Efficient Online Learning for Dynamic Environments

Lijun Zhang

Dept. of Computer Science and Technology, Nanjing University

2018 International Workshop on Signal Processing, Optimization and Control (SPOC 2018)
Outline

1. Introduction
   - Online Learning
   - Regret

2. Learning in Dynamic Environments
   - Adaptive Regret
   - Dynamic Regret

3. Our Work
   - Efficient Algorithms for Adaptive Regret
   - From Adaptive Regret to Dynamic Regret

4. Conclusion
Outline

1. Introduction
   - Online Learning
   - Regret

2. Learning in Dynamic Environments
   - Adaptive Regret
   - Dynamic Regret

3. Our Work
   - Efficient Algorithms for Adaptive Regret
   - From Adaptive Regret to Dynamic Regret

4. Conclusion
What Happens in an Internet Minute?

And Future Growth is Staggering


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

1. Introduction
   - Online Learning
     - Regret

2. Learning in Dynamic Environments
   - Adaptive Regret
   - Dynamic Regret

3. Our Work
   - Efficient Algorithms for Adaptive Regret
   - From Adaptive Regret to Dynamic Regret

4. Conclusion
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

\[ w_t \in \mathbb{R}^d \]

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

A classifier $w_t \in \mathbb{R}^d$

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.

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

Cumulative Loss

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

1. Introduction
   - Online Learning
   - Regret

2. Learning in Dynamic Environments
   - Adaptive Regret
   - Dynamic Regret

3. Our Work
   - Efficient Algorithms for Adaptive Regret
   - From Adaptive Regret to Dynamic Regret

4. Conclusion
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

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

#### Cumulative Loss

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

#### Regret

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

**Cumulative Loss**

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

**Regret**

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

Cumulative Loss of Online Learner - Minimal Loss of Offline Learner

**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

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)

\text{Cumulative Loss of Online Learner} \quad \text{Minimal Loss of 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}

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

$$\Pi_{\mathcal{W}}(x) = \arg\min_{w \in \mathcal{W}} ||w - x||_2$$
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
    \[
    f(w) = \log \left( 1 + \exp(-yx^\top w) \right)
    \]
  - 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**
   - Online Learning
   - Regret

2. **Learning in Dynamic Environments**
   - Adaptive Regret
   - Dynamic Regret

3. **Our Work**
   - Efficient Algorithms for Adaptive Regret
   - From Adaptive Regret to Dynamic Regret

4. **Conclusion**
The Challenge

Regret → Static Regret

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

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

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

One of the decision is reasonably good during \(T\) rounds
The Challenge

Regret → Static Regret

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

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

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

- 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
Outline

1. Introduction
   - Online Learning
   - Regret

2. Learning in Dynamic Environments
   - Adaptive Regret
   - Dynamic Regret

3. Our Work
   - Efficient Algorithms for Adaptive Regret
   - From Adaptive Regret to Dynamic Regret

4. Conclusion
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

- 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

- 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 $\tau$

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

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

Online Learning
Adaptive Regret

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

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

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

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

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

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

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

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

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

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

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

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

- Following the Leading History

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

7: end for
Adaptive Algorithms

Following the Leading History

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

7: **end for**

Experts

$$E^1 = \text{OGD}(f_1, f_2, f_3, \ldots, f_t, \ldots)$$

$$E^2 = \text{OGD}(f_2, f_3, \ldots, f_t, \ldots)$$

$$E^3 = \text{OGD}(f_3, \ldots, f_t, \ldots)$$

$$\vdots$$

$$E^t = \text{OGD}(f_t, \ldots)$$
Adaptive Algorithms

- Following the Leading History

1: \textbf{for} $t = 1, 2, \ldots, T$ \textbf{do}
2: \quad Submit $w_t \in V$ and observes $f_t(\cdot)$
3: \quad Initialize an expert $E^t$ by running OGD($f_t, \ldots$)
4: \quad Add $E^t$ to the set of experts $S_{t+1} = S_t \cup \{E^t\}$
5: \quad Get the prediction $w_{t+1}^i$ for each expert $E^i \in S_{t+1}$

7: \textbf{end for}

- Online Gradient Descent (OGD)

$$w_{t+1}^i = \prod_{W} \left( w_t^i - \eta_t \nabla f_t(w_t^i) \right), \ \forall E^i \in S_{t+1}$$
Adaptive Algorithms

- Following the Leading History
  1: for $t = 1, 2, \ldots, T$ do
  2: Submit $w_t \in W$ 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 $\{w^{i}_{t+1}|E^i \in S_{t+1}\}$

- end for

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

■ Following the Leading History

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

■ Experts

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

- **Exponentially Concave Functions** [Hazan and Seshadhri, 2007]
  \[ R(T, \tau) = O\left(d \log \tau \log T\right) \]
Introduction

Online Learning

Regret

Learning in Dynamic Environments

Adaptive Regret

Dynamic Regret

Our Work

Efficient Algorithms for Adaptive Regret

From Adaptive Regret to Dynamic Regret

Conclusion
Dynamic Regret

- **Static Regret**

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

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

- **General Dynamic Regret**

  \[
  R(u_1, \ldots, u_T) = \sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(u_t)
  \]

  where \( u_1, \ldots, u_T \in W \)
Dynamic Regret

- **Static Regret**

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

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

- **General Dynamic Regret**

\[
R(u_1, \ldots, u_T) = \sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(u_t)
\]

where \( u_1, \ldots, u_T \in \mathcal{W} \)

- **Worst-case Dynamic Regret**

\[
R(w_1^*, \ldots, w_T^*) = \sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(w_t^*)
\]

where \( w_t^* \in \arg\min_{w \in \mathcal{W}} f_t(w) \)
Worst-case Dynamic Regret

- The Challenge

Sublinear Dynamic Regret is **Impossible** in General!
Worst-case Dynamic Regret

The Challenge

Sublinear Dynamic Regret is Impossible in General!

Assumptions

- Functional Variation [Besbes et al., 2015]
  \[ V_T := \sum_{t=2}^{T} \| f_t(\cdot) - f_{t-1}(\cdot) \|_\infty = \sum_{t=2}^{T} \max_{w \in \mathcal{W}} | f_t(w) - f_{t-1}(w) | \]

- Path-length [Mokhtari et al., 2016]
  \[ P_T^* := \sum_{t=2}^{T} \| w_t^* - w_{t-1}^* \| \]

- Squared Path-length [Zhang et al., 2017]
  \[ S_T^* := \sum_{t=2}^{T} \| w_t^* - w_{t-1}^* \|^2 \]
A Representative Result

- **Functional Variation** [Besbes et al., 2015]
  - Restarted online gradient descent
  \[
  R(w_1^*, \ldots, w_T^*) \leq \begin{cases} 
    O\left(\frac{T^{2/3} V_T^{1/3}}{V_T}\right), & \text{Convex Functions} \\
    O\left(\log T \sqrt{TV_T}\right), & \text{Strongly Convex Functions}
  \end{cases}
  \]

- **Advantage**
  - Dynamic regret is sublinear if $V_T$ is sublinear

- **Limitations**
  - The algorithm needs to known an upper bound of $V_T$
  - It cannot be used in practice
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


Outline

1. Introduction
   - Online Learning
   - Regret

2. Learning in Dynamic Environments
   - Adaptive Regret
   - Dynamic Regret

3. Our Work
   - Efficient Algorithms for Adaptive Regret
   - From Adaptive Regret to Dynamic Regret

4. Conclusion
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)**

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

- Following the Leading History

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

- Online Gradient Descent (OGD)

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

- Computational Cost per Iteration

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

where $|S_{t+1}| = O(\log t)$
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
  \[ f_t(w) = \frac{1}{k} \sum_{i=1}^{k} \ell(w^\top x_t^i, y_t^i) \]

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

http://cs.nju.edu.cn/zlj
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 $w_{t+1}^i$ for each expert $E^i \in S_{t+1}$
6. Predict $w_{t+1}$ by combining $\{w_{t+1}^i | E^i \in S_{t+1}\}$
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]

\begin{algorithm}
\begin{algorithmic}[1]
\State \textbf{for} $t = 1, 2, \ldots, T$ \textbf{do}
\State Submit $w_t \in \mathcal{W}$ and observes $f_t(\cdot)$
\State Construct a surrogate loss $\ell_t(\cdot)$ from $\nabla f_t(w_t)$
\State Initialize an expert $E^t$ by running $\text{OGD}(\ell_t, \ldots)$
\State Add $E^t$ to the set of experts $S_{t+1} = S_t \cup \{E^t\}$
\State Get the prediction $w_{t+1}^i$ for each expert $E^i \in S_{t+1}$
\State Predict $w_{t+1}$ by combining $\{w_{t+1}^i | E^i \in S_{t+1}\}$
\State Prune the set of experts $S_{t+1}$
\State \textbf{end for}
\end{algorithmic}
\end{algorithm}

- Experts

\[
E^1 = \text{OGD}(\ell_1, \ell_2, \ell_3, \ldots, \ell_t, \ldots) \\
E^2 = \text{OGD}(\ell_2, \ell_3, \ldots, \ell_t, \ldots) \\
\vdots \\
E^t = \text{OGD}(\ell_t, \ldots) 
\]
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)

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

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

1: for \( t = 1, 2, \ldots, T \) do
2: \( w_t \in \mathcal{W} \) 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)

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

Computational Cost per Iteration

1 gradient evaluation of \( f_t(\cdot) \)
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
    \[ \nabla \ell_t(w_t^i) = \nabla f_t(w_t) \]
  - The regret bound is maintained
    \[ f_t(w_t) - f_t(w) \leq \ell_t(w_t) - \ell_t(w) \]

- **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
    \[ f_t(w_t) - f_t(w) \leq \ell_t(w_t) - \ell_t(w) \]
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
  \[ \nabla \ell_t(w^i_t) = \nabla f_t(w_t) + \lambda(w^i_t - w_t) \]

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

Adaptive Regret

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

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

\[ f_t(w_t) - f_t(w) \leq -\langle \nabla f_t(w_t), w - w_t \rangle - \frac{\beta}{2} (\langle \nabla f_t(w_t), w - w_t \rangle)^2 \]
Exponentially Concave Functions

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

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

\[ f_t(w_t) - f_t(w) \leq \ell_t(w_t) - \ell_t(w) \]
Exponentially Concave Functions

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

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

- Properties
  
  - Gradient evaluation is easy
    
    $$\nabla \ell_t(w_t) = \nabla f_t(w_t) + \beta \langle \nabla f_t(w_t), w_t - w_t \rangle \nabla f_t(w_t)$$
  
  - 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

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

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

- 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
Adaptive Regret versus Dynamic Regret

■ Adaptive Regret

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

- Well-studied

■ Worst-case Dynamic Regret

\[ R(w_1^*, \ldots, w_T^*) = \sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(w_t^*) = \sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} \min_{w \in \mathcal{W}} f_t(w) \]

where \( w_t^* \in \arg\min_{w \in \mathcal{W}} f_t(w) \)

- Partially addressed

[Hall and Willett, 2013, Jadbabaie et al., 2015, Mokhtari et al., 2016, Yang et al., 2016, Zhang et al., 2017]
From Adaptive Regret to Dynamic Regret

Theorem [Zhang et al., 2018]

Let $\mathcal{I}_1 = [s_1, q_1], \mathcal{I}_2 = [s_2, q_2], \ldots, \mathcal{I}_k = [s_k, q_k]$ be a partition of $[1, T]$.

Define the local functional variation of the $i$-th interval as

$$V_T(i) = \sum_{t=s_i+1}^{q_i} \max_{\mathbf{w} \in \mathcal{W}} |f_t(\mathbf{w}) - f_{t-1}(\mathbf{w})|$$

We have

$$R(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*) \leq \min_{\mathcal{I}_1, \ldots, \mathcal{I}_k} \sum_{i=1}^{k} (R(T, |\mathcal{I}_i|) + 2|\mathcal{I}_i| \cdot V_T(i))$$
From Adaptive Regret to Dynamic Regret

Theorem [Zhang et al., 2018]

- Let $\mathcal{I}_1 = [s_1, q_1], \mathcal{I}_2 = [s_2, q_2], \ldots, \mathcal{I}_k = [s_k, q_k]$ be a partition of $[1, T]$.
- Define the local functional variation of the $i$-th interval as

$$V_T(i) = \sum_{t=s_i+1}^{q_i} \max_{w \in \mathcal{W}} |f_t(w) - f_{t-1}(w)|$$

We have

$$R(w_1^*, \ldots, w_T^*) \leq \min_{\mathcal{I}_1, \ldots, \mathcal{I}_k} \sum_{i=1}^{k} \left( R(T, |\mathcal{I}_i|) + 2|\mathcal{I}_i| \cdot V_T(i) \right)$$

Corollary

$$R(w_1^*, \ldots, w_T^*) \leq \min_{1 \leq \tau \leq T} \left( \frac{R(T, \tau) T}{\tau} + 2\tau V_T \right)$$

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

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

\[ \Rightarrow R(w_1^*, \ldots, w_T^*) = O \left( \max \left\{ \sqrt{T \log T}, \frac{T^{2/3} V_T^{1/3} \log^{1/3} T} \right\} \right) \]

Lower bound is \( O(T^{2/3} V_T^{1/3}) \) [Besbes et al., 2015]
Dynamic Regret of Adaptive Algorithms

- Convex Functions
  \[ R(T, \tau) = O\left(\sqrt{\tau \log T}\right) \]
  \[ \Rightarrow R(w_1^*, \ldots, w_T^*) = O\left(\max \left\{ \sqrt{T \log T}, T^{2/3} V_T^{1/3} \log^{1/3} T \right\} \right) \]
- Lower bound is \( O(T^{2/3} V_T^{1/3}) \) [Besbes et al., 2015]

- Strongly Convex Functions
  \[ R(T, \tau) = O\left(\log \tau \log T\right) \]
  \[ \Rightarrow R(w_1^*, \ldots, w_T^*) = O\left(\max \left\{ \log T, \sqrt{TV_T} \log T \right\} \right) \]
- Lower bound is \( O(\sqrt{TV_T}) \) [Besbes et al., 2015]
Dynamic Regret of Adaptive Algorithms

- **Convex Functions**
  \[ R(T, \tau) = O\left(\sqrt{\tau \log T}\right) \]
  \[ \Rightarrow R(w_1^*, \ldots, w_T^*) = O\left(\max\left\{ \sqrt{T \log T}, T^{2/3} V_T^{1/3} \log^{1/3} T \right\}\right) \]
  - Lower bound is \( O(T^{2/3} V_T^{1/3}) \) [Besbes et al., 2015]

- **Strongly Convex Functions**
  \[ R(T, \tau) = O\left(\log \tau \log T\right) \]
  \[ \Rightarrow R(w_1^*, \ldots, w_T^*) = O\left(\max\left\{ \log T, \sqrt{TV_T \log T}\right\}\right) \]
  - Lower bound is \( O(\sqrt{TV_T}) \) [Besbes et al., 2015]

- **Exponentially Concave Functions**
  \[ R(T, \tau) = O\left(d \log \tau \log T\right) \]
  \[ \Rightarrow R(w_1^*, \ldots, w_T^*) = O\left(d \cdot \max\left\{ \log T, \sqrt{TV_T \log T}\right\}\right) \]
Our Contributions

1. Adaptive algorithms can be directly leveraged to minimize the dynamic regret.

2. The dynamic regrets are established without prior knowledge of the functional variation.
   - The results of [Besbes et al., 2015] needs to known an upper bound of $V_T$.

3. This is the first time that exponential concavity is utilized in dynamic regret.
Outline

1 Introduction
   • Online Learning
   • Regret

2 Learning in Dynamic Environments
   • Adaptive Regret
   • Dynamic Regret

3 Our Work
   • Efficient Algorithms for Adaptive Regret
   • From Adaptive Regret to Dynamic Regret

4 Conclusion
Conclusion

- A brief introduction to adaptive regret and dynamic regret
- Efficient algorithms for adaptive regret
- From adaptive regret to dynamic regret

Future Work

- Adaptive regret that does not depend on $T$
- General dynamic regret

$$R(u_1, \ldots, u_T) = \sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(u_t)$$

for any sequence $u_1, \ldots, u_T \in \mathcal{W}$

- Without convexity
Introduction Dynamic Environments Our Work Conclusion

Reference I

Thanks!

Non-stationary stochastic optimization.

Strongly adaptive online learning.

Dynamical models and tracking regret in online convex programming.

Logarithmic regret algorithms for online convex optimization.

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

Online optimization: Competing with dynamic comparators.

Improved Strongly Adaptive Online Learning using Coin Betting.
In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 943–951.
The weighted majority algorithm.

Online optimization in dynamic environments: Improved regret rates for strongly convex problems.

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

Shalev-Shwartz, S. (2011).
Online learning and online convex optimization.

Pegasos: primal estimated sub-gradient solver for SVM.

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

Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient.
Dynamic regret of strongly adaptive methods.
In *Proceedings of the 35th International Conference on Machine Learning*.

Improved dynamic regret for non-degenerate functions.

Online convex programming and generalized infinitesimal gradient ascent.