Optimal regret algorithm for Pseudo-1d Bandit Convex Optimization

Aadirupa Saha[1], Nagarajan Natarajan[2], Praneeth Netrapallli [2,3], Prateek Jain [2,3]
[1] Microsoft Research, New York City, USA
[2] Microsoft Research, Bangalore, India
[3] Authors currently in Google Research, Bangalore, India

38th International Conference on Machine Learning, 2021

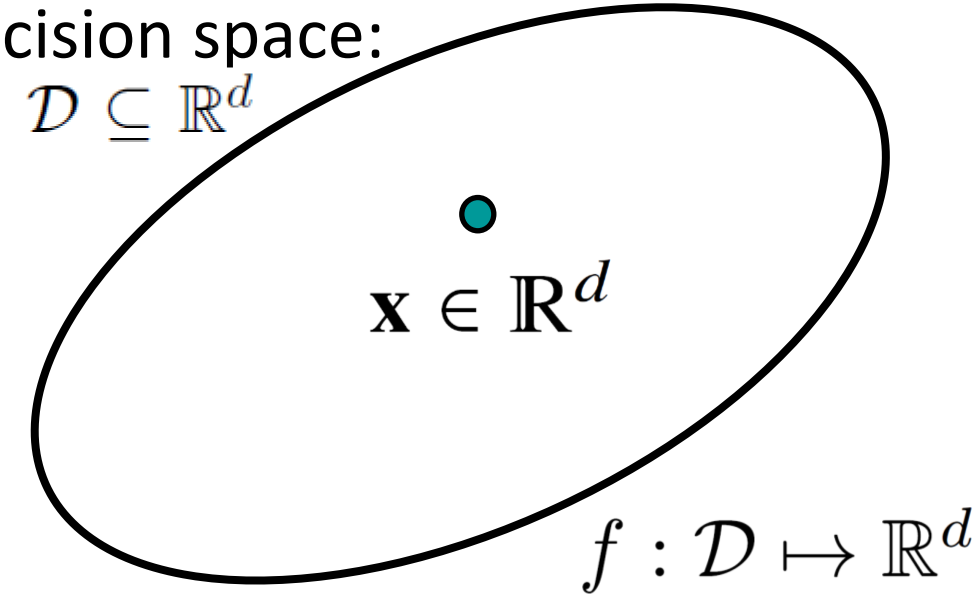
Problem Overview

Bandit Convex Optimization

Problem: Online Convex Optimization

Classical problem of Online Convex Optimization (OCO)

Decision space:



Convex

$$\{f_1, f_2, \ldots, f_t\} : \mathcal{D} \mapsto \mathbb{R}^d$$

At round t = 1,2,…,T

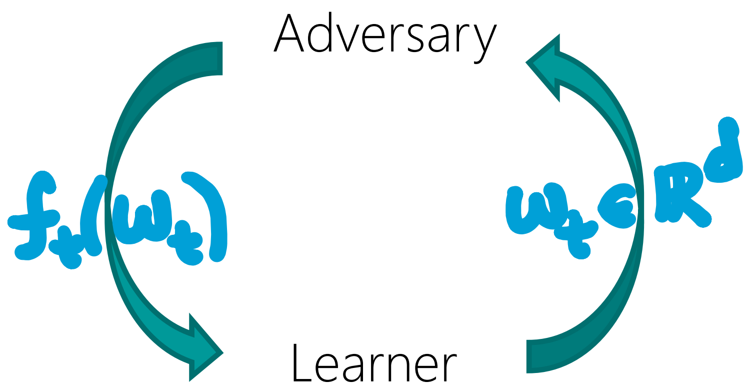
- Environment chooses f_t
- Play point $x_t \in D$
- Receive feedback at x_t

End

First order feedback:
Gradient information $\nabla f_t(\mathbf{x}_t)$

Zeroth order feedback:
Function value $f_t(\mathbf{x}_t)$

Zeroth order / Bandit Convex Optimization



Loss $\boldsymbol{f_t}$ is unobserved every round

Only black-box or zeroth-order access to loss

Regret (minimization) in T time steps:

$$R_T := \sum_{t=1}^{T} f_t(\boldsymbol{w_t}) - \min_{\boldsymbol{w}\in\mathbb{R}^d} \sum_{t=1}^{T} f_t(\boldsymbol{w})$$

Results known:

For general (convex) function lower bound = $\Omega(d\sqrt{T})$

- Gradient descent (w/ 1 point gradient estimate)

[Flaxman et al, Online convex optimization in the bandit setting: gradient descent without a gradient., 2005]

- More structures: Linear, Strong convexity or Smooth functions

[Saha and Tewari, Improved regret guarantees for online smooth convex optimization with bandit feedback, 2011]

- Multipoint estimates

[Ghadimi and Lan, Stochastic first-and zeroth-order methods for nonconvex stochastic programming, 2013]

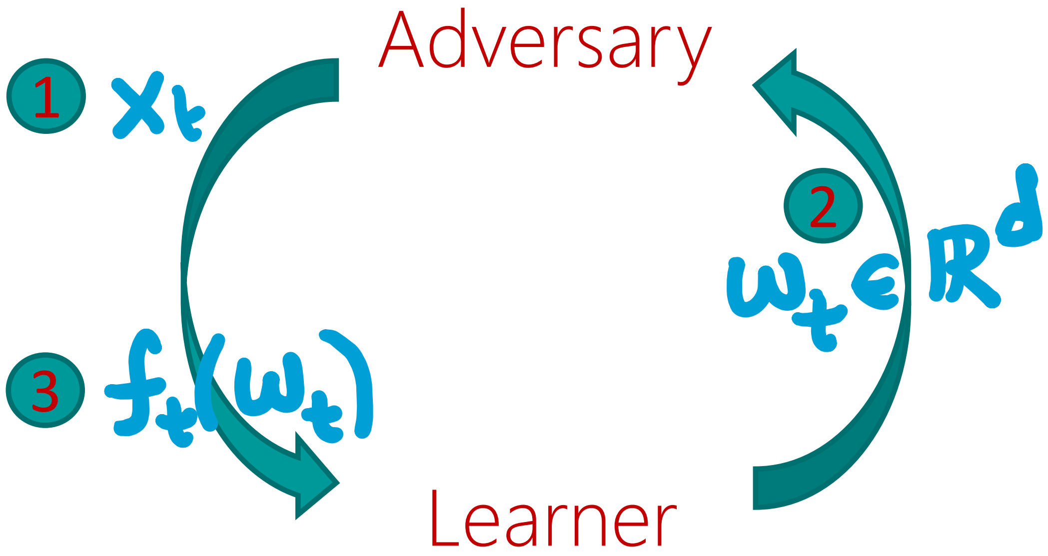
- **Optimal algorithm $\mathbf{O}\left(\mathbf{poly}(\boldsymbol{d})\sqrt{\boldsymbol{T}}\right)$**

[Hazan and Li, An optimal algorithm for bandit convex optimization, 2016]

[Bubeck et al, Kernel-based methods for bandit convex optimization, 2017]

Pseudo1d Bandit Convex Optimization:
Composite and Structured Functions

Pseudo1d BCO:

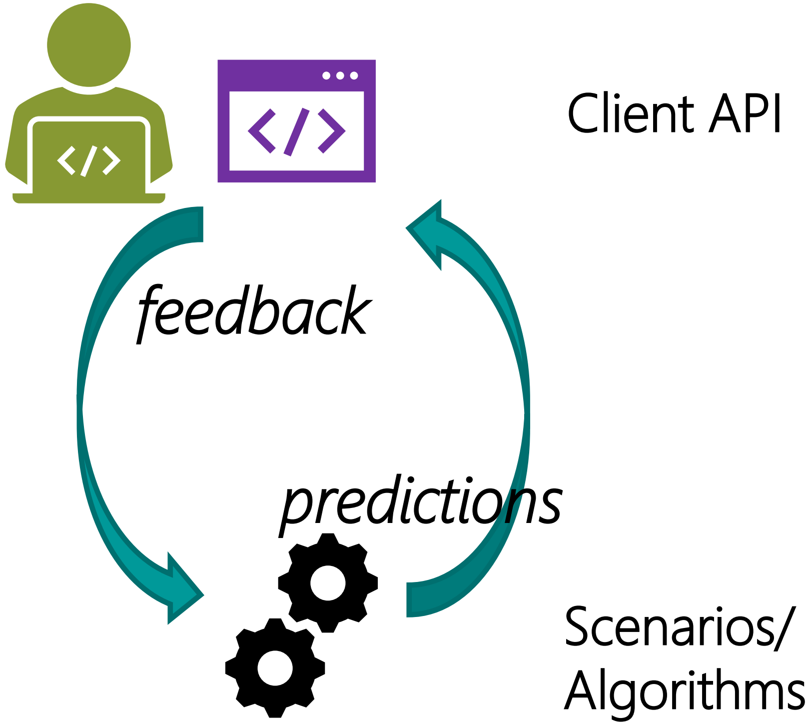


Pseudo-1d loss: $f_t(\boldsymbol{w}) = \ell_t(g(\boldsymbol{w}; x_t))$

Loss $\boldsymbol{\ell_t}$ is unobserved every round learner has point-only or zeroth-order access

But $\boldsymbol{g}$ is known to the learner, say $g(\boldsymbol{w}; x_t) =< \boldsymbol{w}.x_t >$

Applications: Self-Tune Framework (and many more….)



Large-scale parameter tuning in systems & software:

- online tuning
- reward/loss function is unobserved
- feedback is often expensive/limited

Pseudo1d Regret minimization?

$$R_T := \sum_{t=1}^{T} f_t(\boldsymbol{w_t}) - \min_{\boldsymbol{w} \in \mathbb{R}^d} \sum_{t=1}^{T} f_t(\boldsymbol{w})$$

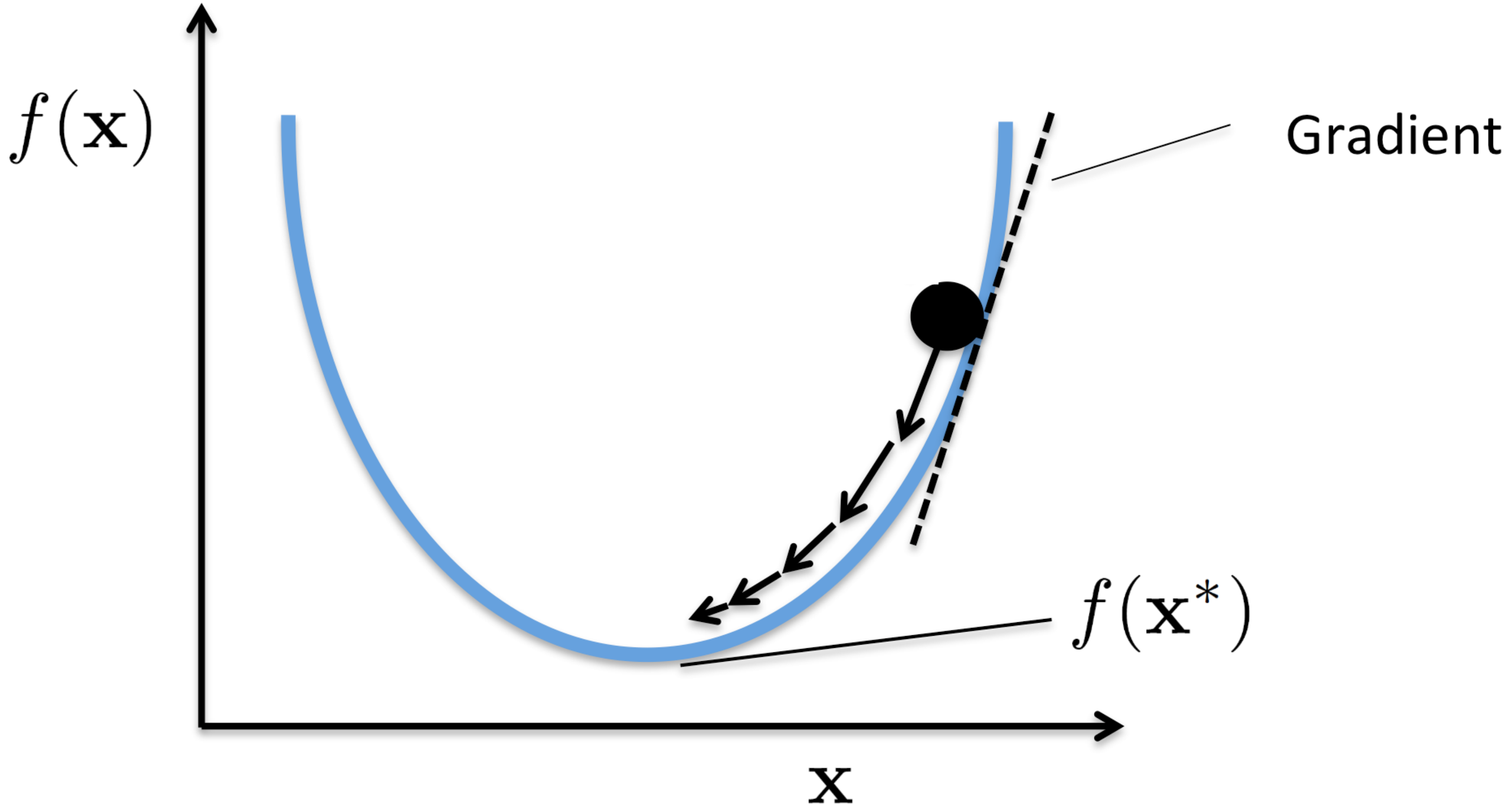
Let's try Gradient Descent!

Online Gradient Descent:

1. Estimate $\widetilde{\nabla}_{w} f_t(\boldsymbol{w_t})$
2. $\boldsymbol{w_{t+1}} = \boldsymbol{w_t} - \eta \widetilde{\nabla}_{w} f_t(\boldsymbol{w_t})$

For convex, Lipschitz f_t, Regret = $O\left(\sqrt{d}T^{3/4}\right)$

[Flaxman et al, Online convex optimization in the bandit setting: Gradient descent without a gradient., 2005]



Careful Gradient Descent!

Can exploit Pseudo-1d structure:

1. Estimate $\tilde{\nabla}_w f_t(\boldsymbol{w}_t) = \ell'_t\big(\langle \boldsymbol{x}_i, \boldsymbol{w}_t\rangle\big) \,.\, \boldsymbol{x}_t$
2. $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta\tilde{\nabla}_w f_t(\boldsymbol{w}_t)$

Independent of dimension d

[Our work]

For convex, Lipschitz losses, Regret = $\mathbf{O}\big(T^{3/4}\big)$

$$|\ell_t(a) - \ell_t(a')| \le L|a - a'|, a, a' \in \mathbb{R}$$

Main Question:

Improved learning rate $O(\sqrt{T})$?

Independent of dimension d?

What is the Lower Bound (in d and T)?

In the worst case, any algo... ret:

$$\left[\sum^{T} \dots \right] = \Omega\left(\min\left(\sqrt{dT}, T^{3/4}\right)\right)$$

Improvement factor $O\left(\sqrt{d}\right)$

An Optimal $O\left(\sqrt{dT}\right)$ Algorithm?

Kernelized Exponential Weight

Kernelized EXP3:

Algorithm 2 Kernelized Exponential Weights for PBCO

```
Input: learning rate: η > 0, ε > 0, max rounds T.
Initialize: w_1 ← 0, p_1 ← 1/vol(W).
for t = 1, 2, ··· T do
    Receive x_t, and define G_t := {g_t(w, x_t) | w ∈ W} ⊆ ℝ
    Define q_t such that dq_t(y) := ∫_{W_t(y)} dp_t(w), ∀y ∈ G_t, where W_t(y) := {w ∈ W | g_t(w, x_t) = y}
    Using x_t and q_t, and given ε, define kernel K'_t : G_t × G_t ↦ ℝ (according to Definition 4)
    Sample y_t ∼ K'_t q_t and pick any w_t ∈ W_t(y_t) uniformly at random
    Play g_t(w_t; x_t) and receive loss f_t(w_t) = ℓ_t(g_t(w_t; x_t))
    f̃_t(w) ← f_t(w_t) / (K'_t q_t(y_t)) · K'_t(y_t, y), for all w ∈ W(y), ∀y ∈ G_t      ▷ estimator of f_t
    p_{t+1}(w) ← p_t(w) exp(−η f̃_t(w)) / ∫_w̃ p_t(w̃) exp(−η f̃_t(w̃)) dw̃, for all w ∈ W
end for
```

Estimating $f_t(\boldsymbol{w})\ \ \forall \boldsymbol{w}$ (step 9)

Exponential wt update (step 10)

Regret Guarantee:

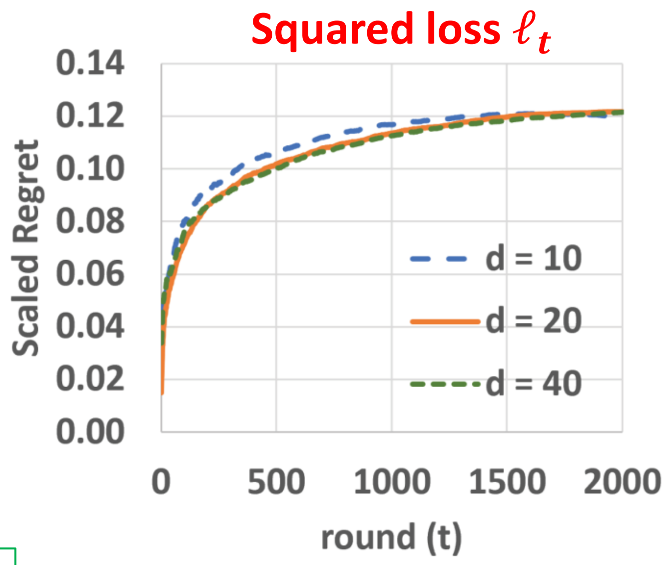
For convex & Lipschitz ℓ_t,

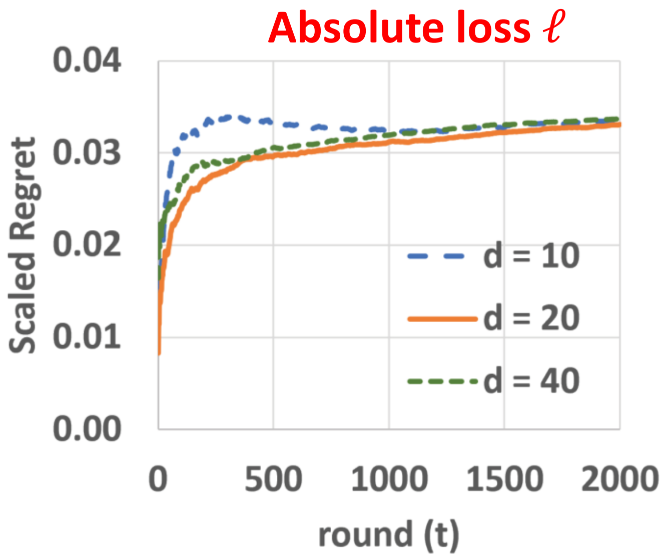
Regret = $O\left(\sqrt{dT}\right)$

Experiments:

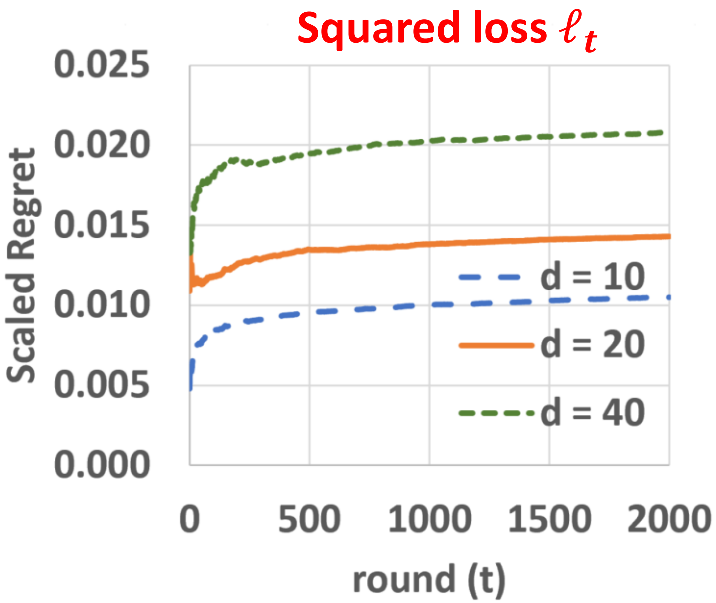
Linear g_t

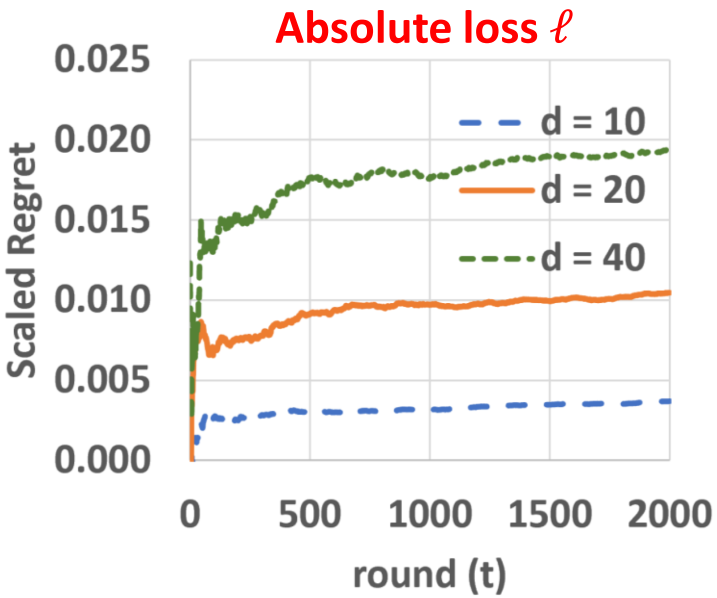
Online Gradient Descent





Linear g_t





Kernelized EXP3

In a nutshell:

- Problem formulation: Pseudo 1d bandit convex optimization
- Proposed algorithms: Design optimal algorithms + Analysis
- Understand fundamental performance limit (regret lower bound)

Future Works:

- Understand the problem complexity for higher dimensional $\ell(\quad)$?
- Can we design an unified algorithm with regret $O\left(min\left(\sqrt{dT}, T^{3/4}\right)\right)$?

Thanks!

Questions @ aasa@microsoft.com