# Smoothed Online Convex Optimization

Lijun Zhang

Nanjing University, China

May 11, 2022

# Outline



LAMDA Learning And Mining from DatA

# Outline

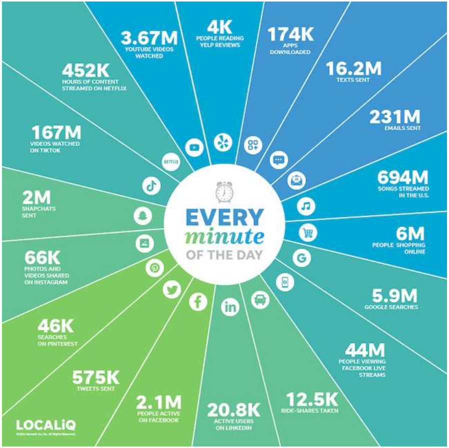
# Background

- Empirical Risk Minimization (ERM)

$$\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(\mathbf{x}_i), y_i))$$

  - $\mathcal{H}$ is a hypothesis space
  - $\ell(\cdot, \cdot)$ is a loss function
  - $(\mathbf{x}_i, y_i)$'s are training samples

- Batch Learning




https://localiq.com/blog/what-happens-in-an-internet-minute/

## 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.

## The Learning Procedure

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

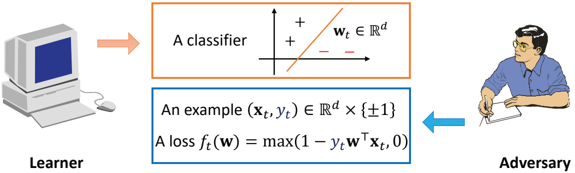
4: **end for**

## 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.

## The Learning Procedure

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**

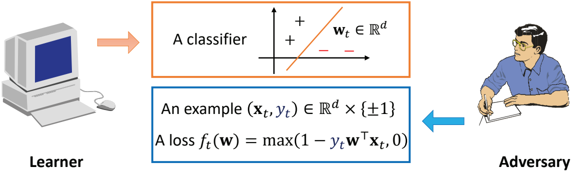


## Online Learning [Shalev-Shwartz, 2011]

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

## The Learning Procedure

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

## The Learning Procedure

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

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

# Outline

# 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

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

LAMDA Learning And Mining from DatA

# Regret

## Cumulative Loss

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

## Regret

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

# Regret

## Cumulative Loss

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

## Regret

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

## Hannan Consistent

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

# Regret

## Cumulative Loss

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

## Regret

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

## Hannan Consistent

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

# Online Convex Optimization (OCO) [Zinkevich, 2003]

- The Learning Procedure

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**

where $\mathcal{W}$ and $f_t$'s are convex

# Online Convex Optimization (OCO) [Zinkevich, 2003]

- The Learning Procedure

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

where $\mathcal{W}$ and $f_t$'s are convex

- Online Gradient Descent (OGD)

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

where

$$\Pi_{\mathcal{W}}[\mathbf{x}] = \operatorname*{argmin}_{\mathbf{w} \in \mathcal{W}} \|\mathbf{w} - \mathbf{x}\|$$

is the projection operator

# Existing Results for OCO

- Convex Functions [Zinkevich, 2003]
  - Online Gradient Descent (OGD)

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

- Strongly Convex Functions [Hazan et al., 2007]
  - Online Gradient Descent (OGD)

$$\text{Regret}(T) = O(\log T)$$

- Exponentially Concave Functions [Hazan et al., 2007]
  - Online Newton Step (ONS)

$$\text{Regret}(T) = O(d \log T)$$

# Outline

# Learning in Changing Environments

## Regret → *Static* Regret

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

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

- One of the decision is reasonably good during $T$ rounds

Learning And Mining from DatA

# Learning in Changing Environments

## Regret → *Static Regret*

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

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

## Changing 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

## Dynamic Regret

$$\text{D-Regret}(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)$$

where $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{W}$ is an arbitrary comparator sequence

## Dynamic Regret

$$\text{D-Regret}(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)$$

where $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{W}$ is an arbitrary comparator sequence

- Online Gradient Descent (OGD) [Zinkevich, 2003]

$$\text{D-Regret}(\mathbf{u}_1, \ldots, \mathbf{u}_T) = O\left(\sqrt{T} \cdot (1 + P_T)\right)$$

where $P_T = \sum_{t=1}^{T} \|\mathbf{u}_{t+1} - \mathbf{u}_t\|$

## Dynamic Regret

$$\text{D-Regret}(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)$$

where $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{W}$ is an arbitrary comparator sequence

- Online Gradient Descent (OGD) [Zinkevich, 2003]

$$\text{D-Regret}(\mathbf{u}_1, \ldots, \mathbf{u}_T) = O\left(\sqrt{T} \cdot (1 + P_T)\right)$$

where $P_T = \sum_{t=1}^{T} \|\mathbf{u}_{t+1} - \mathbf{u}_t\|$

- The First Lower Bound [Zhang et al., 2018a]

$$\text{D-Regret}(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \Omega\left(\sqrt{T} \cdot \sqrt{1 + P_T}\right)$$

- An Optimal Algorithm—Ader [Zhang et al., 2018a]

$$\text{D-Regret}(\mathbf{u}_1, \ldots, \mathbf{u}_T) = O\left(\sqrt{T} \cdot \sqrt{1 + P_T}\right)$$

# Outline



LAMDA Learning And Mining from DatA

# Adaptive Regret

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

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

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

# Adaptive Regret

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

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

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

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

# Adaptive Regret

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

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

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

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

# Adaptive Regret

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

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

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

$$\overbrace{\sum_{t=2}^{\tau+1} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=2}^{\tau+1} f_t(\mathbf{w})}$$

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

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

LAMDA Learning And Mining from DatA

# Adaptive Regret

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

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

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

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

# Adaptive Regret

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

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

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

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

# Existing Results on Adaptive Regret

- Convex Functions [Jun et al., 2017]

$$\text{SA-Regret}(T,\tau) = O\left(\sqrt{\tau \log T}\right)$$

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

$$\text{SA-Regret}(T,\tau) = O\left(\log \tau \log T\right)$$

- Exponentially Concave Functions [Hazan and Seshadhri, 2007]

$$\text{SA-Regret}(T,\tau) = O\left(d \log \tau \log T\right)$$

LAMDA Learning And Mining from DatA

# Existing Results on Adaptive Regret

- Convex Functions [Jun et al., 2017]

$$\text{SA-Regret}(T,\tau) = O\left(\sqrt{\tau \log T}\right)$$

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

$$\text{SA-Regret}(T,\tau) = O\left(\log \tau \log T\right)$$

- Exponentially Concave Functions [Hazan and Seshadhri, 2007]

$$\text{SA-Regret}(T,\tau) = O\left(d \log \tau \log T\right)$$

- A Universal Algorithm–UMA [Zhang et al., 2021b]

$$\text{SA-Regret}(T,\tau) = \begin{cases} O\left(\sqrt{\tau \log T}\right), \text{Convex} \\ O\left(\log \tau \log T\right), \text{Strongly Convex} \\ O\left(d \log \tau \log T\right), \text{Exponentially Concave} \end{cases}$$

LAMDA Learning And Mining from DatA

# Outline

# Smoothed Online Learning

## The Learning Procedure

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 a hitting cost $f_t(\mathbf{w}_t)$,

and a switching cost $m(\mathbf{w}_t, \mathbf{w}_{t-1})$

4: **end for**

- For example, $m(\mathbf{w}_t, \mathbf{w}_{t-1}) = \|\mathbf{w}_t - \mathbf{w}_{t-1}\|$ or $\frac{1}{2}\|\mathbf{w}_t - \mathbf{w}_{t-1}\|^2$

# Smoothed Online Learning

## The Learning Procedure

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 a hitting cost $f_t(\mathbf{w}_t)$,
and a switching cost $m(\mathbf{w}_t, \mathbf{w}_{t-1})$
4: **end for**

- For example, $m(\mathbf{w}_t, \mathbf{w}_{t-1}) = \|\mathbf{w}_t - \mathbf{w}_{t-1}\|$ or $\frac{1}{2}\|\mathbf{w}_t - \mathbf{w}_{t-1}\|^2$

- Applications
  - Stock market: the transaction fee
  - Data center: the wear-and-tear cost
  - Store relocation: the decoration cost

# Smoothed Online Learning

## The Learning Procedure

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 a hitting cost $f_t(\mathbf{w}_t)$,
and a switching cost $m(\mathbf{w}_t, \mathbf{w}_{t-1})$
4: **end for**

- For example, $m(\mathbf{w}_t, \mathbf{w}_{t-1}) = \|\mathbf{w}_t - \mathbf{w}_{t-1}\|$ or $\frac{1}{2}\|\mathbf{w}_t - \mathbf{w}_{t-1}\|^2$

## Cumulative Loss (Hitting Cost + Switching Cost)

$$\text{Cumulative Loss} = \sum_{t=1}^{T} f_t(\mathbf{w}_t) + m(\mathbf{w}_t, \mathbf{w}_{t-1})$$

# Smoothed Online Learning

## The Learning Procedure

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 a hitting cost $f_t(\mathbf{w}_t)$,
and a switching cost $m(\mathbf{w}_t, \mathbf{w}_{t-1})$
4: **end for**

- For example, $m(\mathbf{w}_t, \mathbf{w}_{t-1}) = \|\mathbf{w}_t - \mathbf{w}_{t-1}\|$ or $\frac{1}{2}\|\mathbf{w}_t - \mathbf{w}_{t-1}\|^2$

- Smoothed Online Convex Optimization
  - $f_t$'s are convex functions
  - $\mathcal{W}$ is a convex set

LAMDA Learning And Mining from DatA

# Smoothed Online Learning

## The Learning Procedure

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

2: Adversary chooses a function $f_t(\cdot)$,

then Learner picks a decision $\mathbf{w}_t \in \mathcal{W}$

3: Learner suffers a hitting cost $f_t(\mathbf{w}_t)$,

and a switching cost $m(\mathbf{w}_t, \mathbf{w}_{t-1})$

4: **end for**

- For example, $m(\mathbf{w}_t, \mathbf{w}_{t-1}) = \|\mathbf{w}_t - \mathbf{w}_{t-1}\|$ or $\frac{1}{2}\|\mathbf{w}_t - \mathbf{w}_{t-1}\|^2$

## The Lookahead Setting

The problem is *nontrivial* even when the learner can observe $f_t(\cdot)$ before deciding $\mathbf{w}_t$.

# Outline

## Dynamic Regret with Switching Cost

$$\sum_{t=1}^{T}\left(f_t(\mathbf{w}_t) + m(\mathbf{w}_t, \mathbf{w}_{t-1})\right) - \sum_{t=1}^{T}\left(f_t(\mathbf{u}_t) + m(\mathbf{u}_t, \mathbf{u}_{t-1})\right)$$

where $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{W}$ is an arbitrary comparator sequence

- The standard setting
- The lookahead setting

## Dynamic Regret with Switching Cost

$$\sum_{t=1}^{T} \left(f_t(\mathbf{w}_t) + m(\mathbf{w}_t, \mathbf{w}_{t-1})\right) - \sum_{t=1}^{T} \left(f_t(\mathbf{u}_t) + m(\mathbf{u}_t, \mathbf{u}_{t-1})\right)$$

where $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{W}$ is an arbitrary comparator sequence

- The standard setting
- The lookahead setting

## Dynamic Regret

$$\text{D-Regret}(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)$$

- An Optimal Algorithm—Ader [Zhang et al., 2018a]

$$\text{D-Regret}(\mathbf{u}_1, \ldots, \mathbf{u}_T) = O\left(\sqrt{T} \cdot \sqrt{1 + P_T}\right)$$

# The Standard Setting

## Assumptions

1. All the functions $f_t$'s are convex over their domain $\mathcal{W}$
2. The gradients of all functions are bounded by $G$
3. The diameter of the domain $\mathcal{W}$ is bounded by $D$

# The Standard Setting

## Assumptions

1. All the functions $f_t$'s are convex over their domain $\mathcal{W}$
2. The gradients of all functions are bounded by $G$
3. The diameter of the domain $\mathcal{W}$ is bounded by $D$

- Smoothed Ader (SAder) [Zhang et al., 2021a]

$$\sum_{t=1}^{T} \Big( f_t(\mathbf{w}_t) + \|\mathbf{w}_t - \mathbf{w}_{t-1}\| \Big) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) = O\left(\sqrt{T} \cdot \sqrt{1 + P_T}\right)$$

where $P_T = \sum_{t=1}^{T} \|\mathbf{u}_{t+1} - \mathbf{u}_t\|$

- Optimal according to the lower bound of dynamic regret [Zhang et al., 2018a]
- The switching cost does not make the problem much harder, although we need to modify the algorithm

# The Standard Setting

## Assumptions

1. All the functions $f_t$'s are convex over their domain $\mathcal{W}$
2. The gradients of all functions are bounded by $G$
3. The diameter of the domain $\mathcal{W}$ is bounded by $D$

- Online Gradient Descent (OGD)

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

  - Dynamic regret with switching cost

$$\sum_{t=1}^{T}\Big(f_t(\mathbf{w}_t) + \|\mathbf{w}_t - \mathbf{w}_{t-1}\|\Big) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) = O\left(\frac{1+P_T}{\eta} + \eta T\right)$$

  - We obtain an $O(\sqrt{T}\cdot\sqrt{1+P_T})$ bound if $\eta = \sqrt{(1+P_T)/T}$

# The Standard Setting

## Assumptions

1. All the functions $f_t$'s are convex over their domain $\mathcal{W}$
2. The gradients of all functions are bounded by $G$
3. The diameter of the domain $\mathcal{W}$ is bounded by $D$

- Online Gradient Descent (OGD)

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

- Dynamic regret with switching cost

$$\sum_{t=1}^{T}\left(f_t(\mathbf{w}_t) + \|\mathbf{w}_t - \mathbf{w}_{t-1}\|\right) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) = O\left(\frac{1+P_T}{\eta} + \eta T\right)$$

- We obtain an $O(\sqrt{T} \cdot \sqrt{1+P_T})$ bound if $\eta = \sqrt{(1+P_T)/T}$

But the path-length $P_T$ is unknown.

# Smoothed Ader (SAder) [Zhang et al., 2021a]

## The Basic Idea

- Discretize the possible values of $P_T \in [0, TD]$
- Create one expert for each discrete $P_T$, and combine them

# Smoothed Ader (SAder) [Zhang et al., 2021a]

## The Basic Idea

- Discretize the possible values of $P_T \in [0, TD]$
- Create one expert for each discrete $P_T$, and combine them

## A Set of Experts

- Online Gradient Descent (OGD) with $\eta = 1$
- $\cdots$
- Online Gradient Descent (OGD) with $\eta = 1/\sqrt{T}$

$$\mathbf{w}_{t+1}^{\eta} = \Pi_{\mathcal{W}}\left[\mathbf{w}_t^{\eta} - \eta \nabla f_t(\mathbf{w}_t^{\eta})\right], \ \eta \in \mathcal{H}$$

# Smoothed Ader (SAder) [Zhang et al., 2021a]

## The Basic Idea

- Discretize the possible values of $P_T \in [0, TD]$
- Create one expert for each discrete $P_T$, and combine them

### A Set of Experts

- Online Gradient Descent (OGD) with $\eta = 1$
- $\cdots$
- Online Gradient Descent (OGD) with $\eta = 1/\sqrt{T}$

$$\mathbf{w}_{t+1}^{\eta} = \Pi_{\mathcal{W}}\left[\mathbf{w}_t^{\eta} - \eta \nabla f_t(\mathbf{w}_t^{\eta})\right], \ \eta \in \mathcal{H}$$

Aggregation

### A Meta-algorithm

- The goal: aggregate the predictions from experts
- The challenge: ensure a small switching cost

LAMDA Learning And Mining from DatA

# Smoothed Ader (SAder) [Zhang et al., 2021a]

## The Basic Idea

- Discretize the possible values of $P_T \in [0, TD]$
- Create one expert for each discrete $P_T$, and combine them

■ A Set of Experts

- Online Gradient Descent (OGD) with $\eta = 1$
- $\ldots$
- Online Gradient Descent (OGD) with $\eta = 1/\sqrt{T}$

$$\mathbf{w}_{t+1}^{\eta} = \Pi_{\mathcal{W}}\left[\mathbf{w}_t^{\eta} - \eta \nabla f_t(\mathbf{w}_t^{\eta})\right], \ \eta \in \mathcal{H}$$

Aggregation

■ A Meta-algorithm (Hedge with switching cost)

$$\mathbf{w}_t = \sum_{\eta \in \mathcal{H}} \omega_t^{\eta} \mathbf{w}_t^{\eta}, \quad \omega_{t+1}^{\eta} = \frac{\omega_t^{\eta} e^{-\alpha \ell_t(\mathbf{w}_t^{\eta})}}{\sum_{\mu \in \mathcal{H}} \omega_t^{\mu} e^{-\alpha \ell_t(\mathbf{w}_t^{\mu})}}$$

$$\ell_t(\mathbf{w}_t^{\eta}) = \langle \nabla f_t(\mathbf{w}_t), \mathbf{w}_t^{\eta} - \mathbf{w}_t \rangle + \|\mathbf{w}_t^{\eta} - \mathbf{w}_{t-1}^{\eta}\|$$

# The Lookahead Setting

## Assumptions

1. All the functions $f_t$'s are convex over their domain $\mathcal{W}$
2. ~~The gradients of all functions are bounded by $G$~~
3. The diameter of the domain $\mathcal{W}$ is bounded by $D$

# The Lookahead Setting

## Assumptions

1. All the functions $f_t$'s are convex over their domain $\mathcal{W}$
2. ~~The gradients of all functions are bounded by $G$~~
3. The diameter of the domain $\mathcal{W}$ is bounded by $D$

- Lookahead SAder [Zhang et al., 2021a]

$$\sum_{t=1}^{T}\left(f_t(\mathbf{w}_t)+\|\mathbf{w}_t-\mathbf{w}_{t-1}\|\right)-\sum_{t=1}^{T}f_t(\mathbf{u}_t)=O\left(\sqrt{T}\cdot\sqrt{1+P_T}\right)$$

where $P_T=\sum_{t=1}^{T}\|\mathbf{u}_{t+1}-\mathbf{u}_t\|$

# The Lookahead Setting

## Assumptions

1. All the functions $f_t$'s are convex over their domain $\mathcal{W}$
2. ~~The gradients of all functions are bounded by $G$~~
3. The diameter of the domain $\mathcal{W}$ is bounded by $D$

- Lookahead SAder [Zhang et al., 2021a]

$$\sum_{t=1}^{T}\left(f_t(\mathbf{w}_t) + \|\mathbf{w}_t - \mathbf{w}_{t-1}\|\right) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) = O\left(\sqrt{T}\cdot\sqrt{1+P_T}\right)$$

where $P_T = \sum_{t=1}^{T}\|\mathbf{u}_{t+1} - \mathbf{u}_t\|$

- The first $\Omega(\sqrt{T}\cdot\sqrt{1+P_T})$ lower bound for lookahead setting [Zhang et al., 2021a]
  - Our lookahead SAder is optimal

# The Lookahead Setting

## Assumptions

1. All the functions $f_t$'s are convex over their domain $\mathcal{W}$
2. ~~The gradients of all functions are bounded by $G$~~
3. The diameter of the domain $\mathcal{W}$ is bounded by $D$

- Lookahead SAder [Zhang et al., 2021a]

$$\sum_{t=1}^{T}\left(f_t(\mathbf{w}_t) + \|\mathbf{w}_t - \mathbf{w}_{t-1}\|\right) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) = O\left(\sqrt{T}\cdot\sqrt{1+P_T}\right)$$

where $P_T = \sum_{t=1}^{T}\|\mathbf{u}_{t+1} - \mathbf{u}_t\|$

- The first $\Omega(\sqrt{T}\cdot\sqrt{1+P_T})$ lower bound for lookahead setting [Zhang et al., 2021a]
  - Our lookahead SAder is optimal

# Lookahead SAder [Zhang et al., 2021a]

## The Basic Idea

- Discretize the possible values of $P_T \in [0, TD]$
- Create one expert for each discrete $P_T$, and combine them

# Lookahead SAder [Zhang et al., 2021a]

## The Basic Idea

- Discretize the possible values of $P_T \in [0, TD]$
- Create one expert for each discrete $P_T$, and combine them

### A Set of Experts

- Balancing two costs directly with $\eta = 1$
- $\cdots$
- Balancing two costs directly $\eta = 1/\sqrt{T}$

$$\min_{\mathbf{x}\in\mathcal{X}} \quad f_t(\mathbf{x}) + \frac{1}{2\eta}\|\mathbf{x} - \mathbf{x}_{t-1}^{\eta}\|^2, \ \eta \in \mathcal{H}$$

# Lookahead SAder [Zhang et al., 2021a]

## The Basic Idea

- Discretize the possible values of $P_T \in [0, TD]$
- Create one expert for each discrete $P_T$, and combine them

### A Set of Experts

- Balancing two costs directly with $\eta = 1$
- $\cdots$
- Balancing two costs directly $\eta = 1/\sqrt{T}$

$$\min_{\mathbf{x}\in\mathcal{X}} \quad f_t(\mathbf{x}) + \frac{1}{2\eta}\|\mathbf{x} - \mathbf{x}_{t-1}^{\eta}\|^2, \ \eta \in \mathcal{H}$$

Aggregation

### A Meta-algorithm (Lookahead Hedge with switching cost)

$$\mathbf{w}_t = \sum_{\eta\in\mathcal{H}} \omega_t^{\eta}\mathbf{w}_t^{\eta}, \quad \omega_t^{\eta} = \frac{\omega_{t-1}^{\eta} e^{-\beta\ell_t(\mathbf{w}_t^{\eta})}}{\sum_{\mu\in\mathcal{H}} \omega_{t-1}^{\mu} e^{-\beta\ell_t(\mathbf{w}_t^{\mu})}}$$

$$\ell_t(\mathbf{w}_t^{\eta}) = \langle\nabla f_t(\mathbf{w}_t), \mathbf{w}_t^{\eta} - \mathbf{w}_t\rangle + \|\mathbf{w}_t^{\eta} - \mathbf{w}_{t-1}^{\eta}\|$$

# Outline

## Adaptive Regret with Switching Cost

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

- The standard setting

## Adaptive Regret with Switching Cost

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

- The standard setting

## Adaptive Regret

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

- Convex Functions [Jun et al., 2017]

$$\text{SA-Regret}(T,\tau) = O\left(\sqrt{\tau \log T}\right)$$

# The Standard Setting

## Assumptions

1. All the functions $f_t$'s are convex over their domain $\mathcal{W}$
2. The gradients of all functions are bounded by $G$
3. The diameter of the domain $\mathcal{W}$ is bounded by $D$

- Smoothed OGD [Zhang et al., 2022]

1. Adaptive regret with switching cost

$$\sum_{t=s}^{s+\tau-1} \left(f_t(\mathbf{w}_t) + \|\mathbf{w}_t - \mathbf{w}_{t+1}\| - f_t(\mathbf{w})\right) = O\left(\sqrt{\tau \log T}\right)$$

# The Standard Setting

## Assumptions

1. All the functions $f_t$'s are convex over their domain $\mathcal{W}$
2. The gradients of all functions are bounded by $G$
3. The diameter of the domain $\mathcal{W}$ is bounded by $D$

- Smoothed OGD [Zhang et al., 2022]

1. Adaptive regret with switching cost

$$\sum_{t=s}^{s+\tau-1} \left(f_t(\mathbf{w}_t)+\|\mathbf{w}_t-\mathbf{w}_{t+1}\|-f_t(\mathbf{w})\right) = O\left(\sqrt{\tau \log T}\right)$$

2. Dynamic regret with switching cost in every interval

$$\sum_{t=s}^{s+\tau-1} \left(f_t(\mathbf{w}_t)+\|\mathbf{w}_t-\mathbf{w}_{t+1}\|-f_t(\mathbf{u}_t)\right) = O\left(\sqrt{\tau(1+P_{r,s}) \log T}\right)$$

where $P_{r,s} = \sum_{t=r}^{s} \|\mathbf{u}_t - \mathbf{u}_{t+1}\|$

# Existing Framework for Adaptive Regret

- An Expert-algorithm
  - Online Gradient Descent (OGD) [Zinkevich, 2003]

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

# Existing Framework for Adaptive Regret

- An Expert-algorithm
  - Online Gradient Descent (OGD) [Zinkevich, 2003]

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

- A Set of Intervals
  - Geometric covering intervals [Daniely et al., 2015]

| $t$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | $\cdots$ |
|---|---|---|---|---|---|---|---|---|
| $\mathcal{I}_0$ | [ ] | [ ] | [ ] | [ ] | [ ] | [ ] | [ ] | $\cdots$ |
| $\mathcal{I}_1$ | | [ | ] | [ | ] | [ | ] | $\cdots$ |
| $\mathcal{I}_2$ | | | | [ | | | ] | $\cdots$ |

# Existing Framework for Adaptive Regret

- An Expert-algorithm
  - Online Gradient Descent (OGD) [Zinkevich, 2003]

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

- A Set of Intervals
  - Geometric covering intervals [Daniely et al., 2015]

| $t$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | $\cdots$ |
|---|---|---|---|---|---|---|---|---|
| $\mathcal{I}_0$ | [OGD($f_1$)] | [OGD($f_2$)] | [OGD($f_3$)] | [OGD($f_4$)] | [OGD($f_5$)] | [OGD($f_6$)] | [OGD($f_7$)] | $\cdots$ |
| $\mathcal{I}_1$ | | [OGD($f_2$, | $f_3$)] | [OGD($f_4$, | $f_5$)] | [OGD($f_6$, | $f_7$)] | $\cdots$ |
| $\mathcal{I}_2$ | | | | [OGD($f_4$, | $f_5$, | $f_6$, | $f_7$)] | $\cdots$ |

# Existing Framework for Adaptive Regret

- An Expert-algorithm
  - Online Gradient Descent (OGD) [Zinkevich, 2003]

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

- A Set of Intervals
  - Geometric covering intervals [Daniely et al., 2015]

| $t$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | $\cdots$ |
|---|---|---|---|---|---|---|---|---|
| $\mathcal{I}_0$ | [OGD($f_1$)] | [OGD($f_2$)] | [OGD($f_3$)] | [OGD($f_4$)] | [OGD($f_5$)] | [OGD($f_6$)] | [OGD($f_7$)] | $\cdots$ |
| $\mathcal{I}_1$ | | [OGD($f_2$, | $f_3$)] | [OGD($f_4$, | $f_5$)] | [OGD($f_6$, | $f_7$)] | $\cdots$ |
| $\mathcal{I}_2$ | | | | [OGD($f_4$, | $f_5$, | $f_6$, | $f_7$)] | $\cdots$ |

- A Meta-algorithm (which supports sleeping experts)
  - Sleeping Coin Betting [Jun et al., 2017]

# Existing Framework for Adaptive Regret

- An Expert-algorithm
  - Online Gradient Descent (OGD) [Zinkevich, 2003]

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

- A Set of Intervals
  - Geometric covering intervals [Daniely et al., 2015]

| $t$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | $\cdots$ |
|---|---|---|---|---|---|---|---|---|
| $\mathcal{I}_0$ | [OGD($f_1$)] | [OGD($f_2$)] | [OGD($f_3$)] | [OGD($f_4$)] | [OGD($f_5$)] | [OGD($f_6$)] | [OGD($f_7$)] | $\cdots$ |
| $\mathcal{I}_1$ | | [OGD($f_2$, | $f_3$)] | [OGD($f_4$, | $f_5$)] | [OGD($f_6$, | $f_7$)] | $\cdots$ |
| $\mathcal{I}_2$ | | | | [OGD($f_4$, | $f_5$, | $f_6$, | $f_7$)] | $\cdots$ |

- A Meta-algorithm (which supports sleeping experts)
  - Sleeping Coin Betting [Jun et al., 2017]

The dynamic change of experts makes it difficult to bound the switching cost of the meta-algorithm.

# Smoothed OGD [Zhang et al., 2022]

## The Basic Idea

- Create a set of OGD with different step sizes
- Combine them *sequentially* by Discounted-Normal-Predictor

# Smoothed OGD [Zhang et al., 2022]

## The Basic Idea

- Create a set of OGD with different step sizes
- Combine them *sequentially* by Discounted-Normal-Predictor

## A Set of Experts

- OGD with $\eta = 1/\sqrt{T}$
- OGD with $\eta = 2/\sqrt{T}$
- OGD with $\eta = 4/\sqrt{T}$
- $\cdots$
- OGD with $\eta = 1$

Learning And Mining from DatA

# Smoothed OGD [Zhang et al., 2022]

## The Basic Idea

- Create a set of OGD with different step sizes
- Combine them *sequentially* by Discounted-Normal-Predictor

## A Set of Experts

- OGD with $\eta = 1/\sqrt{T}$
- OGD with $\eta = 2/\sqrt{T}$
- OGD with $\eta = 4/\sqrt{T}$
- $\cdots$
- OGD with $\eta = 1$

} Aggregation

## A Meta-algorithm

- The goal: aggregate the predictions from experts
- The 1st challenge: support the adaptive regret
- The 2nd challenge: ensure a small switching cost

# Smoothed OGD [Zhang et al., 2022]

## The Basic Idea

- Create a set of OGD with different step sizes
- Combine them *sequentially* by Discounted-Normal-Predictor

- A Set of Experts
  - OGD with $\eta = 1/\sqrt{T}$
  - OGD with $\eta = 2/\sqrt{T}$
  - OGD with $\eta = 4/\sqrt{T}$
  - $\cdots$
  - OGD with $\eta = 1$

Sequential Aggregation

- A Meta-algorithm (Discounted-Normal-Predictor)
  - It *automatically* controls the switching cost [Kapralov and Panigrahy, 2010] [Daniely and Mansour, 2019]
  - We further utilize *conservative updating*

- Discounted-Normal-Predictor [Kapralov and Panigrahy, 2010]
  - Designed for the bit prediction problem
  - Receive a sequence of bits $b_1, \cdots, b_T \in [-1, 1]$
  - Output confidence levels $c_1, \ldots, c_T \in [-1, 1]$
  - Maximize the cumulative payoff $\sum_{t=1}^{T} c_t b_t$

## Discounted-Normal-Predictor [Kapralov and Panigrahy, 2010]

- Designed for the bit prediction problem
- Receive a sequence of bits $b_1, \cdots, b_T \in [-1, 1]$
- Output confidence levels $c_1, \ldots, c_T \in [-1, 1]$
- Maximize the cumulative payoff $\sum_{t=1}^{T} c_t b_t$

## The Learning Procedure

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

2: Predict $g(x_t)$ where

$$g(x) = \text{sign}(x) \cdot \min\left(Z \cdot \text{erf}\left(\frac{|x|}{4\sqrt{n}}\right) e^{\frac{x^2}{16n}}, 1\right)$$

3: Receive $b_t$

4: Set

$$x_{t+1} = \begin{cases} \rho x_t + b_t, & |x_t| < U(n) \text{ or } g(x_t) b_t < 0; \\ \rho x_t, & \text{otherwise} \end{cases}$$

5: **end for**

## Discounted-Normal-Predictor [Kapralov and Panigrahy, 2010]

- Aggregate two experts $E^1$ and $E^2$
- Define the bit as $b_t = \ell_t^1 - \ell_t^2 \in [-1, 1]$
- Output a confidence level $c_t \in [0, 1]$
- Predict the weighted average $c_t \mathbf{w}_t^1 + (1 - c_t)\mathbf{w}_t^2$

## The Learning Procedure

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

2: Predict $g(x_t)$ where

$$g(x) = \text{sign}(x) \cdot \min\left(Z \cdot \text{erf}\left(\frac{|x|}{4\sqrt{n}}\right) e^{\frac{x^2}{16n}}, 1\right)$$

3: Receive $b_t$

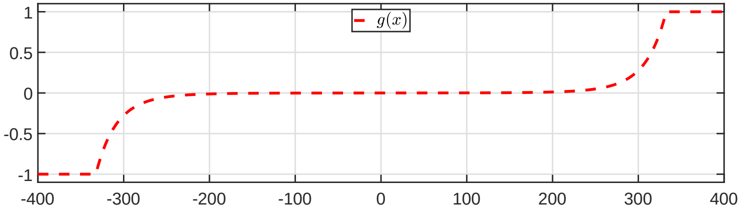
4: Set

$$x_{t+1} = \begin{cases} \rho x_t + b_t, & |x_t| < U(n) \text{ or } g(x_t) b_t < 0; \\ \rho x_t, & \text{otherwise} \end{cases}$$

5: **end for**

- Discounted-Normal-Predictor [Kapralov and Panigrahy, 2010]
  - Aggregate two experts $E^1$ and $E^2$
  - Define the bit as $b_t = \ell_t^1 - \ell_t^2 \in [-1, 1]$
  - Output a confidence level $c_t \in [0, 1]$
  - Predict the weighted average $c_t \mathbf{w}_t^1 + (1 - c_t)\mathbf{w}_t^2$
- The Underlying Rationale
  - Ordinary Differential Equation

$$g'(x) = \frac{x}{8L^2} g(x) + \frac{Z}{2\sqrt{n\pi}}$$



# Outline

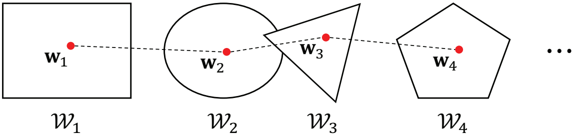
## Competitive Ratio

$$\frac{\sum_{t=1}^{T}\left(f_t(\mathbf{w}_t)+m(\mathbf{w}_t,\mathbf{w}_{t-1})\right)}{\min_{\mathbf{u}_0,\mathbf{u}_1,\ldots,\mathbf{u}_T\in\mathcal{X}}\sum_{t=1}^{T}\left(f_t(\mathbf{u}_t)+m(\mathbf{u}_t,\mathbf{u}_{t-1})\right)}$$

## Competitive Ratio

$$\frac{\sum_{t=1}^{T} \left(f_t(\mathbf{w}_t) + m(\mathbf{w}_t, \mathbf{w}_{t-1})\right)}{\min_{\mathbf{u}_0,\mathbf{u}_1,\ldots,\mathbf{u}_T \in \mathcal{X}} \sum_{t=1}^{T} \left(f_t(\mathbf{u}_t) + m(\mathbf{u}_t, \mathbf{u}_{t-1})\right)}$$

- Convex Body Chasing (CBC)
  - Select one point from convex bodies $\mathcal{W}_1, \ldots, \mathcal{W}_T \subseteq \mathbb{R}^d$

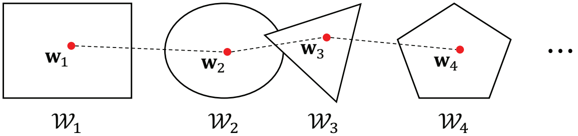



  - Minimize the total movement $\sum \|\mathbf{w}_t - \mathbf{w}_{t-1}\|$

## Competitive Ratio

$$\frac{\sum_{t=1}^{T} \left(f_t(\mathbf{w}_t) + m(\mathbf{w}_t, \mathbf{w}_{t-1})\right)}{\min_{\mathbf{u}_0,\mathbf{u}_1,\ldots,\mathbf{u}_T \in \mathcal{X}} \sum_{t=1}^{T} \left(f_t(\mathbf{u}_t) + m(\mathbf{u}_t, \mathbf{u}_{t-1})\right)}$$

- Convex Body Chasing (CBC)
  - Select one point from convex bodies $\mathcal{W}_1, \ldots, \mathcal{W}_T \subseteq \mathbb{R}^d$




  - Minimize the total movement $\sum \|\mathbf{w}_t - \mathbf{w}_{t-1}\|$
  - Lower bound: $\Omega(\sqrt{d})$ [Friedman and Linial, 1993]
  - Upper bound: $O(\min(d, \sqrt{d \log T}))$ [Argue et al., 2020, Sellke, 2020]

## Competitive Ratio

$$\frac{\sum_{t=1}^{T}\left(f_t(\mathbf{w}_t)+m(\mathbf{w}_t,\mathbf{w}_{t-1})\right)}{\min_{\mathbf{u}_0,\mathbf{u}_1,\ldots,\mathbf{u}_T\in\mathcal{X}}\sum_{t=1}^{T}\left(f_t(\mathbf{u}_t)+m(\mathbf{u}_t,\mathbf{u}_{t-1})\right)}$$

## Research on Competitive Ratio

Identify sufficient conditions and develop algorithms for

*dimension-free competitive ratio in lookahead setting*

- Polyhedral functions [Chen et al., 2018, Lin et al., 2020] [Zhang et al., 2021a]
- Quadratic growth functions [Goel et al., 2019, Lin et al., 2020] [Zhang et al., 2021a]
- Strongly convex functions [Goel et al., 2019]

The function can not be too flat.

# Applications

- Online Convex Optimization with Memory [Anava et al., 2015]

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

$$f_t(\mathbf{w}_{t-m}, \ldots, \mathbf{w}_t)$$

and updates $\mathbf{w}_t$
4: **end for**

# Applications

- Online Convex Optimization with Memory [Anava et al., 2015]

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

$$f_t(\mathbf{w}_{t-m}, \ldots, \mathbf{w}_t)$$

and updates $\mathbf{w}_t$
4: **end for**

- Online Non-stochastic Control [Agarwal et al., 2019]
  - The loss $c_t(\mathbf{x}_t, \mathbf{w}_t)$ depends on the current state $\mathbf{x}_t$ and the decision $\mathbf{w}_t$
  - Linear dynamical system

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{w}_t + \delta_t$$

where $\delta_t$ denotes the disturbance

# Outline

# Conclusion and Future Work

- Smoothed Online Learning
  - Minimize the sum of hitting cost and switching cost
  - Dynamic regret with switching cost, Adaptive regret with switching cost, Competitive ratio

# Conclusion and Future Work

## Smoothed Online Learning

- Minimize the sum of hitting cost and switching cost
- Dynamic regret with switching cost, Adaptive regret with switching cost, Competitive ratio

## Future Work

- Improve the rates under additional assumptions
- Control the switching cost directly [Wang et al., 2021]

$$\min \sum_{t=1}^{T} f_t(\mathbf{w}_t) \quad \text{s.t.} \sum_{t=1}^{T} \|\mathbf{w}_t - \mathbf{w}_{t-1}\| \leq B$$

- The relation with continual learning

$$\sum_{t=1}^{T} \underbrace{f_t(\mathbf{w}_t)}_{\text{Perform well on each task}} + \underbrace{\|\mathbf{w}_t - \mathbf{w}_{t-1}\|}_{\text{Avoid catastrophic forgetting}}$$

# Reference I

## Thanks!

Agarwal, N., Bullins, B., Hazan, E., Kakade, S., and Singh, K. (2019).
Online control with adversarial disturbances.
In *Proceedings of the 36th International Conference on Machine Learning*, pages 111–119.

Anava, O., Hazan, E., and Mannor, S. (2015).
Online learning for adversaries with memory: Price of past mistakes.
In *Advances in Neural Information Processing Systems 28*, volume 28, pages 784–792.

Argue, C., Gupta, A., Guruganesh, G., and Tang, Z. (2020).
Chasing convex bodies with linear competitive ratio.
In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1519–1524.

Chen, N., Goel, G., and Wierman, A. (2018).
Smoothed online convex optimization in high dimensions via online balanced descent.
In *Proceedings of the 31st Conference on Learning Theory*, pages 1574–1594.

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.

Daniely, A. and Mansour, Y. (2019).
Competitive ratio vs regret minimization: achieving the best of both worlds.
In *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, pages 333–368.

# Reference II


Friedman, J. and Linial, N. (1993).
On convex body chasing.
*Discrete & Computational Geometry*, 9:293–321.

Goel, G., Lin, Y., Sun, H., and Wierman, A. (2019).
Beyond online balanced descent: An optimal algorithm for smoothed online optimization.
In *Advances in Neural Information Processing Systems 32*, pages 1875–1885.

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.

Kapralov, M. and Panigrahy, R. (2010).
Prediction strategies without loss.
*ArXiv e-prints*, arXiv:1008.3672.

Lin, Y., Goel, G., and Wierman, A. (2020).
Online optimization with predictions and non-convex losses.
*Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(1):18:1–18:32.

# Reference III


Sellke, M. (2020).
Chasing convex bodies optimally.
In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1509–1518.

Shalev-Shwartz, S. (2011).
Online learning and online convex optimization.
*Foundations and Trends in Machine Learning*, 4(2):107–194.

Wang, G., Wan, Y., Yang, T., and Zhang, L. (2021).
Online convex optimization with continuous switching constraint.
In *Advances in Neural Information Processing Systems 34*, pages 28636–28647.

Zhang, L., Jiang, W., Lu, S., and Yang, T. (2021a).
Revisiting smoothed online learning.
In *Advances in Neural Information Processing Systems 34*, pages 13599–13612.

Zhang, L., Jiang, W., Yi, J., and Yang, T. (2022).
Smoothed online convex optimization based on discounted-normal-predictor.
*ArXiv e-prints*, arXiv:2205.00741.

Zhang, L., Lu, S., and Zhou, Z.-H. (2018a).
Adaptive online learning in dynamic environments.
In *Advances in Neural Information Processing Systems 31*, pages 1323–1333.

Zhang, L., Wang, G., Tu, W.-W., Jiang, W., and Zhou, Z.-H. (2021b).
Dual adaptivity: A universal algorithm for minimizing the adaptive regret of convex functions.
In *Advances in Neural Information Processing Systems 34*, pages 24968–24980.

# Reference IV

Zhang, L., Yang, T., Jin, R., and Zhou, Z.-H. (2018b).
Dynamic regret of strongly adaptive methods.
In *Proceedings of the 35th International Conference on Machine Learning*, pages 5882–5891.

Zinkevich, M. (2003).
Online convex programming and generalized infinitesimal gradient ascent.
In *Proceedings of the 20th International Conference on Machine Learning*, pages 928–936.