Efficient Online Learning for Dynamic Environments

Lijun Zhang

Nanjing University

CSIAM-BDAI 2018
Outline

1. Introduction
2. Adaptive Regret for Dynamic Environments
3. Efficient Algorithms for Adaptive Regret
4. Conclusion
Outline

1. Introduction
2. Adaptive Regret for Dynamic Environments
3. Efficient Algorithms for Adaptive Regret
4. Conclusion
What Happens in an Internet Minute?

And Future Growth is Staggering

Today, the number of networked devices = the global population
By 2015, the number of networked devices = 2x the global population
In 2015, it would take you 5 years to view all video crossing IP networks each second


http://cs.nju.edu.cn/zlj
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 \( w_t \in \mathcal{W} \)
   Adversary chooses a function \( f_t(\cdot) \)
4: end for

A classifier

\[
A \text{ example } (x_t, y_t) \in \mathbb{R}^d \times \{\pm 1\}
\]

A loss \( f_t(w) = \max(1 - y_t w^T x_t, 0) \)
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 \( w_t \in \mathcal{W} \)
   Adversary chooses a function \( f_t(\cdot) \)
3: Learner suffers loss \( f_t(w_t) \) and updates \( w_t \)
4: end for

A classifier \( \mathcal{W} \) \( w_t \in \mathbb{R}^d \)

An example \( (x_t, y_t) \in \mathbb{R}^d \times \{\pm 1\} \)
A loss \( f_t(w) = \max(1 - y_t w^T x_t, 0) \)
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: \hspace{1em} Learner picks a decision \( w_t \in \mathcal{W} \)
3: \hspace{1em} Adversary chooses a function \( f_t(\cdot) \)
4: \hspace{1em} Learner suffers loss \( f_t(w_t) \) and updates \( w_t \)
5: end for

Cumulative Loss

\[
\text{Cumulative Loss} = \sum_{t=1}^{T} f_t(w_t)
\]
Cumulative Loss

\[
\text{Cumulative Loss} = \sum_{t=1}^{T} f_t(w_t)
\]
Regret

Cumulative Loss

\[ \text{Cumulative Loss} = \sum_{t=1}^{T} f_t(w_t) \]

Regret

\[ \text{Regret} = \sum_{t=1}^{T} f_t(w_t) - \min_{w \in W} \sum_{t=1}^{T} f_t(w) \]
Regret

Cumulative Loss

Cumulative Loss $= \sum_{t=1}^{T} f_t(w_t)$

Regret

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

Cumulative Loss of Online Learner

Minimal Loss in Offline Learner
Regret

Cumulative Loss

\[
\text{Cumulative Loss} = \sum_{t=1}^{T} f_t(w_t)
\]

Regret

\[
\text{Regret} = \sum_{t=1}^{T} f_t(w_t) - \min_{w \in W} \sum_{t=1}^{T} f_t(w)
\]

Hannan Consistent

\[
\limsup_{T \to \infty} \frac{1}{T} \left( \sum_{t=1}^{T} f_t(w_t) - \min_{w \in W} \sum_{t=1}^{T} f_t(w) \right) = 0, \text{ with probability } 1
\]
Regret

Cumulative Loss

\[ \text{Cumulative Loss} = \sum_{t=1}^{T} f_t(w_t) \]

Regret

\[ \text{Regret} = \sum_{t=1}^{T} f_t(w_t) - \min_{w \in \mathcal{W}} \sum_{t=1}^{T} f_t(w) \]

Cumulative Loss of Online Learner

Minimal Loss in Offline Learner

Hannan Consistent

\[ \sum_{t=1}^{T} f_t(w_t) - \min_{w \in \mathcal{W}} \sum_{t=1}^{T} f_t(w) = o(T), \text{ with probability 1} \]
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
Online Gradient Descent (OGD)

Algorithm

1: for \( t = 1, 2, \ldots, T \) do
2: Learner picks a decision \( w_t \in \mathcal{W} \)
   Adversary chooses a function \( f_t(\cdot) \)
3: Learner suffers loss \( f_t(w_t) \) and
   \[
   w_{t+1} = \Pi_{\mathcal{W}}(w_t - \eta_t \nabla f_t(w_t))
   \]
4: end for

The Projection Operator

\[
\Pi_{\mathcal{W}}(x) = \arg\min_{w \in \mathcal{W}} \| w - x \|_2
\]

http://cs.nju.edu.cn/zlj
Theoretical Guarantee

- Convex Functions [Zinkevich, 2003]
  \[ f_t(w) \geq f_t(w') + \langle \nabla f_t(w'), w - w' \rangle, \forall w, w' \in \mathcal{W} \]

- Online Gradient Descent with \( \eta_t = 1/\sqrt{t} \)
  \[ \sum_{t=1}^{T} f_t(w_t) - \min_{w \in \mathcal{W}} \sum_{t=1}^{T} f_t(w) \leq \frac{D^2}{2} \sqrt{T} + \left( \sqrt{T} - \frac{1}{2} \right) G^2 = O\left( \sqrt{T} \right) \]
### Theoretical Guarantee

**Convex Functions** [Zinkevich, 2003]

\[ f_t(w) \geq f_t(w') + \langle \nabla f_t(w'), w - w' \rangle, \quad \forall w, w' \in \mathcal{W} \]

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

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

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

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

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

\[
\sum_{t=1}^{T} f_t(w_t) - \min_{w \in \mathcal{W}} \sum_{t=1}^{T} f_t(w) \leq \frac{G^2}{2\lambda} (1 + \log T) = O\left( \log T \right)
\]

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

**Definition**

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

- For twice differentiable functions
  \[
  \alpha \nabla f(w)[\nabla f(w)]^\top \preceq \nabla^2 f(w), \forall w \in \mathcal{W}.
  \]

- **Examples**
  - Logistic Loss for Classification
    \[
    f(w) = \log \left(1 + \exp(-y x^\top w)\right)
    \]
  - Square Loss for Regression
    \[
    f(w) = (x^\top w - y)^2
    \]
  - Negative Logarithm Loss for Portfolio Management
    \[
    f(w) = -\log(x^\top w)
    \]
Online Newton Step (ONS)

**Algorithm** [Hazan et al., 2007]

1. for $t = 1, 2, \ldots, T$ do
2. $w_{t+1} = \Pi_{\mathcal{W}}^{A_t} \left[ w_t - \frac{1}{\beta} A_t^{-1} \nabla f_t(w_t) \right]$

   $= \arg\min_{w \in \mathcal{W}} (w - w'_{t+1}) A_t (w - w'_{t+1})^T$

   where

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

3. end for

**Regret**

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

1. Introduction
2. Adaptive Regret for Dynamic Environments
3. Efficient Algorithms for Adaptive Regret
4. Conclusion
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 \( \mathbf{w}_* \in \arg\min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \)

● One of the decision is reasonably good during \( T \) rounds
Introduction  Adaptive Regret  Efficient Algorithms  Conclusion

The Challenge

Regret → Static Regret

\[
\text{Regret} = \sum_{t=1}^{T} f_t(w_t) - \min_{w \in W} \sum_{t=1}^{T} f_t(w)
\]

\[
\quad = \sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(w^*)
\]

where \( w^* \in \arg\min_{w \in W} \sum_{t=1}^{T} f_t(w) \)

- 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

http://cs.nju.edu.cn/zlj  Online Learning
Adaptive Regret

Adaptive Regret

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

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

- Minimize the static regret over all intervals of length \( \tau \)
Adaptive Regret

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

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

- Minimize the static regret over all intervals of length \( \tau \)

\[ f_1(\cdot), f_2(\cdot), \ldots, f_\tau(\cdot), f_{\tau+1}(\cdot), \ldots, f_s(\cdot), f_{s+1}(\cdot), \ldots, f_{s+\tau-1}(\cdot), f_{s+\tau}(\cdot), \ldots \]
Adaptive Regret

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

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

- Minimize the static regret over all intervals of length \( \tau \)

\[ \sum_{t=1}^{\tau} f_t(w_t) - \min_{w \in W} \sum_{t=1}^{\tau} f_t(w) \]
Adaptive Regret

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

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

Minimize the static regret over all intervals of length $\tau$

\[
\sum_{t=2}^{\tau+1} f_t(w_t) - \min_{w \in \mathcal{W}} \sum_{t=2}^{\tau+1} f_t(w) \]

\[
\sum_{t=1}^{\tau} f_t(w_t) - \min_{w \in \mathcal{W}} \sum_{t=1}^{\tau} f_t(w) \]
Adaptive Regret

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

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

- Minimize the static regret over all intervals of length \( \tau \)

\[ \sum_{t=1}^{\tau} f_t(w_t) - \min_{w \in \mathcal{W}} \sum_{t=1}^{\tau} f_t(w) \]

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

\[ \sum_{t=s}^{s+\tau-1} f_t(w_t) - \min_{w \in \mathcal{W}} \sum_{t=s}^{s+\tau-1} f_t(w) \]
Adaptive Regret

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

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

- Minimize the static regret over all intervals of length \( \tau \)

\[ \sum_{t=2}^{\tau+1} f_t(w_t) - \min_{w \in \mathcal{W}} \sum_{t=2}^{\tau+1} f_t(w) \]
\[ f_1(\cdot), f_2(\cdot), \ldots, f_\tau(\cdot), f_{\tau+1}(\cdot), \ldots, f_s(\cdot), f_{s+1}(\cdot), \ldots, f_{s+\tau-1}(\cdot), f_{s+\tau}(\cdot), \ldots \]

\[ \sum_{t=1}^{\tau} f_t(w_t) - \min_{w \in \mathcal{W}} \sum_{t=1}^{\tau} f_t(w) \]

\[ \sum_{t=s+1}^{s+\tau} f_t(w_t) - \min_{w \in \mathcal{W}} \sum_{t=s+1}^{s+\tau} f_t(w) \]

\[ \sum_{t=s}^{s+\tau-1} f_t(w_t) - \min_{w \in \mathcal{W}} \sum_{t=s}^{s+\tau-1} f_t(w) \]
Adaptive Algorithms

- Following the Leading History

1: for \( t = 1, 2, \ldots, T \) do
2: \( \text{Submit } w_t \in \mathcal{W} \text{ and observes } f_t(\cdot) \)

7: end for
Adaptive Algorithms

Following the Leading History

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

7: end for

Experts

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

- Following the Leading History
  1: \textbf{for} \( t = 1, 2, \ldots, T \) \textbf{do}
  2: \hspace{1em} Submit \( w_t \in \mathcal{W} \) and observes \( f_t(\cdot) \)
  3: \hspace{1em} Initialize an expert \( E^t \) by running OGD(\( f_t, \ldots \))
      Add \( E^t \) to the set of experts \( S_{t+1} = S_t \cup \{ E^t \} \)
  4: \hspace{1em} Get the prediction \( w^i_{t+1} \) for each expert \( E^i \in S_{t+1} \)
  7: \textbf{end for}

- Online Gradient Descent (OGD)
  \[
  w^i_{t+1} = \Pi_{\mathcal{W}} \left( w_t^i - \eta_t \nabla f_t(w_t^i) \right), \forall E^i \in S_{t+1}
  \]
Adaptive Algorithms

- Following the Leading History
  1: **for** $t = 1, 2, \ldots, T$ **do**
  2: Submit $w_t \in \mathcal{W}$ 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 $S_{t+1} = S_t \cup \{E^t\}$
  4: Get the prediction $w^i_{t+1}$ for each expert $E^i \in S_{t+1}$
  5: Predict $w_{t+1}$ by combining $\{w^i_{t+1} | E^i \in S_{t+1}\}$

- Prediction with Expert Advice
  - Exponential Weighting
    \[
    w_{t+1} = \sum_{E^i \in S_{t+1}} p^i_{t+1} w^i_{t+1}
    \]
    \[
    p^i_{t+1} = p^i_t e^{-\alpha_t f_t(w^i_t)}
    \]
Adaptive Algorithms

- Following the Leading History

1: for $t = 1, 2, \ldots, T$ do
2:   Submit $w_t \in \mathcal{W}$ 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 $S_{t+1} = S_t \cup \{E^t\}$
4:   Get the prediction $w^i_{t+1}$ for each expert $E^i \in S_{t+1}$
5:   Predict $w_{t+1}$ by combining $\{w^i_{t+1}|E^i \in S_{t+1}\}$
6:   Prune the set of experts $S_{t+1}$
7: end for

- Experts

$$E^1 = \text{OGD}(f_1, f_2, f_3, \ldots, f_t, \ldots)$$
$$E^2 = \text{OGD}(f_2, f_3, \ldots, f_t, \ldots)$$
$$E^3 = \text{OGD}(f_3, \ldots, f_t, \ldots)$$
$$\ldots$$
$$E^t = \text{OGD}(f_t, \ldots)$$
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]
  \[ R(T, \tau) = O(\log \tau \log T) \]

- Exponentially Concave Functions [Hazan and Seshadhri, 2007]
  \[ R(T, \tau) = O(d \log \tau \log T) \]
Our Recent Work I

**Regret**


---

http://cs.nju.edu.cn/zlj Online Learning
Our Recent Work II

Adaptive Regret


Dynamic Regret


Our Recent Work II

**Adaptive Regret**


**Dynamic Regret**


Adaptive Algorithms

- **Following the Leading History**

1. **for** $t = 1, 2, \ldots, T$ **do**
2. Submit $w_t \in \mathcal{W}$ 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 $S_{t+1} = S_t \cup \{E^t\}$
4. Get the prediction $w^{i}_{t+1}$ for each expert $E^i \in S_{t+1}$
5. Predict $w_{t+1}$ by combining $\{w^{i}_{t+1}|E^i \in S_{t+1}\}$
6. Prune the set of experts $S_{t+1}$
7. **end for**

- **Online Gradient Descent (OGD)**

\[
    w^{i}_{t+1} = \prod_{\mathcal{W}} \left( w^{i}_t - \eta_t \nabla f_t(w^{i}_t) \right), \quad \forall E^i \in S_{t+1}
\]
Adaptive Algorithms

- Following the Leading History
  1: for $t = 1, 2, \ldots, T$ do
  2: Submit $w_t \in \mathcal{W}$ 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 $S_{t+1} = S_t \cup \{E^t\}$
  4: Get the prediction $w^i_{t+1}$ for each expert $E^i \in S_{t+1}$
  5: Predict $w_{t+1}$ by combining $\{w^i_{t+1} | E^i \in S_{t+1}\}$
  6: Prune the set of experts $S_{t+1}$
  7: end for

- Online Gradient Descent (OGD)

$$w^i_{t+1} = \prod_{\mathcal{W}} \left( w^i_t - \eta_t \nabla f_t(w^i_t) \right), \forall E^i \in S_{t+1}$$

- Computational Cost per Iteration

$$|S_{t+1}| \text{ gradient evaluations of } f_t(\cdot)$$

where $|S_{t+1}| = O(\log t)$

---

[URL](http://cs.nju.edu.cn/zlj)
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
  \[ f_t(w) = \frac{1}{k} \sum_{i=1}^{k} \ell(w^\top x_i^t, y_i^t) \]

  Gradient evaluations are expensive when \( k \) is large

http://cs.nju.edu.cn/zlj

Online Learning
Online Learning with Surrogate Loss

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

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

- Experts
  \[ E^1 = \text{OGD}(\ell_1, \ell_2, \ell_3, \ldots, \ell_t, \ldots) \]
  \[ E^2 = \text{OGD}(\ell_2, \ell_3, \ldots, \ell_t, \ldots) \]
  \[ \ldots \]
  \[ E^t = \text{OGD}(\ell_t, \ldots) \]
Online Learning with Surrogate Loss

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

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

- Online Gradient Descent (OGD)

$$w^i_{t+1} = \Pi_{\mathcal{W}} \left( w^i_{t} - \eta_t \nabla \ell_t(w^i_{t}) \right), \forall E^i \in S_{t+1}$$
Online Learning with Surrogate Loss

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

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

Online Gradient Descent (OGD)

$$w^i_{t+1} = \Pi_{\mathcal{W}} \left( w^i_t - \eta_t \nabla \ell_t(w^i_t) \right), \forall E^i \in S_{t+1}$$

Computational Cost per Iteration

1 gradient evaluation of $f_t(\cdot)$
Convex Functions

- First-order Condition

\[ f_t(w_t) - f_t(w) \leq -\langle \nabla f_t(w_t), w - w_t \rangle \]
Convex Functions

- First-order Condition
  
  \[ f_t(w_t) - f_t(w) \leq -\langle \nabla f_t(w_t), w - w_t \rangle \]

- Surrogate Loss
  
  \[ \ell_t(w) = \langle \nabla f_t(w_t), w - w_t \rangle \]

  which is convex
Convex Functions

- **First-order Condition**
  
  \[ f_t(w_t) - f_t(w) \leq -\langle \nabla f_t(w_t), w - w_t \rangle \]

- **Surrogate Loss**
  
  \[ \ell_t(w) = \langle \nabla f_t(w_t), w - w_t \rangle \]
  
  which is convex

- **Properties**
  
  - Gradient evaluation is easy
    
    \[ \nabla \ell_t(w_t^i) = \nabla f_t(w_t) \]

  - The regret bound is maintained
    
    \[ f_t(w_t) - f_t(w) \leq \ell_t(w_t) - \ell_t(w) \]
Convex Functions

- First-order Condition
  \[ f_t(w_t) - f_t(w) \leq -\langle \nabla f_t(w_t), w - w_t \rangle \]

- Surrogate Loss
  \[ \ell_t(w) = \langle \nabla f_t(w_t), w - w_t \rangle \]
  which is convex

- Properties
  - Gradient evaluation is easy
    \[ \nabla \ell_t(w_i^t) = \nabla f_t(w_t) \]
  - The regret bound is maintained
    \[ f_t(w_t) - f_t(w) \leq \ell_t(w_t) - \ell_t(w) \]

- Adaptive Regret
  \[ R(T, \tau) = O \left( \sqrt{\tau \log T} \right) \]
Strongly Convex Functions

- First-order Condition

\[ f_t(w_t) - f_t(w) \leq -\langle \nabla f_t(w_t), w - w_t \rangle - \frac{\lambda}{2} \| w - w_t \|^2 \]
Strongly Convex Functions

- First-order Condition
  \[ f_t(w_t) - f_t(w) \leq -\langle \nabla f_t(w_t), w - w_t \rangle - \frac{\lambda}{2} \| w - w_t \|^2 \]

- Surrogate Loss
  \[ \ell_t(w) = \langle \nabla f_t(w_t), w - w_t \rangle + \frac{\lambda}{2} \| w - w_t \|^2 \]

which is strongly convex
Strongly Convex Functions

- First-order Condition
  \[ f_t(w_t) - f_t(w) \leq -\langle \nabla f_t(w_t), w - w_t \rangle - \frac{\lambda}{2} \| w - w_t \|^2 \]

- Surrogate Loss
  \[ \ell_t(w) = \langle \nabla f_t(w_t), w - w_t \rangle + \frac{\lambda}{2} \| w - w_t \|^2 \]
  which is strongly convex

- Properties
  - Gradient evaluation is easy
    \[ \nabla \ell_t(w_t^i) = \nabla f_t(w_t) + \lambda (w_t^i - w_t) \]
  - The regret bound is maintained
    \[ f_t(w_t) - f_t(w) \leq \ell_t(w_t) - \ell_t(w) \]
Strongly Convex Functions

- First-order Condition
  \[ f_t(w_t) - f_t(w) \leq -\langle \nabla f_t(w_t), w - w_t \rangle - \frac{\lambda}{2} \|w - w_t\|^2 \]

- Surrogate Loss
  \[ \ell_t(w) = \langle \nabla f_t(w_t), w - w_t \rangle + \frac{\lambda}{2} \|w - w_t\|^2 \]
  which is strongly convex

- Properties
  - Gradient evaluation is easy
    \[ \nabla \ell_t(w_t^i) = \nabla f_t(w_t) + \lambda (w_t^i - w_t) \]
  - The regret bound is maintained
    \[ f_t(w_t) - f_t(w) \leq \ell_t(w_t) - \ell_t(w) \]

- Adaptive Regret
  \[ R(T, \tau) = O(\log \tau \log T) \]
Exponentially Concave Functions

- First-order Condition [Hazan et al., 2007]
  \[ f_t(w_t) - f_t(w) \leq -\langle \nabla f_t(w_t), w - w_t \rangle - \frac{\beta}{2} \left( \langle \nabla f_t(w_t), w - w_t \rangle \right)^2 \]

- Surrogate Loss
  \[ \ell_t(w) = \langle \nabla f_t(w_t), w - w_t \rangle + \frac{\beta}{2} \left( \langle \nabla f_t(w_t), w - w_t \rangle \right)^2 \]
  which is exponentially concave

- Properties
  - Gradient evaluation is easy
    \[ \nabla \ell_t(w_t^i) = \nabla f_t(w_t) + \beta \langle \nabla f_t(w_t), w_t^i - w_t \rangle \nabla f_t(w_t) \]
  - The regret bound is maintained
    \[ f_t(w_t) - f_t(w) \leq \ell_t(w_t) - \ell_t(w) \]

- Adaptive Regret
  \[ R(T, \tau) = O(d \log \tau \log T) \]
Nuclear-norm Regularized Matrix Regression

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

\[ y_t = \text{trace}(W^*_\top X_t) + \epsilon_t, \text{ where } W_* \text{ changes twice} \]
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

\[ f_t(w) = \frac{1}{k} \sum_{i=1}^{k} \log \left( 1 + \exp \left( -y_t^i w^\top x_t^i \right) \right) \]

- the IJCNN01 dataset, where labels are flipped twice
Experiments

- Mini-batch Logistic Regression

\[ f_t(w) = \frac{1}{k} \sum_{i=1}^{k} \log \left( 1 + \exp \left( -y_t^i w^\top x_t^i \right) \right) \]

- the IJCNN01 dataset, where labels are flipped twice

---

http://cs.nju.edu.cn/zlj

Online Learning
Outline

1. Introduction
2. Adaptive Regret for Dynamic Environments
3. Efficient Algorithms for Adaptive Regret
4. Conclusion
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
Reference I

Thanks!

Strongly adaptive online learning.

Logarithmic regret algorithms for online convex optimization.

Adaptive algorithms for online decision problems.
Electronic Colloquium on Computational Complexity, 88.

Improved Strongly Adaptive Online Learning using Coin Betting.

The weighted majority algorithm.

The perceptron: a probabilistic model for information storage and organization in the brain.
Reference II


