Beyond Online Balanced Descent: An Optimal Algorithm for Smoothed Online Convex Optimization

Gautam Goel

Based on joint work with Yiheng Lin, Haoyuan Sun, and Adam Wierman

Portfolio Optimization

Adaptive Control

Portfolio Optimization

Adaptive Control

This talk: how do we design online learning algorithms that adapt to dynamic environments while accounting for switching costs?

Online Convex Optimization (OCO) with **one-step lookahead** and **switching costs**

An online learner plays a series of rounds against an adaptive adversary. In the t-th round:

1. The adversary chooses an m-strongly-convex cost function $f_t : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$.
2. **After** observing f_t, the learner picks a point $x_t \in \mathbb{R}^d$.
3. The online learner pays the **hitting cost** $f_t(x_t)$ as well as a **switching cost** $\frac{1}{2}\|x_t - x_{t-1}\|_2^2$ which penalizes the learner for changing its decisions between rounds.

$$\text{Competitive Ratio} = \sup_{f_1,\ldots,f_T} \frac{\sum_{t=1}^{T} f_t(x_t) + \frac{1}{2}\|x_t - x_{t-1}\|^2}{\underbrace{\min_{x_1,\ldots,x_T} \sum_{t=1}^{T} f_t(x_t) + \frac{1}{2}\|x_t - x_{t-1}\|^2}_{\text{Dynamic optimal solution}}}.$$

Online Balanced Descent (OBD)

Key idea #1: Project onto level sets (otherwise you incur extra switching cost!).

Online Balanced Descent (OBD)

Key idea #1: Project onto level sets (otherwise you incur extra switching cost!).

Key idea #2: Pick level set so that switching cost hitting cost.

Theorem (Goel, Lin, Sun, Wierman '19)

Suppose the hitting cost functions are m-strongly convex with respect to the ℓ_2 norm and the switching cost is given by $c(x_t; x_{t-1}) = \frac{1}{2}\|x_t - x_{t-1}\|_2^2$. Any online algorithm must have a competitive ratio at least $\frac{1}{2}\left(1 + \sqrt{1 + \frac{4}{m}}\right)$. A modified version of OBD, called Regularized-OBD (R-OBD) exactly achieves the optimal $\frac{1}{2}\left(1 + \sqrt{1 + \frac{4}{m}}\right)$ competitive ratio.