Efficient Online Learning for Dynamic Environments

Lijun Zhang

Nanjing University

CSIAM-BDAI 2018

Outline

Outline

What Happens in an Internet Minute?

http://www.dailyinfographic.com/what-happens-in-an-internet-minute-infographic

Online Learning [Shalev-Shwartz, 2011]

Online learning is the process of answering a sequence of questions given (maybe partial) knowledge of answers to previous questions and possibly additional information.

Online Learning [Shalev-Shwartz, 2011]

Online learning is the process of answering a sequence of questions given (maybe partial) knowledge of answers to previous questions and possibly additional information.

Online Learning

1: **for** $t = 1, 2, \ldots, T$ **do**

4: **end for**

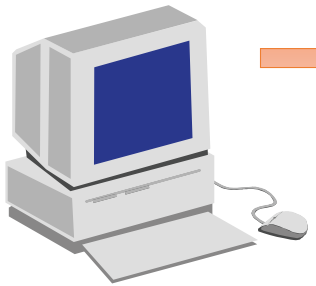
Online Learning [Shalev-Shwartz, 2011]

Online learning is the process of answering a sequence of questions given (maybe partial) knowledge of answers to previous questions and possibly additional information.

Online Learning

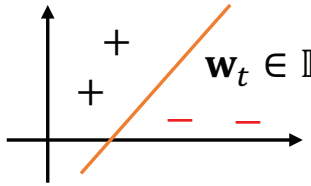
1: **for** $t = 1, 2, \ldots, T$ **do**
2: Learner picks a decision $\mathbf{w}_t \in \mathcal{W}$
Adversary chooses a function $f_t(\cdot)$

4: **end for**



Learner

A classifier



An example $(\mathbf{x}_t, y_t) \in \mathbb{R}^d \times \{\pm 1\}$

A loss $f_t(\mathbf{w}) = \max(1 - y_t \mathbf{w}^\top \mathbf{x}_t, 0)$

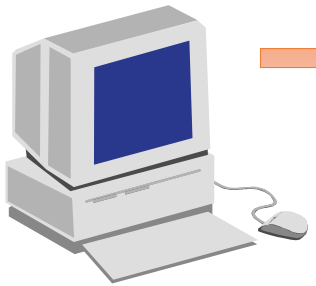
Adversary

Online Learning [Shalev-Shwartz, 2011]

Online learning is the process of answering a sequence of questions given (maybe partial) knowledge of answers to previous questions and possibly additional information.

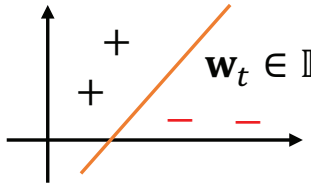
Online Learning

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Learner picks a decision $\mathbf{w}_t \in \mathcal{W}$
Adversary chooses a function $f_t(\cdot)$
3: Learner suffers loss $f_t(\mathbf{w}_t)$ and updates $\mathbf{w}_t$
4: **end for**



Learner

A classifier



An example $(\mathbf{x}_t, y_t) \in \mathbb{R}^d \times \{\pm 1\}$

A loss $f_t(\mathbf{w}) = \max(1 - y_t \mathbf{w}^\top \mathbf{x}_t, 0)$

Adversary

Online Learning [Shalev-Shwartz, 2011]

Online learning is the process of answering a sequence of questions given (maybe partial) knowledge of answers to previous questions and possibly additional information.

Online Learning

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Learner picks a decision $\mathbf{w}_t \in \mathcal{W}$
Adversary chooses a function $f_t(\cdot)$
3: Learner suffers loss $f_t(\mathbf{w}_t)$ and updates $\mathbf{w}_t$
4: **end for**

Cumulative Loss

$$\text{Cumulative Loss} = \sum_{t=1}^{T} f_t(\mathbf{w}_t)$$

Regret

Cumulative Loss

$$\text{Cumulative Loss} = \sum_{t=1}^{T} f_t(\mathbf{w}_t)$$

Regret

Cumulative Loss

$$\text{Cumulative Loss} = \sum_{t=1}^{T} f_t(\mathbf{w}_t)$$

Regret

$$\text{Regret} = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})$$

Regret

Cumulative Loss

$$\text{Cumulative Loss} = \sum_{t=1}^{T} f_t(\mathbf{w}_t)$$

Regret

$$\text{Regret} = \underbrace{\sum_{t=1}^{T} f_t(\mathbf{w}_t)}_{\text{Cumulative Loss of Online Learner}} - \underbrace{\min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})}_{\text{Minimal Loss in Offline Learner}}$$

Regret

Cumulative Loss

$$\text{Cumulative Loss} = \sum_{t=1}^{T} f_t(\mathbf{w}_t)$$

Regret

$$\text{Regret} = \underbrace{\sum_{t=1}^{T} f_t(\mathbf{w}_t)}_{\text{Cumulative Loss of Online Learner}} - \underbrace{\min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})}_{\text{Minimal Loss in Offline Learner}}$$

Hannan Consistent

$$\limsup_{T\to\infty} \frac{1}{T}\left(\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})\right) = 0, \text{ with probability 1}$$

LAMDA Learning And Mining from DatA

Regret

Cumulative Loss

$$\text{Cumulative Loss} = \sum_{t=1}^{T} f_t(\mathbf{w}_t)$$

Regret

$$\text{Regret} = \underbrace{\sum_{t=1}^{T} f_t(\mathbf{w}_t)}_{\text{Cumulative Loss of Online Learner}} - \underbrace{\min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})}_{\text{Minimal Loss in Offline Learner}}$$

Hannan Consistent

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) = o(T), \text{ with probability 1}$$

LAMDA Learning And Mining from DatA

Types of Online Learning

- Perceptron [Rosenblatt, 1958]
  - Online Classification
- Prediction with Expert Advice [Littlestone and Warmuth, 1994]
  - There are K experts in $\mathcal{W}$
  - As a meta-algorithm to combine different methods
- Online Convex Optimization [Zinkevich, 2003]
  - $f_1(\cdot), \ldots, f_T(\cdot)$ are convex
  - Online Classification, e.g., Online SVM
  - Online Regression, e.g., Online Least Squares

LAMDA Learning And Mining from DatA

Online Gradient Descent (OGD)

- Algorithm

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Learner picks a decision $\mathbf{w}_t \in \mathcal{W}$
Adversary chooses a function $f_t(\cdot)$
3: Learner suffers loss $f_t(\mathbf{w}_t)$ and

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}} \left(\mathbf{w}_t - \eta_t \nabla f_t(\mathbf{w}_t) \right)$$

4: **end for**

- The Projection Operator

$$\Pi_{\mathcal{W}}(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{w} \in \mathcal{W}} \|\mathbf{w} - \mathbf{x}\|_2$$

Theoretical Guarantee

- Convex Functions [Zinkevich, 2003]

$$f_t(\mathbf{w}) \geq f_t(\mathbf{w}') + \langle \nabla f_t(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \rangle, \ \forall \mathbf{w}, \mathbf{w}' \in \mathcal{W}$$

  - Online Gradient Descent with $\eta_t = 1/\sqrt{t}$

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \leq \frac{D^2}{2}\sqrt{T} + \left(\sqrt{T} - \frac{1}{2}\right) G^2 = O\left(\sqrt{T}\right)$$

Theoretical Guarantee

- Convex Functions [Zinkevich, 2003]

$$f_t(\mathbf{w}) \geq f_t(\mathbf{w}') + \langle \nabla f_t(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \rangle, \ \forall \mathbf{w}, \mathbf{w}' \in \mathcal{W}$$

  - Online Gradient Descent with $\eta_t = 1/\sqrt{t}$

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \leq \frac{D^2}{2}\sqrt{T} + \left(\sqrt{T} - \frac{1}{2}\right) G^2 = O\left(\sqrt{T}\right)$$

- Strongly Convex Functions [Hazan et al., 2007]

$$f_t(\mathbf{w}) \geq f_t(\mathbf{w}') + \langle \nabla f_t(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \rangle + \frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}'\|^2, \ \forall \mathbf{w}, \mathbf{w}' \in \mathcal{W}$$

  - Online Gradient Descent with $\eta_t = 1/(\lambda t)$

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \leq \frac{G^2}{2\lambda}(1 + \log T) = O(\log T)$$

  - E.g., Online SVM [Shalev-Shwartz et al., 2007]

Exponentially Concave Functions

Definition

A function $f(\cdot) : \mathcal{W} \mapsto \mathbb{R}$ is α-exp-concave if $\exp(-\alpha f(\cdot))$ is concave over $\mathcal{W}$.

- For twice differentiable functions

$$\alpha \nabla f(\mathbf{w})[\nabla f(\mathbf{w})]^\top \preceq \nabla^2 f(\mathbf{w}), \ \forall \mathbf{w} \in \mathcal{W}.$$

- Examples
  - Logistic Loss for Classification

$$f(\mathbf{w}) = \log\left(1 + \exp(-y\mathbf{x}^\top \mathbf{w})\right)$$

  - Square Loss for Regression

$$f(\mathbf{w}) = (\mathbf{x}^\top \mathbf{w} - y)^2$$

  - Negative Logarithm Loss for Portfolio Management

$$f(\mathbf{w}) = -\log(\mathbf{x}^\top \mathbf{w})$$

LAMDA Learning And Mining from DatA

Online Newton Step (ONS)

- Algorithm [Hazan et al., 2007]

1: **for** $t = 1, 2, \ldots, T$ **do**

2:

$$\begin{aligned}\mathbf{w}_{t+1} =& \Pi_{\mathcal{W}}^{A_t}\left[\mathbf{w}_t - \frac{1}{\beta}A_t^{-1}\nabla f_t(\mathbf{w}_t)\right] \\ =& \operatorname*{argmin}_{\mathbf{w}\in\mathcal{W}} \left(\mathbf{w} - \mathbf{w}'_{t+1}\right) A_t \left(\mathbf{w} - \mathbf{w}'_{t+1}\right)^\top\end{aligned}$$

where

$$A_t = A_{t-1} + \nabla f_t(\mathbf{w}_t)[\nabla f_t(\mathbf{w}_t)]^\top,\ \mathbf{w}'_{t+1} = \mathbf{w}_t - \frac{1}{\beta}A_t^{-1}\nabla f_t(\mathbf{w}_t)$$

3: **end for**

- Regret

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \leq 5\left(\frac{1}{\alpha} + GD\right) d \log T = O\left(d \log T\right)$$

LAMDA Learning And Mining from DatA

Outline

The Challenge

Regret → Static Regret

$$\begin{aligned}\text{Regret} &= \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \\ &= \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}_*)\end{aligned}$$

where $\mathbf{w}_* \in \operatorname{argmin}_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})$

- One of the decision is reasonably good during T rounds

The Challenge

Regret → Static Regret

$$\text{Regret} = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})$$
$$= \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}_*)$$

where $\mathbf{w}_* \in \operatorname{argmin}_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})$

- One of the decision is reasonably good during T rounds

Dynamic Environments

Different decisions will be good in different periods

- Recommendation: the interests of a user could change
- Stock market: the best stock changes over time

LAMDA Learning And Mining from DatA

Adaptive Regret

Adaptive Regret

[Hazan and Seshadhri, 2007, Daniely et al., 2015]

$$R(T,\tau) = \max_{[s,s+\tau-1]\subseteq[T]} \left(\sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}) \right)$$

- Minimize the static regret over all intervals of length τ

Adaptive Regret

Adaptive Regret

[Hazan and Seshadhri, 2007, Daniely et al., 2015]

$$R(T,\tau) = \max_{[s,s+\tau-1]\subseteq[T]} \left(\sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}) \right)$$

- Minimize the static regret over all intervals of length τ

$$f_1(\cdot), f_2(\cdot), \ldots, f_\tau(\cdot), f_{\tau+1}(\cdot), \ldots, f_s(\cdot), f_{s+1}(\cdot), \ldots, f_{s+\tau-1}(\cdot), f_{s+\tau}(\cdot), \ldots$$

LAMDA Learning And Mining from DatA

Adaptive Regret

Adaptive Regret

[Hazan and Seshadhri, 2007, Daniely et al., 2015]

$$R(T,\tau) = \max_{[s,s+\tau-1]\subseteq[T]} \left(\sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}) \right)$$

- Minimize the static regret over all intervals of length τ

$$\underbrace{f_1(\cdot), f_2(\cdot), \ldots, f_\tau(\cdot)}_{\sum_{t=1}^{\tau} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{\tau} f_t(\mathbf{w})}, f_{\tau+1}(\cdot), \ldots, f_s(\cdot), f_{s+1}(\cdot), \ldots, f_{s+\tau-1}(\cdot), f_{s+\tau}(\cdot), \ldots$$

LAMDA Learning And Mining from DatA

Adaptive Regret

Adaptive Regret

[Hazan and Seshadhri, 2007, Daniely et al., 2015]

$$R(T,\tau) = \max_{[s,s+\tau-1]\subseteq[T]} \left(\sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}) \right)$$

- Minimize the static regret over all intervals of length τ

$$\underbrace{f_1(\cdot), \overbrace{f_2(\cdot), \ldots, f_\tau(\cdot)}^{\sum_{t=2}^{\tau+1} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=2}^{\tau+1} f_t(\mathbf{w})}, f_{\tau+1}(\cdot)}_{\sum_{t=1}^{\tau} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{\tau} f_t(\mathbf{w})}, \ldots, f_s(\cdot), f_{s+1}(\cdot), \ldots, f_{s+\tau-1}(\cdot), f_{s+\tau}(\cdot), \ldots$$

LAMDA Learning And Mining from DatA

Adaptive Regret

Adaptive Regret

[Hazan and Seshadhri, 2007, Daniely et al., 2015]

$$R(T,\tau) = \max_{[s,s+\tau-1]\subseteq[T]} \left(\sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}) \right)$$

- Minimize the static regret over all intervals of length τ

$$\underbrace{f_1(\cdot), \overbrace{f_2(\cdot), \ldots, f_\tau(\cdot)}^{\sum_{t=2}^{\tau+1} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=2}^{\tau+1} f_t(\mathbf{w})}, f_{\tau+1}(\cdot)}_{\sum_{t=1}^{\tau} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{\tau} f_t(\mathbf{w})}, \ldots, \underbrace{f_s(\cdot), f_{s+1}(\cdot), \ldots, f_{s+\tau-1}(\cdot)}_{\sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=s}^{s+\tau-1} f_t(\mathbf{w})}, f_{s+\tau}(\cdot), \ldots$$

LAMDA Learning And Mining from DatA

Adaptive Regret

Adaptive Regret

[Hazan and Seshadhri, 2007, Daniely et al., 2015]

$$R(T,\tau) = \max_{[s,s+\tau-1]\subseteq[T]} \left(\sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}) \right)$$

- Minimize the static regret over all intervals of length τ

$$\underbrace{f_1(\cdot), \overbrace{f_2(\cdot), \ldots, f_\tau(\cdot)}^{\sum_{t=2}^{\tau+1} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=2}^{\tau+1} f_t(\mathbf{w})}, f_{\tau+1}(\cdot)}_{\sum_{t=1}^{\tau} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{\tau} f_t(\mathbf{w})}, \ldots, \underbrace{f_s(\cdot), \overbrace{f_{s+1}(\cdot), \ldots, f_{s+\tau-1}(\cdot)}^{\sum_{t=s+1}^{s+\tau} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=s+1}^{s+\tau} f_t(\mathbf{w})}, f_{s+\tau}(\cdot)}_{\sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=s}^{s+\tau-1} f_t(\mathbf{w})}, \ldots$$

LAMDA Learning And Mining from DatA

Adaptive Algorithms

- Following the Leading History

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Submit $\mathbf{w}_t \in \mathcal{W}$ and observes $f_t(\cdot)$

7: **end for**

Adaptive Algorithms

- Following the Leading History

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Submit $\mathbf{w}_t \in \mathcal{W}$ and observes $f_t(\cdot)$
3: Initialize an expert E^t by running OGD$(f_t, \ldots)$
Add E^t to the set of experts $\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{E^t\}$

7: **end for**

- Experts

$$E^1 = \mathrm{OGD}(f_1, f_2, f_3, \ldots, f_t, \ldots)$$
$$E^2 = \mathrm{OGD}(f_2, f_3, \ldots, f_t, \ldots)$$
$$E^3 = \mathrm{OGD}(f_3, \ldots, f_t, \ldots)$$
$$\ldots$$
$$E^t = \mathrm{OGD}(f_t, \ldots)$$

Adaptive Algorithms

- Following the Leading History

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Submit $\mathbf{w}_t \in \mathcal{W}$ and observes $f_t(\cdot)$
3: Initialize an expert E^t by running OGD$(f_t, \ldots)$
Add E^t to the set of experts $\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{E^t\}$
4: Get the prediction $\mathbf{w}_{t+1}^i$ for each expert $E^i \in \mathcal{S}_{t+1}$

7: **end for**

- Online Gradient Descent (OGD)

$$\mathbf{w}_{t+1}^i = \Pi_{\mathcal{W}} \left(\mathbf{w}_t^i - \eta_t \nabla f_t(\mathbf{w}_t^i) \right), \ \forall E^i \in \mathcal{S}_{t+1}$$

Adaptive Algorithms

- Following the Leading History

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Submit $\mathbf{w}_t \in \mathcal{W}$ and observes $f_t(\cdot)$
3: Initialize an expert E^t by running OGD$(f_t, \ldots)$
Add E^t to the set of experts $\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{E^t\}$
4: Get the prediction $\mathbf{w}_{t+1}^i$ for each expert $E^i \in \mathcal{S}_{t+1}$
5: Predict $\mathbf{w}_{t+1}$ by combining $\{\mathbf{w}_{t+1}^i | \mathrm{E}^i \in \mathcal{S}_{t+1}\}$
7: **end for**

- Prediction with Expert Advice
  - Exponential Weighting

$$\mathbf{w}_{t+1} = \sum_{E^i \in \mathcal{S}_{t+1}} p_{t+1}^i \mathbf{w}_{t+1}^i$$

$$p_{t+1}^i = p_t^i e^{-\alpha f_t(\mathbf{w}_t^i)}$$

LAMDA Learning And Mining from DatA

Adaptive Algorithms

- Following the Leading History

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Submit $\mathbf{w}_t \in \mathcal{W}$ and observes $f_t(\cdot)$
3: Initialize an expert E^t by running OGD$(f_t, \ldots)$
Add E^t to the set of experts $\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{E^t\}$
4: Get the prediction $\mathbf{w}_{t+1}^i$ for each expert $E^i \in \mathcal{S}_{t+1}$
5: Predict $\mathbf{w}_{t+1}$ by combining $\{\mathbf{w}_{t+1}^i | \mathrm{E}^i \in \mathcal{S}_{t+1}\}$
6: Prune the set of experts $\mathcal{S}_{t+1}$
7: **end for**

- Experts

$$E^1 = \mathrm{OGD}(f_1, f_2, f_3, \ldots, f_t, \ldots)$$
$$\sout{E^2 = \mathrm{OGD}(f_2, f_3, \ldots, f_t, \ldots)}$$
$$E^3 = \mathrm{OGD}(f_3, \ldots, f_t, \ldots)$$
$$\ldots$$
$$\sout{E^t = \mathrm{OGD}(f_t, \ldots)}$$

Theoretical Guarantee

- Convex Functions [Jun et al., 2017]

$$R(T,\tau) = O\left(\sqrt{\tau \log T}\right)$$

- Strongly Convex Functions [Zhang et al., 2018]

$$R(T,\tau) = O\left(\log \tau \log T\right)$$

- Exponentially Concave Functions [Hazan and Seshadhri, 2007]

$$R(T,\tau) = O\left(d \log \tau \log T\right)$$

LAMDA
Learning And Mining from DatA

Outline

Our Recent Work I

Regret

1. Yuanyu Wan, Nan Wei, and **Lijun Zhang**. Efficient Adaptive Online Learning via Frequent Directions. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (**IJCAI**), 2018.
2. Bo-Jian Hou, **Lijun Zhang**, and Zhi-Hua Zhou. Learning with Feature Evolvable Streams. In *Advances in Neural Information Processing Systems 30* (**NIPS**), 2017.
3. **Lijun Zhang**, Tianbao Yang, Rong Jin, Yichi Xiao, and Zhi-Hua Zhou. Online Stochastic Linear Optimization under One-bit Feedback. In *Proceedings of the 33rd International Conference on Machine Learning* (**ICML**), 2016.
4. **Lijun Zhang**, Tianbao Yang, Rong Jin, and Zhi-Hua Zhou. Online Bandit Learning for a Special Class of Non-convex Losses. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence* (**AAAI**), 2015.
5. **Lijun Zhang**, Jinfeng Yi, Rong Jin, Ming Lin, and Xiaofei He. Online Kernel Learning with a Near Optimal Sparsity Bound. In *Proceedings of the 30th International Conference on Machine Learning* (**ICML**), 2013.
6. **Lijun Zhang**, Rong Jin, Chun Chen, Jiajun Bu, and Xiaofei He. Efficient Online Learning for Large-Scale Sparse Kernel Logistic Regression. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence* (**AAAI**), 2012.

LAMDA
Learning And Mining from DatA

Our Recent Work II

Adaptive Regret

1. **Lijun Zhang**, Tianbao Yang, Rong Jin, and Zhi-Hua Zhou. Dynamic Regret of Strongly Adaptive Methods. In *Proceedings of the 35th International Conference on Machine Learning* (**ICML**), 2018.
2. Guanghui Wang, Dakuan Zhao, and **Lijun Zhang**. Minimizing Adaptive Regret with One Gradient per Iteration. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (**IJCAI**), 2018.

Dynamic Regret

1. **Lijun Zhang**, Tianbao Yang, Jinfeng Yi, Rong Jin, and Zhi-Hua Zhou. Improved Dynamic Regret for Non-degenerate Functions. In *Advances in Neural Information Processing Systems 30* (**NIPS**), 2017.
2. Tianbao Yang, **Lijun Zhang**, Rong Jin, and Jinfeng Yi. Tracking Slowly Moving Clairvoyant: Optimal Dynamic Regret of Online Learning with True and Noisy Gradient. In *Proceedings of the 33rd International Conference on Machine Learning* (**ICML**), 2016.

LAMDA
Learning And Mining from DatA

Our Recent Work II

Adaptive Regret

1. **Lijun Zhang**, Tianbao Yang, Rong Jin, and Zhi-Hua Zhou. Dynamic Regret of Strongly Adaptive Methods. In *Proceedings of the 35th International Conference on Machine Learning* (**ICML**), 2018.
2. Guanghui Wang, Dakuan Zhao, and **Lijun Zhang**. Minimizing Adaptive Regret with One Gradient per Iteration. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (**IJCAI**), 2018.

Dynamic Regret

1. **Lijun Zhang**, Tianbao Yang, Jinfeng Yi, Rong Jin, and Zhi-Hua Zhou. Improved Dynamic Regret for Non-degenerate Functions. In *Advances in Neural Information Processing Systems 30* (**NIPS**), 2017.
2. Tianbao Yang, **Lijun Zhang**, Rong Jin, and Jinfeng Yi. Tracking Slowly Moving Clairvoyant: Optimal Dynamic Regret of Online Learning with True and Noisy Gradient. In *Proceedings of the 33rd International Conference on Machine Learning* (**ICML**), 2016.

LAMDA
Learning And Mining from DatA

Adaptive Algorithms

- Following the Leading History

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Submit $\mathbf{w}_t \in \mathcal{W}$ and observes $f_t(\cdot)$
3: Initialize an expert E^t by running OGD$(f_t, \ldots)$
Add E^t to the set of experts $\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{E^t\}$
4: Get the prediction $\mathbf{w}_{t+1}^i$ for each expert $E^i \in \mathcal{S}_{t+1}$
5: Predict $\mathbf{w}_{t+1}$ by combining $\{\mathbf{w}_{t+1}^i | \mathrm{E}^i \in \mathcal{S}_{t+1}\}$
6: Prune the set of experts $\mathcal{S}_{t+1}$
7: **end for**

- Online Gradient Descent (OGD)

$$\mathbf{w}_{t+1}^i = \Pi_{\mathcal{W}} \left(\mathbf{w}_t^i - \eta_t \nabla f_t(\mathbf{w}_t^i) \right), \ \forall E^i \in \mathcal{S}_{t+1}$$

Adaptive Algorithms

- Following the Leading History

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Submit $\mathbf{w}_t \in \mathcal{W}$ and observes $f_t(\cdot)$
3: Initialize an expert E^t by running OGD$(f_t, \ldots)$
Add E^t to the set of experts $\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{E^t\}$
4: Get the prediction $\mathbf{w}_{t+1}^i$ for each expert $E^i \in \mathcal{S}_{t+1}$
5: Predict $\mathbf{w}_{t+1}$ by combining $\{\mathbf{w}_{t+1}^i | \mathrm{E}^i \in \mathcal{S}_{t+1}\}$
6: Prune the set of experts $\mathcal{S}_{t+1}$
7: **end for**

- Online Gradient Descent (OGD)

$$\mathbf{w}_{t+1}^i = \Pi_{\mathcal{W}} \left(\mathbf{w}_t^i - \eta_t \nabla f_t(\mathbf{w}_t^i) \right), \ \forall E^i \in \mathcal{S}_{t+1}$$

- Computational Cost per Iteration

$|\mathcal{S}_{t+1}|$ gradient evaluations of $f_t(\cdot)$

where $|\mathcal{S}_{t+1}| = O(\log t)$

Gradient Evaluations

- Nuclear-norm Regularized Losses

$$f_t(W) = \ell_t(W) + \lambda \|W\|_*$$

where $W \in \mathbb{R}^{m \times n}$

  - Low-rank matrix regression
  - Low-rank matrix approximation
  - Low-rank multiclass classification

Gradient evaluations are expensive when m and n are large

- Mini-batch Losses

$$f_t(\mathbf{w}) = \frac{1}{k} \sum_{i=1}^{k} \ell(\mathbf{w}^\top \mathbf{x}_t^i, y_t^i)$$

Gradient evaluations are expensive when k is large

LAMDA Learning And Mining from DatA

Online Learning with Surrogate Loss

- Following the Leading History [Wang et al., 2018]

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Submit $\mathbf{w}_t \in \mathcal{W}$ and observes $f_t(\cdot)$
3: Construct a surrogate loss $\ell_t(\cdot)$ from $\nabla f_t(\mathbf{w}_t)$
4: Initialize an expert E^t by running OGD$(\ell_t, \ldots)$
Add E^t to the set of experts $\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{E^t\}$
5: Get the prediction $\mathbf{w}_{t+1}^i$ for each expert $E^i \in \mathcal{S}_{t+1}$
6: Predict $\mathbf{w}_{t+1}$ by combining $\{\mathbf{w}_{t+1}^i | \mathrm{E}^i \in \mathcal{S}_{t+1}\}$
7: Prune the set of experts $\mathcal{S}_{t+1}$
8: **end for**

Online Learning with Surrogate Loss

- Following the Leading History [Wang et al., 2018]

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Submit $\mathbf{w}_t \in \mathcal{W}$ and observes $f_t(\cdot)$
3: Construct a surrogate loss $\ell_t(\cdot)$ from $\nabla f_t(\mathbf{w}_t)$
4: Initialize an expert E^t by running OGD$(\ell_t, \ldots)$
Add E^t to the set of experts $\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{E^t\}$
5: Get the prediction $\mathbf{w}_{t+1}^i$ for each expert $E^i \in \mathcal{S}_{t+1}$
6: Predict $\mathbf{w}_{t+1}$ by combining $\{\mathbf{w}_{t+1}^i | \mathrm{E}^i \in \mathcal{S}_{t+1}\}$
7: Prune the set of experts $\mathcal{S}_{t+1}$
8: **end for**

- Experts

$$E^1 = \mathrm{OGD}(\ell_1, \ell_2, \ell_3, \ldots, \ell_t, \ldots)$$
$$E^2 = \mathrm{OGD}(\ell_2, \ell_3, \ldots, \ell_t, \ldots)$$
$$\ldots$$
$$E^t = \mathrm{OGD}(\ell_t, \ldots)$$

LAMDA Learning And Mining from DatA

Online Learning with Surrogate Loss

- Following the Leading History [Wang et al., 2018]

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Submit $\mathbf{w}_t \in \mathcal{W}$ and observes $f_t(\cdot)$
3: Construct a surrogate loss $\ell_t(\cdot)$ from $\nabla f_t(\mathbf{w}_t)$
4: Initialize an expert E^t by running OGD$(\ell_t, \ldots)$
Add E^t to the set of experts $\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{E^t\}$
5: Get the prediction $\mathbf{w}_{t+1}^i$ for each expert $E^i \in \mathcal{S}_{t+1}$
6: Predict $\mathbf{w}_{t+1}$ by combining $\{\mathbf{w}_{t+1}^i | \mathrm{E}^i \in \mathcal{S}_{t+1}\}$
7: Prune the set of experts $\mathcal{S}_{t+1}$
8: **end for**

- Online Gradient Descent (OGD)

$$\mathbf{w}_{t+1}^i = \Pi_{\mathcal{W}} \left(\mathbf{w}_t^i - \eta_t \nabla \ell_t(\mathbf{w}_t^i) \right), \ \forall E^i \in \mathcal{S}_{t+1}$$

Online Learning with Surrogate Loss

- Following the Leading History [Wang et al., 2018]

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Submit $\mathbf{w}_t \in \mathcal{W}$ and observes $f_t(\cdot)$
3: Construct a surrogate loss $\ell_t(\cdot)$ from $\nabla f_t(\mathbf{w}_t)$
4: Initialize an expert E^t by running OGD$(\ell_t, \ldots)$
Add E^t to the set of experts $\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{E^t\}$
5: Get the prediction $\mathbf{w}_{t+1}^i$ for each expert $E^i \in \mathcal{S}_{t+1}$
6: Predict $\mathbf{w}_{t+1}$ by combining $\{\mathbf{w}_{t+1}^i | \mathrm{E}^i \in \mathcal{S}_{t+1}\}$
7: Prune the set of experts $\mathcal{S}_{t+1}$
8: **end for**

- Online Gradient Descent (OGD)

$$\mathbf{w}_{t+1}^i = \Pi_{\mathcal{W}} \left(\mathbf{w}_t^i - \eta_t \nabla \ell_t(\mathbf{w}_t^i) \right), \ \forall E^i \in \mathcal{S}_{t+1}$$

- Computational Cost per Iteration

1 gradient evaluation of $f_t(\cdot)$

LAMDA Learning And Mining from DatA

Convex Functions

- First-order Condition

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}) \leq -\langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle$$

Convex Functions

- First-order Condition

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}) \leq -\langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle$$

- Surrogate Loss

$$\ell_t(\mathbf{w}) = \langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle$$

which is convex

Convex Functions

- First-order Condition

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}) \leq -\langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle$$

- Surrogate Loss

$$\ell_t(\mathbf{w}) = \langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle$$

which is convex

- Properties
  - Gradient evaluation is easy

$$\nabla \ell_t(\mathbf{w}_t^i) = \nabla f_t(\mathbf{w}_t)$$

  - The regret bound is maintained

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}) \leq \ell_t(\mathbf{w}_t) - \ell_t(\mathbf{w})$$

Convex Functions

- First-order Condition

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}) \leq -\langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle$$

- Surrogate Loss

$$\ell_t(\mathbf{w}) = \langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle$$

which is convex

- Properties
  - Gradient evaluation is easy

$$\nabla \ell_t(\mathbf{w}_t^i) = \nabla f_t(\mathbf{w}_t)$$

  - The regret bound is maintained

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}) \leq \ell_t(\mathbf{w}_t) - \ell_t(\mathbf{w})$$

- Adaptive Regret

$$R(T, \tau) = O\left(\sqrt{\tau \log T}\right)$$

LAMDA Learning And Mining from DatA

Strongly Convex Functions

- First-order Condition

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}) \leq -\langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle - \frac{\lambda}{2} \|\mathbf{w} - \mathbf{w}_t\|^2$$

Strongly Convex Functions

- First-order Condition

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}) \leq -\langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle - \frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}_t\|^2$$

- Surrogate Loss

$$\ell_t(\mathbf{w}) = \langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle + \frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}_t\|^2$$

which is strongly convex

Strongly Convex Functions

- First-order Condition

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}) \leq -\langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle - \frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}_t\|^2$$

- Surrogate Loss

$$\ell_t(\mathbf{w}) = \langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle + \frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}_t\|^2$$

which is strongly convex

- Properties
  - Gradient evaluation is easy

$$\nabla \ell_t(\mathbf{w}_t^i) = \nabla f_t(\mathbf{w}_t) + \lambda(\mathbf{w}_t^i - \mathbf{w}_t)$$

  - The regret bound is maintained

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}) \leq \ell_t(\mathbf{w}_t) - \ell_t(\mathbf{w})$$

Strongly Convex Functions

- First-order Condition

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}) \leq -\langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle - \frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}_t\|^2$$

- Surrogate Loss

$$\ell_t(\mathbf{w}) = \langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle + \frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}_t\|^2$$

which is strongly convex

- Properties
  - Gradient evaluation is easy

$$\nabla \ell_t(\mathbf{w}_t^i) = \nabla f_t(\mathbf{w}_t) + \lambda(\mathbf{w}_t^i - \mathbf{w}_t)$$

  - The regret bound is maintained

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}) \leq \ell_t(\mathbf{w}_t) - \ell_t(\mathbf{w})$$

- Adaptive Regret

$$R(T, \tau) = O(\log \tau \log T)$$

Exponentially Concave Functions

- First-order Condition [Hazan et al., 2007]

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}) \leq -\langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle - \frac{\beta}{2} \left(\langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle \right)^2$$

- Surrogate Loss

$$\ell_t(\mathbf{w}) = \langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle + \frac{\beta}{2} \left(\langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle \right)^2$$

which is exponentially concave

- Properties
  - Gradient evaluation is easy

$$\nabla \ell_t(\mathbf{w}_t^i) = \nabla f_t(\mathbf{w}_t) + \beta \langle \nabla f_t(\mathbf{w}_t), \mathbf{w}_t^i - \mathbf{w}_t \rangle \nabla f_t(\mathbf{w}_t)$$

  - The regret bound is maintained

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}) \leq \ell_t(\mathbf{w}_t) - \ell_t(\mathbf{w})$$

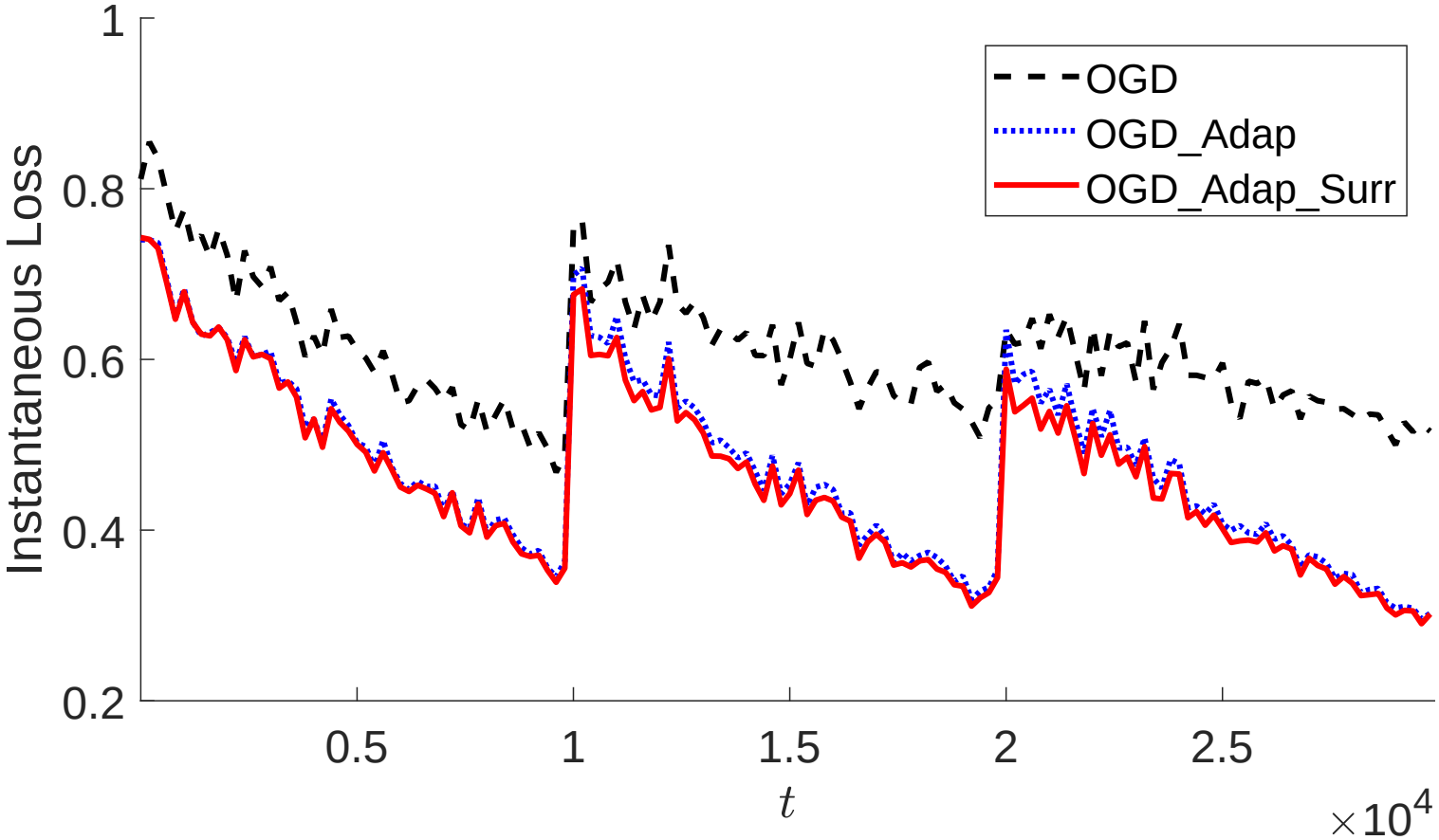
- Adaptive Regret

$$R(T, \tau) = O\left(d \log \tau \log T\right)$$

Experiments

- Nuclear-norm Regularized Matrix Regression

$$f_t(W) = \frac{1}{2}\left(y_t - \text{trace}(W^\top X_t)\right)^2 + b\|W\|_*$$

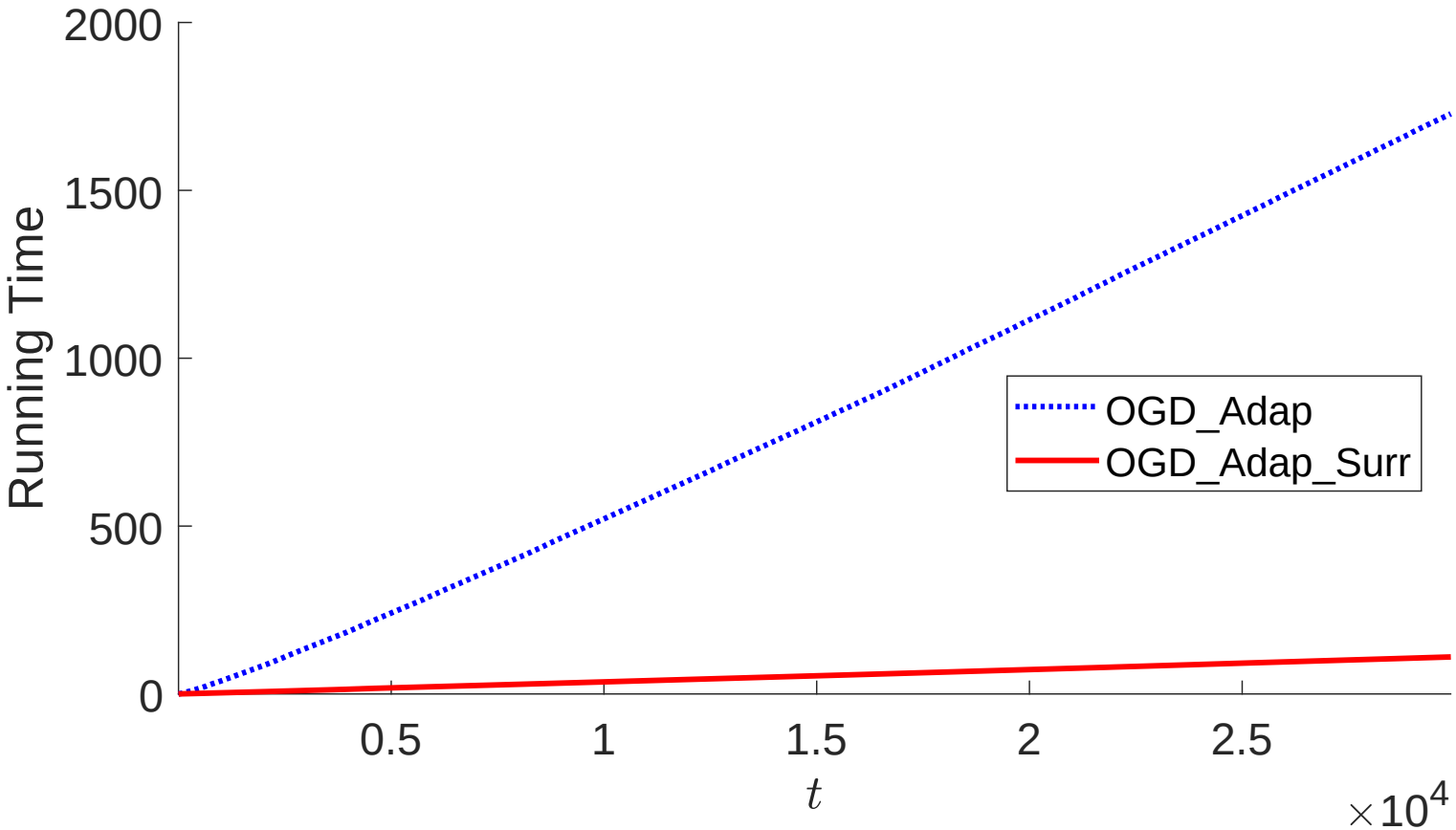


- $y_t = \text{trace}(W_*^\top X_t) + \epsilon_t$, where W_* changes twice

Experiments

- Nuclear-norm Regularized Matrix Regression

$$f_t(W) = \frac{1}{2}\left(y_t - \text{trace}(W^\top X_t)\right)^2 + b\|W\|_*$$

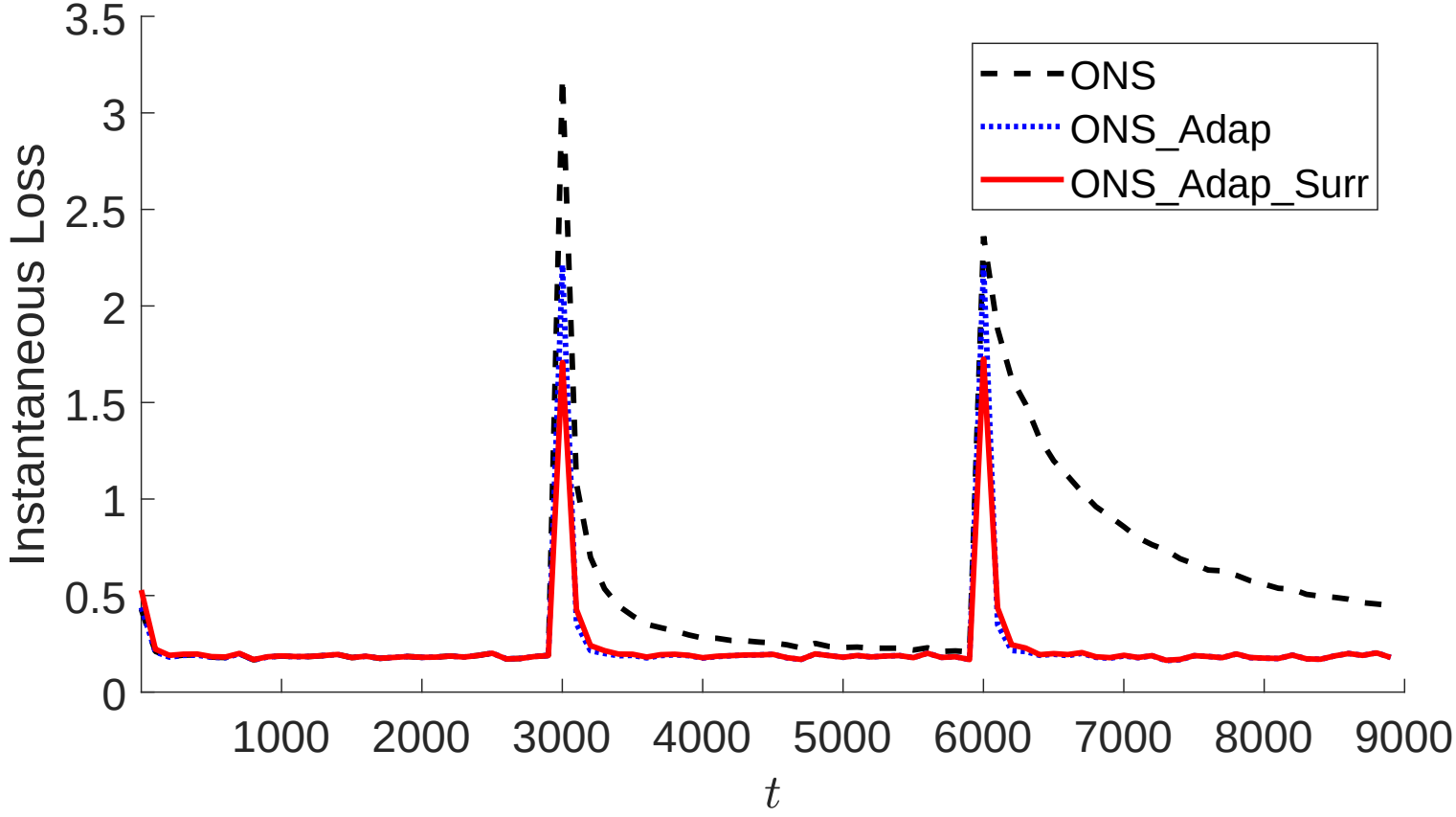


- $y_t = \text{trace}(W_*^\top X_t) + \epsilon_t$, where W_* changes twice

Experiments

- Mini-batch Logistic Regression

$$f_t(\mathbf{w}) = \frac{1}{k}\sum_{i=1}^{k} \log\left(1+\exp\left(-y_t^i \mathbf{w}^\top \mathbf{x}_t^i\right)\right)$$

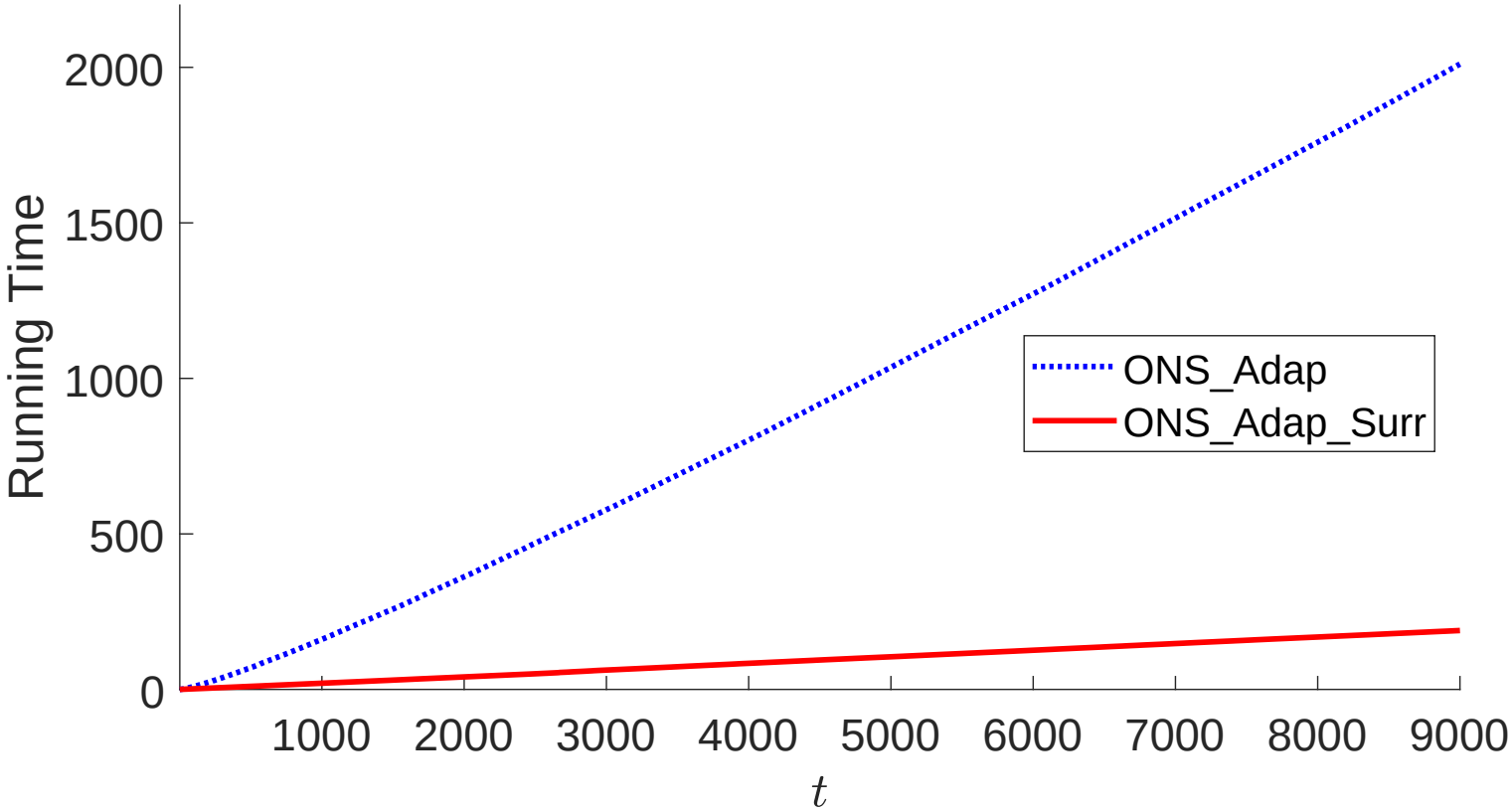


- the IJCNN01 dataset, where labels are flipped twice

Experiments

- Mini-batch Logistic Regression

$$f_t(\mathbf{w}) = \frac{1}{k}\sum_{i=1}^{k} \log\left(1+\exp\left(-y_t^i \mathbf{w}^\top \mathbf{x}_t^i\right)\right)$$



- the IJCNN01 dataset, where labels are flipped twice

Outline

Conclusion and Future Work

Conclusion

- A brief introduction to online learning
- Efficient algorithms for adaptive regret
  - 1 gradient evaluation per iteration
- Empirical evaluations of the proposed algorithms

Future Work

- Remove the $\log T$ factor in adaptive regret
- Other metric (e.g., dynamic regret) for dynamic environments
- Without convexity

LAMDA
Learning And Mining from DatA

Reference I

Thanks!

Daniely, A., Gonen, A., and Shalev-Shwartz, S. (2015).
Strongly adaptive online learning.
In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1405–1411.

Hazan, E., Agarwal, A., and Kale, S. (2007).
Logarithmic regret algorithms for online convex optimization.
Machine Learning, 69(2-3):169–192.

Hazan, E. and Seshadhri, C. (2007).
Adaptive algorithms for online decision problems.
Electronic Colloquium on Computational Complexity, 88.

Jun, K.-S., Orabona, F., Wright, S., and Willett, R. (2017).
Improved Strongly Adaptive Online Learning using Coin Betting.
In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 943–951.

Littlestone, N. and Warmuth, M. K. (1994).
The weighted majority algorithm.
Information and Computation, 108(2):212–261.

Rosenblatt, F. (1958).
The perceptron: a probabilistic model for information storage and organization in the brain.
Psychological Review, 65:386–407.

LAMDA
Learning And Mining from DatA

Reference II

Shalev-Shwartz, S. (2011).

Online learning and online convex optimization.
Foundations and Trends in Machine Learning, 4(2):107–194.

Shalev-Shwartz, S., Singer, Y., and Srebro, N. (2007).

Pegasos: primal estimated sub-gradient solver for SVM.
In *Proceedings of the 24th International Conference on Machine Learning*, pages 807–814.

Wang, G., Zhao, D., and Zhang, L. (2018).

Minimizing adaptive regret with one gradient per iteration.
In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*.

Zhang, L., Yang, T., Jin, R., and Zhou, Z.-H. (2018).

Dynamic regret of strongly adaptive methods.
In *Proceedings of the 35th International Conference on Machine Learning*.

Zinkevich, M. (2003).

Online convex programming and generalized infinitesimal gradient ascent.
In *Proceedings of the 20th International Conference on Machine Learning*, pages 928–936.

LAMDA
Learning And Mining from DatA