman, A. (2019). Beyond online balanced descent: An optimal algorithm for smoothed online optimization. Advances in Neural Information Processing Systems, 32.
Main contributions

1. Calibration-aware ML training
2. Novel differentiable expert algorithm
3. Better robustness and average performance
Algorithm 1 Machine Learning Augmented R-OBD (MLA-ROBD)

**Input:** 0 < λ₁ ≤ 1, λ₂ ≥ 0, λ₃ ≥ 0, the initialized action x₀.
1: for t = 1, ⋯ , T
2:   Receive the context yₜ.
3:   \( vₜ \leftarrow \arg \min_{x \in \mathcal{X}} f(x, yₜ) \) //Minimizer of the current hitting cost
4:   \( \tilde{xₜ} \leftarrow hₖ(xₜ₋₁, yₜ) \) //ML prediction
5:   \( xₜ \leftarrow \arg \min_{x \in \mathcal{X}} f(x, yₜ) + \lambda₁ c(x, xₜ₋₁) + \lambda₂ c(x, vₜ) + \lambda₃ c(x, \tilde{xₜ}) \) //Calibrating the ML prediction
Competitive ratio

Theorem 4.1

If the ML predictions are $\rho$-accurate, the competitive ratio of MLA-ROBD has a competitive ratio upper bound of

$$\max \left( \frac{m + \lambda_2 \beta}{m \lambda_1}, 1 + \frac{\beta^2}{\alpha} \cdot \frac{\lambda_1}{(\lambda_2 + \lambda_3)\beta + m} \right) + \frac{\lambda_3 \beta}{2\lambda_1} \rho$$

Moreover, by optimally setting $\lambda_2$ with $\theta = \lambda_3 / \lambda_1$ being the trust parameter, the competitive ratio upper bound becomes

$$1 + \frac{1}{2} \left[ \sqrt{(1 + \frac{\beta}{m \theta})^2 + \frac{4\beta^2}{ma}} - \left(1 + \frac{\beta}{m \theta}\right) \right] + \frac{\beta \theta}{2} \cdot \rho$$

Better competitive ratio than R-OBBD when ML predictions are good

With proper $\theta$, better competitive ratio than pure ML
Comparison of competitive ratio
How to learn?

\[\text{loss} = \sum_{t=1}^{T} \text{cost}_t(\tilde{x}_t)\]

\[= \sum_{t=1}^{T} f(\tilde{x}_t, y_t) + c(\tilde{x}_{t-1}, x_t)\]
How to learn?

Training

\[
\text{loss} = \sum_{t=1}^{T} \text{cost}_t(x_t)
\]

\[
= \sum_{t=1}^{T} f(x_t, y_t) + c(x_{t-1}, x_t)
\]

Inference

Calibration

Learning Model

Post-calibration is good, but pre-calibration may not be good, because it is not even considered.
Expert-calibrated learning to optimize (EC-L2O)

\[ \text{loss} = (1 - \mu) \sum_{t=1}^{T} \text{cost}(x_t) + \mu \text{ReLU} \left( \sum_{t=1}^{T} l_t(\tilde{x}_t) - \bar{\rho} \right) \]

\[ = (1 - \mu) \sum_{t=1}^{T} f(x_t, y_t) + c(x_{t-1}, x_t) + \mu \text{ReLU} \left( \sum_{t=1}^{T} \frac{||\tilde{x}_t - x^*_t||^2}{\text{cost}(\pi^*, s)} - \bar{\rho} \right) \]
Expert differentiation for EC-L2O

Technical challenges

1. The expert-calibrator (i.e. MLA-ROBD) is an *implicit* layer: Closed-form differentiation is nontrivial, but vital for gradient-based optimization algorithm.
2. Back-propagation needs to be performed recurrently through time

Key idea

\[
\nabla_{x_t} f(x_t, y_t) + \lambda_1 \nabla_{x_t} c(x_t, x_{t-1}) + \lambda_2 \nabla_{x_t} c(x_t, v_t) + \lambda_3 \nabla_{x_t} c(x_t, \tilde{x}_t) = 0.
\]
Performance analysis

**Average cost**

\[
\text{AVG}(R_\lambda \circ h_{\hat{W}}) \leq \text{AVG}(R_\lambda \circ h_{W*}) + \frac{\mu}{1 - \mu} \mathbb{E}[l(h_{W*}, s)] + \frac{1}{1 - \mu} \mathbb{E}_\mathcal{D} + O \left( \sqrt{\log(1/\delta) / |\mathcal{D}|} + D(\mathbb{P}, \mathbb{P}') \right)
\]

**Optimal average cost**

\[
\text{AVG}(R_\lambda \circ h_{W*})
\]

**Loss from restraining the prediction error**

\[
\frac{\mu}{1 - \mu} \mathbb{E}[l(h_{W*}, s)]
\]

**Training error**

\[
\frac{1}{1 - \mu} \mathbb{E}_\mathcal{D}
\]

**Error from data finiteness and distribution shift**

\[
O \left( \sqrt{\log(1/\delta) / |\mathcal{D}|} + D(\mathbb{P}, \mathbb{P}') \right)
\]

**Tail competitive ratio**

With probability at least \( \delta, \delta \in (0, 1) \), CR of EC-L2O satisfies

\[
\frac{\text{cost}(R_\lambda \circ h_{\hat{W}}, s)}{\text{cost}(\pi^*, s)} \leq 1 + \frac{1}{2} \left[ \sqrt{1 + \frac{\beta}{m} \theta}^2 + \frac{4\beta^2}{m\alpha} - \left( 1 + \frac{\beta}{m\theta} \right) \right] + \frac{\beta}{2} \theta \cdot \rho_{\text{tail}}
\]

\[
\rho_{\text{tail}} = \bar{\rho} + \mathbb{E}[l(h_{W*}, s)] + \frac{1 - \mu}{\mu} \mathbb{E}[\text{cost}(R_\lambda \circ h_{W*}, s)] + \frac{1}{\mu} \mathbb{E}_\mathcal{D} + O \left( \sqrt{\log(2/\delta) / |\mathcal{D}|} \right) + O(D(\mathbb{P}, \mathbb{P}') + O \left( \frac{\omega^2 \sqrt{T}}{\nu} \parallel \Gamma \parallel \sqrt{\frac{1}{2} \log \left( \frac{4}{\delta} \right)} \right)
\]

Sequence deviation from average
Case study: Datacenter demand response

Total energy generated by the renewables

\[ P_{r,t} = P_{\text{wind},t} + P_{\text{solar},t} \]

The energy reduction request at time step \( t \)

\[ y_t = \max(P_{s,t} - P_{r,t}, 0) \]

Image credit: Nithya Balakrishnan, Powering India's Data Center Boom Through Low-Cost, Low-Carbon Energy Solutions, April 2021.
Simulation setup

(a) Testing of Apr. to Jun.  
(b) Testing of Jul. to Sep.  
(c) Testing of Oct. to Dec.

tSNE of training-testing distribution discrepancies
## Baseline algorithms

| Algorithm   | Description                                                                 |
|-------------|-----------------------------------------------------------------------------|
| **Oracle**  | Offline optimal oracle with complete context information                    |
| **R-OBD [1]** | State-of-the-art expert algorithm                                           |
| **PureML**  | Standalone ML optimizer without considering calibration                       |
| **Switch [2]** | ML-augmented algorithm modified based on metrical switching costs [2] to dynamically switch between R-OBD and PureML |
| **MLA-ROBD** | The same RNN architecture and calibration algorithm as EC-L2O, but trained as a standalone optimizer |

[1] Goel, G., Lin, Y., Sun, H., & Wierman, A. (2019). Beyond online balanced descent: An optimal algorithm for smoothed online optimization. *Advances in Neural Information Processing Systems*, 32.

[2] Antoniadis, A., Coester, C., Elias, M., Polak, A., & Simon, B. (2020, November). Online metric algorithms with untrusted predictions. In *International Conference on Machine Learning* (pp. 345-355). PMLR.
Simulation results
Tail cost comparison
Take-home message

1. Calibration-aware ML training
2. Novel differentiable expert algorithm
3. Better robustness and average performance