Google Research

Learning Rate Schedules in the Presence of Distribution Shift

USC University of Southern California

Matthew Fahrbach[1], Adel Javanmard[1,2], Vahab Mirrokni[1], Pratik Worah[1]
[1]Google Research, [2]University of Southern California

Problem statement

Setup: ***Online sequential learning*** where at each time step $t \in [T]$:

1. Observe batch of B examples $\{(\mathbf{x}_{t,k}, y_{t,k})\}_{k=1}^{B}$ from distribution P_t
2. Incur loss $L_t(\theta_t) = \frac{1}{B}\sum_{k=1}^{B} \ell(f(\mathbf{x}_{t,k};\theta_t), y_{t,k})$
3. Update model weights with one step of SGD: $\theta_{t+1} \leftarrow \theta_t - \eta_t \nabla L_t(\theta_t)$

Def: **Dynamic regret** is defined w.r.t. optimal model weights at each time step:

$$\theta_t^* = \operatorname{argmin}_{\theta} \mathbb{E}_{(\mathbf{x},y)\sim P_t}[\ell(f(\mathbf{x};\theta), y)]$$

$$\mathrm{Reg}(T) = \sum_{t=1}^{T} L_t(\theta_t) - L_t(\theta_t^*)$$

Goal: Design learning rate schedule $\{\eta_t\}_{t=1}^{T}$ with bounded regret $\mathrm{Reg}(T)$ in terms of distribution shift $\gamma_t = \| \theta_t^* - \theta_{t+1}^* \|_2$.

Motivation: online deep learning recommender systems (DLRS) → same loss function ℓ, time-varying data distributions P_t

Example: Chasing a moving target

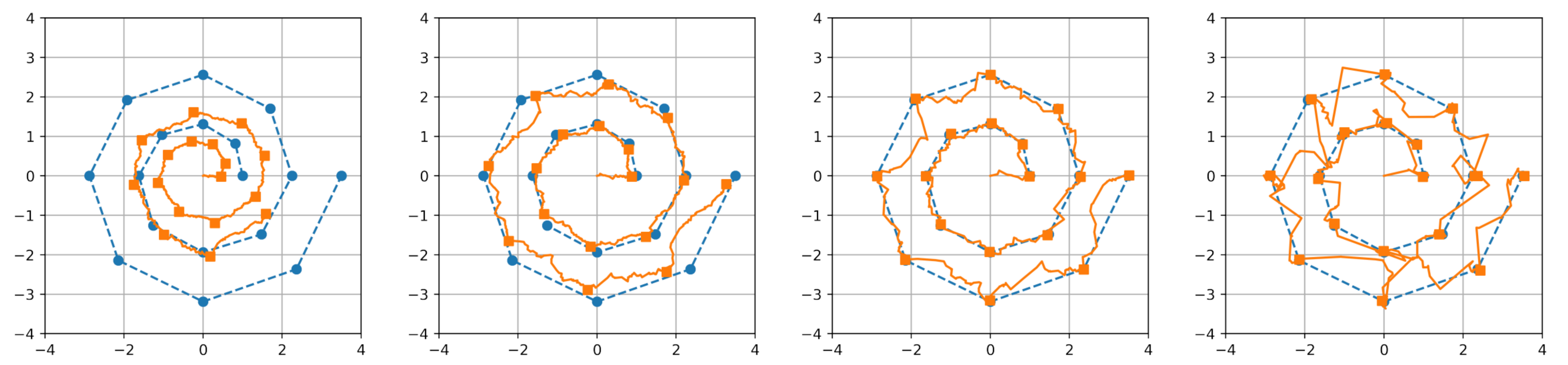


Figure 1: SGD trajectories for online linear regression with different constant learning rates. The discrete blue spirals are the optimal model weights $\theta_t^* \in \mathbb{R}^2$, which start at $(1,0)$ and jump clockwise every 100 steps. The orange paths are the learned weights θ_t, starting at $\theta_0 = 0$ for $0 \le t \le 17 \cdot 100$. The orange squares depict the position every 100 steps. We use batch size $B_t = 1$ and step sizes $\eta_t \in \{0.003, 0.01, 0.03, 0.1\}$ from left to right. The rightmost SGD is the most out of control, but it incurs the least regret because it adapts to changes in θ_t^* the fastest without diverging.

Linear regression

Time-varying coefficients model: At each time $t \in [T]$, we get B covariate-response pairs

$$y_{t,k} = \langle \mathbf{x}_{t,k}, \theta_t^* \rangle + \varepsilon_{t,k},$$

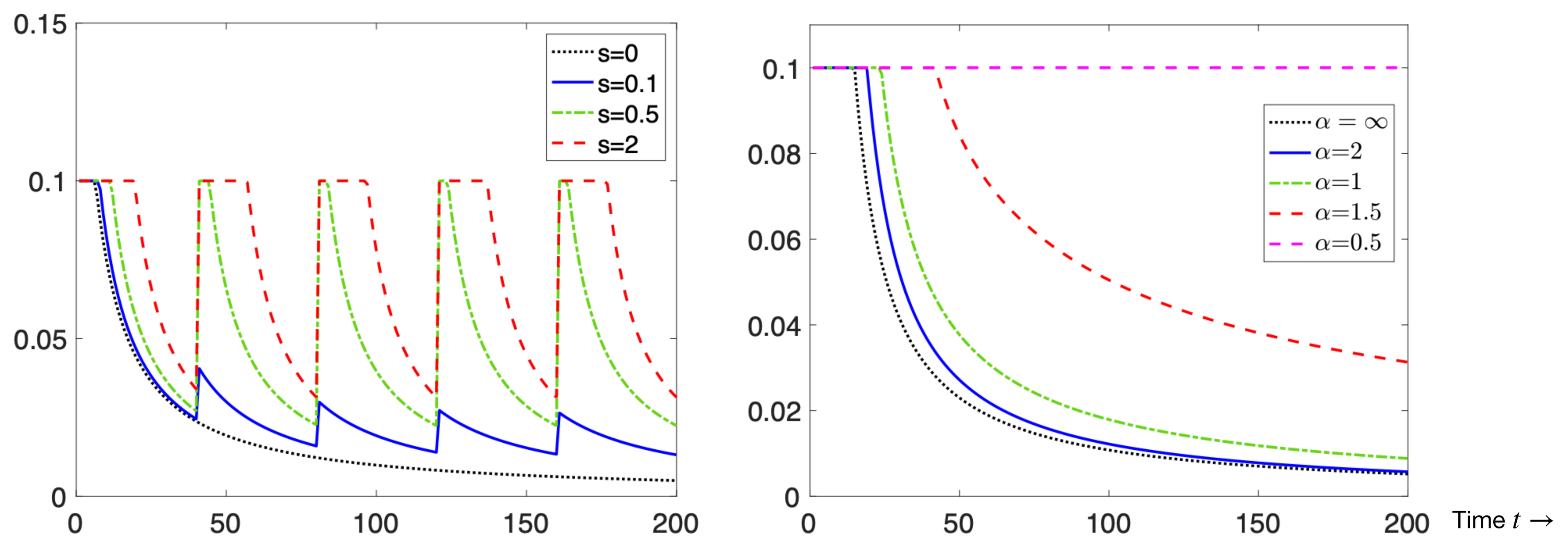
where $\mathbf{x}_{t,k} \sim N(0, \mathbf{I})$, $\varepsilon_{t,k} = N(0, \sigma^2)$ is random noise, and ℓ is ***least-squares loss***.

Approach: Analyze effect of learning rate schedule on SGD undergoing distribution shift in the continuous time-limit.

- Tools: stochastic differential equations (SDEs), Euler–Maruyama method, Itô's lemma

Main result: Solve SDE → discretize to get ***optimal online learning rate schedule***

Case studies for linear regression: Optimal learning rate schedules η_t^*



(a) Bursty distribution shifts (b) Smooth distribution shifts

Bursty shifts: Jump process where γ_t jumps to s every episode (40 steps) and then is zero for the rest of the episode. We set max step size $\eta_{max} = 0.1$.

Smooth shifts: γ_t changes continuously as $\gamma_t = 1/t^{\alpha}$ for a constant value α. Smaller values of α (i.e., larger distribution shifts) induce larger rates.

No shift: We also analyze the simplest case when there is no distribution shift and recover the optimal learning rate schedule η_t^* which is asymptotically $1/t$.

Summary of results

1. Large distribution shifts → larger learning rates
   - Insights from linear regression also apply to **general convex** and **non-convex losses**
2. We formulate the problem as dynamic regret minimization, where the target θ_t^* moves and we chase it via SGD
3. Differences with related dynamic regret works: Besbes-Bur-Zeevi (Operations Research 2015) and Yang-Zhang-Jin-Yi (ICML 2016):
   i) Supports **adaptive schedules** (vs. choosing a fixed constant step size in advance)
   ii) Supports **adaptivity in the choice of distribution** at each time step, in contrast with an arbitrary but fixed sequence of loss functions satisfying a variation budget constraint
   iii) Same loss function for **lower** and **upper bounds** → match up to constant factors

Experiments

High-dimensional regression: linear regression, logistic regression

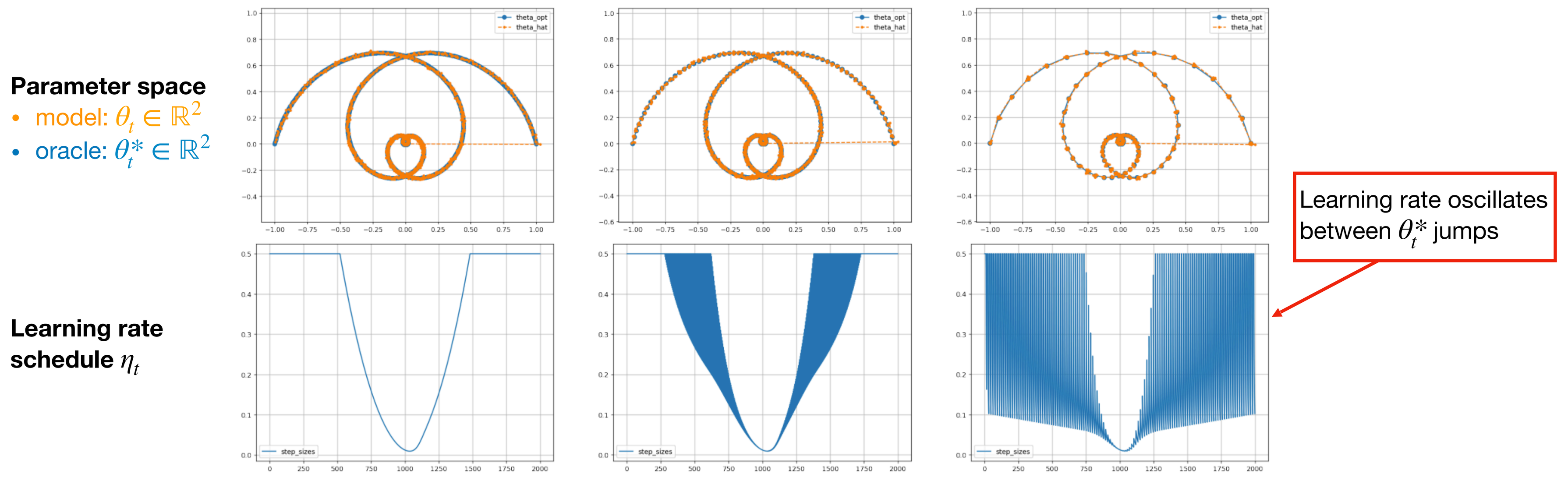


Figure 4: SGD trajectories of Algorithm 1 (top); and oscillating learning rates η_t as we discretize the path defined by θ_t^* where $\eta_{max} = 0.5$ (bottom).

Neural network application: flow cytometry → classify stream of RNA expressions that arrive from shifting data distribution