Online Learning and Online Convex Optimization

Introduction and Some New Trends

Behrad Moniri Mahdi Sabbaghi

Department of Electrical Engineering
Sharif University of Technology

Statistics Reading Group
Tehran Institute for Advanced Studies (TeIAS)

1. Introduction
2. Realizable Setting
3. Online Convex Optimization (OCO)
4. Follow The Leader
5. Follow the Regularized Leader
6. Online Mirror Descent
   - Regret Analysis
   - Normalized Exponentiated Gradient
   - L_p Algorithm
7. Bandits
   - Multi-Armed Bandits
   - Stochastic Bandits
8. New Trends
   - Bandits
   - Parameter-Free Online Learning
   - Combining Online Learning Guarantees
   - Predictable Sequences (a.k.a. Hints)
9. Bibliography

Section 1

Introduction

Introduction

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

Introduction

- Online learning is the process of answering a sequence of questions given (maybe partial) knowledge of the correct answers to previous questions and possibly additional available information.
- Many interesting theoretical properties and practical applications.

Setting

Online Learning

for $t = 1, 2, \dots$
receive question $\mathbf{x}_t \in \mathcal{X}$
predict $p_t \in D$
receive true answer $y_t \in \mathcal{Y}$
suffer loss $l(p_t, y_t)$

- Online Classification
- Online Regression
- Learning from Expert Advice
- **Online Convex Optimization**

Goals and Assumptions

- The learner's ultimate goal is to minimize the cumulative loss suffered along its run.

Goals and Assumptions

- The learner's ultimate goal is to minimize the cumulative loss suffered along its run.
- learning is hopeless if there is no relation between past and present rounds.

Goals and Assumptions

- The learner's ultimate goal is to minimize the cumulative loss suffered along its run.
- learning is hopeless if there is no relation between past and present rounds.
- i.i.d. in classical statistical learning theory vs. adversarial in online learning.

- Naturally, an adversary can make the cumulative loss to our online learning algorithm arbitrarily large.

- Naturally, an adversary can make the cumulative loss to our online learning algorithm arbitrarily large.
- It can ask the same question on each round, wait for the answer, and provide the opposite answer as the correct answer.

- Naturally, an adversary can make the cumulative loss to our online learning algorithm arbitrarily large.
- It can ask the same question on each round, wait for the answer, and provide the opposite answer as the correct answer.
- To make non-trivial statements, we make several natural assumptions.

Two scenarios

There are two main scenarios:

- **The Realizable Setting** (the simple one):
  Answers are generated by some mapping $h^* : \mathcal{X} \to \mathcal{Y}$ and $h^* \in \mathcal{H}$.
  - $\mathcal{H}$ is known by the learner.
  - $h^* \in \mathcal{H}$ is chosen by the adversary.

Two scenarios

There are two main scenarios:

- **The Realizable Setting** (the simple one):
  Answers are generated by some mapping $h^* : \mathcal{X} \to \mathcal{Y}$ and $h^* \in \mathcal{H}$.
  - $\mathcal{H}$ is known by the learner.
  - $h^* \in \mathcal{H}$ is chosen by the adversary.
- **Regret Setting** (the more interesting one): No longer assume answers are generated by $h^* \in \mathcal{H}$, but require the learner to be competitive with the best fixed predictor from $\mathcal{H}$:

$$\mathrm{Regret}_T(h^*) = \sum_{t=1}^{T} l(p_t, y_t) - \sum_{t=1}^{T} l(h^*(x_t), y_t) \tag{1}$$

$$\mathrm{Regret}_T(\mathcal{H}) = \max_{h^* \in \mathcal{H}} \mathrm{Regret}_T(h^*) \tag{2}$$

Low regret algorithm: $o(T)$ Regret.

Section 2

Realizable Setting

Realizable Setting

- Analogous to the PAC-Learning Setting
- With this restriction on the sequence, the learner should make as few mistakes as possible, i.e:

$$l_t(p_t, y_t) = \mathbf{1}\{p_t \neq y_t\} \tag{3}$$

Realizable Setting

- Analogous to the PAC-Learning Setting
- With this restriction on the sequence, the learner should make as few mistakes as possible, i.e:

$$l_t(p_t, y_t) = \mathbf{1}\{p_t \neq y_t\} \tag{3}$$

- Suppose we have given a sequence:

$$S = (x_1, h^*(y_1)), \ldots, (x_T, h^*(y_T))$$

Objective: minimize $\mathcal{M}_{\mathcal{A}}(\mathcal{H}) := \sup_{S \in \mathcal{S}} \sum_{t=1}^{N} \mathbf{1}\{p_{A_t} \neq y_t\}$

Realizable Setting

- Analogous to the PAC-Learning Setting
- With this restriction on the sequence, the learner should make as few mistakes as possible, i.e:

$$l_t(p_t, y_t) = \mathbf{1}\{p_t \neq y_t\} \tag{3}$$

- Suppose we have given a sequence:

$$S = (x_1, h^*(y_1)), \ldots, (x_T, h^*(y_T))$$

Objective: minimize $\mathcal{M}_{\mathcal{A}}(\mathcal{H}) := \sup_{S \in \mathcal{S}} \sum_{t=1}^{N} \mathbf{1}\{p_{A_t} \neq y_t\}$

- A bound on $\mathcal{M}_{\mathcal{A}}(\mathcal{H})$ is called a **mistake-bound**

Learnability

Definition

*We say that a hypothesis class $\mathcal{H}$ is **online learnable** if there exists an algorithm $\mathcal{A}$ for which $\mathcal{M}_{\mathcal{A}}(\mathcal{H}) < B < \infty$.*

Learnability

Definition

We say that a hypothesis class $\mathcal{H}$ is ***online learnable*** *if there exists an algorithm $\mathcal{A}$ for which $\mathcal{M}_{\mathcal{A}}(\mathcal{H}) < B < \infty$.*

- Let's start with a simplifying assumption: $|\mathcal{H}| < \infty$

Learnability

Definition

We say that a hypothesis class $\mathcal{H}$ is ***online learnable*** *if there exists an algorithm $\mathcal{A}$ for which $\mathcal{M}_{\mathcal{A}}(\mathcal{H}) < B < \infty$.*

- Let's start with a simplifying assumption: $|\mathcal{H}| < \infty$
- basically, we can eliminate each h with false output in every step.

Learnability

Definition

We say that a hypothesis class $\mathcal{H}$ is ***online learnable*** *if there exists an algorithm $\mathcal{A}$ for which $\mathcal{M}_{\mathcal{A}}(\mathcal{H}) < B < \infty$.*

- Let's start with a simplifying assumption: $|\mathcal{H}| < \infty$
- basically, we can eliminate each h with false output in every step. $\rightarrow$ Consistent Algorithm

Consistent Algorithm

Consistent

input: A finite hypothesis class $\mathcal{H}$
initialize: $V_1 = \mathcal{H}$
for $t = 1, 2, \ldots$
  receive $\mathbf{x}_t$
  choose any $h \in V_t$
  predict $p_t = h(\mathbf{x}_t)$
  receive true label $y_t = h^\star(\mathbf{x}_t)$
  update $V_{t+1} = \{h \in V_t : h(\mathbf{x}_t) = y_t\}$

Consistent Algorithm

Consistent

input: A finite hypothesis class $\mathcal{H}$
initialize: $V_1 = \mathcal{H}$
for $t = 1, 2, \ldots$
  receive $\mathbf{x}_t$
  choose any $h \in V_t$
  predict $p_t = h(\mathbf{x}_t)$
  receive true label $y_t = h^\star(\mathbf{x}_t)$
  update $V_{t+1} = \{h \in V_t : h(\mathbf{x}_t) = y_t\}$

- It's easy to see that:

$$\mathcal{M}_{\text{Consistent}}(\mathcal{H}) < |\mathcal{H}| - 1 \tag{4}$$

Halving

Halving

input: A finite hypothesis class $\mathcal{H}$
initialize: $V_1 = \mathcal{H}$
for $t = 1, 2, \ldots$
  receive $\mathbf{x}_t$
  predict $p_t = \operatorname{argmax}_{r \in \{0,1\}} |\{h \in V_t : h(\mathbf{x}_t) = r\}|$
    (in case of a tie predict $p_t = 1$)
  receive true label $y_t = h^\star(\mathbf{x}_t)$
  update $V_{t+1} = \{h \in V_t : h(\mathbf{x}_t) = y_t\}$

Halving

Halving

input: A finite hypothesis class $\mathcal{H}$
initialize: $V_1 = \mathcal{H}$
for $t = 1, 2, \ldots$
  receive $\mathbf{x}_t$
  predict $p_t = \text{argmax}_{r \in \{0,1\}} |\{h \in V_t : h(\mathbf{x}_t) = r\}|$
    (in case of a tie predict $p_t = 1$)
  receive true label $y_t = h^\star(\mathbf{x}_t)$
  update $V_{t+1} = \{h \in V_t : h(\mathbf{x}_t) = y_t\}$

Theorem

Let H be a finite hypothesis class. The Halving algorithm enjoys the mistake bound:

$$\mathcal{M}_{\text{Halving}}(\mathcal{H}) \leq \log_2(|\mathcal{H}|) \tag{5}$$

Halving

Proof.

Whenever the algorithm make a mistake, we will simply have:

$$|V_{t+1}| \leq \frac{|V_t|}{2}$$

Therefore, if M is the total number of mistakes, we have:

$$1 \leq |V_{T+1}| \leq |\mathcal{H}|2^{-M} \tag{6}$$

□

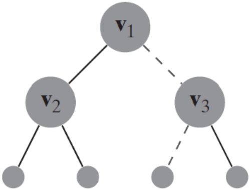
Optimality

- Learner $\Longleftrightarrow$ Adversary

Optimality

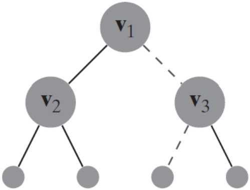
- Learner $\iff$ Adversary
- Suppose that the environment wants to have the learner make mistake on the all first T rounds of the game.Then, it must output $y_t = 1 - p_t \;\; \forall t \leq T$, and the only question is how it should choose the instances x_t in such a way that ensures that for some $h \in \mathcal{H}$ we have $y_t = h(x_t)$ for all t.

Littlestone's Dimension



	h_1	h_2	h_3	h_4
$\mathbf{v}_1$	0	0	1	1
$\mathbf{v}_2$	0	1	*	*
$\mathbf{v}_3$	*	*	0	1

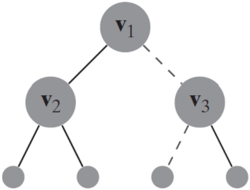
Littlestone's Dimension



	h_1	h_2	h_3	h_4
$\mathbf{v}_1$	0	0	1	1
$\mathbf{v}_2$	0	1	*	*
$\mathbf{v}_3$	*	*	0	1

- A tree of depth T
- with $2^{T+1} - 1$ nodes

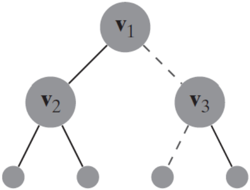
Littlestone's Dimension



	h_1	h_2	h_3	h_4
$\mathbf{v}_1$	0	0	1	1
$\mathbf{v}_2$	0	1	*	*
$\mathbf{v}_3$	*	*	0	1

- A tree of depth T
- with $2^{T+1} - 1$ nodes
- If the learner predicts $p_t = 1$ ($p_t = 0$), the adversary will declare that this is a wrong prediction and $y_t = 0$ ($y_t = 1$)! and will traverse to the left(right) child of the current node.

Littlestone's Dimension



	h_1	h_2	h_3	h_4
$\mathbf{v}_1$	0	0	1	1
$\mathbf{v}_2$	0	1	*	*
$\mathbf{v}_3$	*	*	0	1

- A tree of depth T
- with $2^{T+1} - 1$ nodes
- If the learner predicts $p_t = 1$ ($p_t = 0$), the adversary will declare that this is a wrong prediction and $y_t = 0$ ($y_t = 1$)! and will traverse to the left(right) child of the current node.
  $\longrightarrow i_{t+1} = 2i_t + y_t = 2^{t-1} + \sum_{j=1}^{t-1} y_j 2^{t-1-j}$

Littlestone's Dimension

Definition

A shattered tree of depth d is a sequence of instances $v_1, ..., v_{2^d-1}$ *in* $\mathcal{X}$ *such that for every labeling* $(y_1, \ldots, y_d) \in \{0,1\}^d$ *there exists* $h \in \mathcal{H}$ *such that for all* $t \in [d]$ *we have* $h(v_{i_t}) = y_t$ *where* $i_{t+1} = 2i_t + y_t = 2^{t-1} + \sum_{j=1}^{t-1} y_j 2^{t-1-j}$

We saw a shattered tree of depth 2 in last slide.

Definition

Littlestone's Dimension (*Ldim****)****: Ldim($\mathcal{H}$) is the maximal integer T such that there exists a shattered tree of depth T, which is shattered by $\mathcal{H}$.*

Littlestone's Dimension

Theorem

No algorithm can have a mistake bound strictly smaller than $Ldim(\mathcal{H})$*. namely, for every algorithm,* $\mathcal{A}$*, we have*

$$\mathcal{M}_{\mathcal{A}}(\mathcal{H}) \geq Ldim(\mathcal{H}) \tag{7}$$

Proof.

Let $T = Ldim(H)$. If the adversary sets $x_t = v_{i_t}$ and $y_t = 1 - p_t$ for all $t \in [T]$, then the learner makes T mistakes while the definition of $Ldim$ implies that there exists a hypothesis $h \in \mathcal{H}$ such that $y_t = h(x_t)$ for all t. □

Clearly, We have $Ldim(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$

Standard Optimal Algorithm

Standard Optimal Algorithm (SOA)

input: A hypothesis class $\mathcal{H}$
initialize: $V_1 = \mathcal{H}$
for $t = 1, 2, \ldots$
  receive $\mathbf{x}_t$
  for $r \in \{0, 1\}$ let $V_t^{(r)} = \{h \in V_t : h(\mathbf{x}_t) = r\}$
  predict $p_t = \text{argmax}_{r \in \{0,1\}} \text{Ldim}(V_t^{(r)})$
    (in case of a tie predict $p_t = 1$)
  receive true label y_t
  update $V_{t+1} = \{h \in V_t : h(\mathbf{x}_t) = y_t\}$

Theorem

SOA enjoys the mistake bound

$$\mathcal{M}_{\mathrm{SOA}}(\mathcal{H}) \leq Ldim(\mathcal{H}) \tag{8}$$

Standard Optimal Algorithm

Standard Optimal Algorithm (SOA)

input: A hypothesis class $\mathcal{H}$
initialize: $V_1 = \mathcal{H}$
for $t = 1, 2, \ldots$
  receive $\mathbf{x}_t$
  for $r \in \{0, 1\}$ let $V_t^{(r)} = \{h \in V_t : h(\mathbf{x}_t) = r\}$
  predict $p_t = \text{argmax}_{r \in \{0,1\}} \text{Ldim}(V_t^{(r)})$
    (in case of a tie predict $p_t = 1$)
  receive true label y_t
  update $V_{t+1} = \{h \in V_t : h(\mathbf{x}_t) = y_t\}$

Proof.

It suffices to show that $Ldim(V_{t+1}) \leq Ldim(V_t) - 1$
by contradiction suppose that $Ldim(V_{t+1}) = Ldim(V_t)$, then will have $Ldim(V_t^{(r)}) = Ldim(V_t)$ for $r = 0, 1$ which contracts our first assumption! □

Randomization

- Now suppose there is no such a target function h^* and adversary can set loss function whatever it wants, by this we also mean it can change target function in every step!
- If the learner was using a deterministic algorithm, it would be pretty unfair because the adversary knew the output every time. So we may want to assume a randomized setting.

Randomization

- Now suppose there is no such a target function h^* and adversary can set loss function whatever it wants, by this we also mean it can change target function in every step!
- If the learner was using a deterministic algorithm, it would be pretty unfair because the adversary knew the output every time. So we may want to assume a randomized setting. take this problem for example:

$$l_t(p_t, y_t) = \mathbf{1}\{p_t \neq y_t\}, \quad p_t, y_t \in \{0, 1\}$$

Randomization

- Now suppose there is no such a target function h^* and adversary can set loss function whatever it wants, by this we also mean it can change target function in every step!
- If the learner was using a deterministic algorithm, it would be pretty unfair because the adversary knew the output every time. So we may want to assume a randomized setting. take this problem for example:

$$l_t(p_t, y_t) = \mathbf{1}\{p_t \neq y_t\}, \quad p_t, y_t \in \{0, 1\}$$

obviously enough, adversary sets $y_t = \bar{p}_t$ and loss is 1 all the time! however if we set $p_t = 0$ with probability α and $p_t = 1$ otherwise, we have:

$$\mathbb{E}[l_t] = |\alpha - y_t| \tag{9}$$

Randomization

Another important example of randomization that will get back to it:

```
Weighted Majority
```

parameter: $\eta \in (0, 1)$
initialize: $\mathbf{w}_1 = (1/d, \ldots, 1/d)$
for $t = 1, 2, \ldots$
  choose $i \sim \mathbf{w}_t$ and predict according to the advice of the i'th expert receive costs of all experts $\mathbf{z}_t \in [0, 1]^d$
  update rule $\forall i,\ w_{t+1}[i] = \frac{w_t[i]e^{-\eta z_t[i]}}{\sum_j w_t[j]e^{-\eta z_t[j]}}$

Online Convex Optimization

```
Online Convex Optimization (OCO)
```

input: A convex set S
for $t = 1, 2, \ldots$
  predict a vector $\mathbf{w}_t \in S$
  receive a convex loss function $f_t : S \to \mathbb{R}$
  suffer loss $f_t(\mathbf{w}_t)$

The regret of the algorithm is defined as

$$\text{Regret}_T(\mathbf{u}) = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}). \tag{10}$$

Examples

- **Convex Optimization:** The adversary plays a fixed f:

$$f\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_t\right) - f(\mathbf{w}*) \le \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{w}_t) - \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{w}^*) \le \frac{\mathrm{Regret}_T(\mathbf{w}^*)}{T}$$

- **Online Linear Regression:** This problem is just an example of OCO.
  - Learner receives $\mathbf{x}_t$.
  - Learner decides $\mathbf{w}_t$.
  - Adversary plays y_t.
  - Learner pays the loss $l = |\langle \mathbf{w}, \mathbf{x}_t \rangle - y_t|$

Examples

- **Convex Optimization:** The adversary plays a fixed f:

$$f\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_t\right) - f(\mathbf{w}*) \leq \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{w}_t) - \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{w}^*) \leq \frac{\mathrm{Regret}_T(\mathbf{w}^*)}{T}$$

- **Online Linear Regression:** This problem is just an example of OCO.
  - Learner receives $\mathbf{x}_t$.
  - Learner decides $\mathbf{w}_t$.
  - Adversary plays y_t.
  - Learner pays the loss $l = |\langle \mathbf{w}, \mathbf{x}_t \rangle - y_t|$
- Other online prediction problems do not fit into the online convex optimization framework.

Examples

- **Convex Optimization:** The adversary plays a fixed f:

$$f\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_t\right) - f(\mathbf{w}*) \leq \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{w}_t) - \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{w}^*) \leq \frac{\mathrm{Regret}_T(\mathbf{w}^*)}{T}$$

- **Online Linear Regression:** This problem is just an example of OCO.
  - Learner receives $\mathbf{x}_t$.
  - Learner decides $\mathbf{w}_t$.
  - Adversary plays y_t.
  - Learner pays the loss $l = |\langle \mathbf{w}, \mathbf{x}_t \rangle - y_t|$
- Other online prediction problems do not fit into the online convex optimization framework.
- We will use convexification tricks.

Convexification: Randomization

- **Randomization**: On each round, choose from the advice of d given experts.
  - At round t, the learner chooses $\mathbf{w}_t \in S$
  - An expert p_t is chosen at random according to $\mathbf{w}_t$.
  - The cost vector $\mathbf{y}_t$ is revealed.

Convexification: Randomization

- **Randomization**: On each round, choose from the advice of d given experts.
  - At round t, the learner chooses $\mathbf{w}_t \in S$
  - An expert p_t is chosen at random according to $\mathbf{w}_t$.
  - The cost vector $\mathbf{y}_t$ is revealed.
  - Expected loss:

$$\mathbb{E}[y_t[p_t]] = \sum_{i=1}^{d} \mathbb{P}[p_t = i] y_t[i] = \langle \mathbf{w}_t, \mathbf{y}_t \rangle.$$

  - Note that the adversary does not know the outcome p_t; it is random.

Convexification: Randomization

- **Randomization**: On each round, choose from the advice of d given experts.
  - At round t, the learner chooses $\mathbf{w}_t \in S$
  - An expert p_t is chosen at random according to $\mathbf{w}_t$.
  - The cost vector $\mathbf{y}_t$ is revealed.
  - Expected loss:

$$\mathbb{E}[y_t[p_t]] = \sum_{i=1}^{d} \mathbb{P}[p_t = i] y_t[i] = \langle \mathbf{w}_t, \mathbf{y}_t \rangle.$$

  - Note that the adversary does not know the outcome p_t; it is random.
  - The regret:

$$\mathrm{Regret}_T(\mathbf{u}) = \sum_{t=1}^{T} \langle \mathbf{w}_t, \mathbf{y}_t \rangle - \langle \mathbf{w}_t, \mathbf{u} \rangle$$

Section 4

Follow The Leader

Follow the Leader

The most natural algorithm is Follow-The-Leader (FTL):

$$\forall t, \quad \mathbf{w}_t = \underset{\mathbf{w} \in S}{\operatorname{argmin}} \sum_{i=1}^{t-1} f_i(\mathbf{w})$$

Follow the Leader

The most natural algorithm is Follow-The-Leader (FTL):

$$\forall t, \quad \mathbf{w}_t = \underset{\mathbf{w} \in S}{\operatorname{argmin}} \sum_{i=1}^{t-1} f_i(\mathbf{w})$$

To analyze FTL, we first prove the following lemma:

Difference Lemma

Let $\mathbf{w}_1, \mathbf{w}_2, \ldots$ be the sequence of vectors produced by FTL. Then, for all $\mathbf{u} \in S$, we have:

$$\mathrm{Regret}_T(\mathbf{u}) = \sum_{t=1}^{T} \left(f_t(\mathbf{w}_t) - f_t(\mathbf{u}) \right) \leq \sum_{t=1}^{T} \left(f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1}) \right).$$

Equivalently,

$$\sum_{t=1}^{T} f_t(\mathbf{w}_{t+1}) \leq \sum_{t=1}^{T} f_t(\mathbf{u}). \tag{11}$$

Follow the Leader

Proof

We prove (11) by induction. The base for $T = 1$ follows from the definition of $\mathbf{w}_{t+1}$. Assume the inequality hold for $T - 1$, then for all $\mathbf{u} \in S$ we have

$$\sum_{t=1}^{T-1} f_t(\mathbf{w}_{t+1}) \leq \sum_{t=1}^{T-1} f_t(\mathbf{u}). \tag{12}$$

Follow the Leader

Proof

We prove (11) by induction. The base for $T = 1$ follows from the definition of $\mathbf{w}_{t+1}$. Assume the inequality hold for $T - 1$, then for all $\mathbf{u} \in S$ we have

$$\sum_{t=1}^{T-1} f_t(\mathbf{w}_{t+1}) \leq \sum_{t=1}^{T-1} f_t(\mathbf{u}). \tag{12}$$

Adding $f_T(\mathbf{w}_{T+1})$ to both sides, we get

$$\sum_{t=1}^{T} f_t(\mathbf{w}_{t+1}) \leq f_T(\mathbf{w}_{T+1}) + \sum_{t=1}^{T-1} f_t(\mathbf{u}). \tag{13}$$

Follow the Leader

Proof

We prove (11) by induction. The base for $T = 1$ follows from the definition of $\mathbf{w}_{t+1}$. Assume the inequality hold for $T-1$, then for all $\mathbf{u} \in S$ we have

$$\sum_{t=1}^{T-1} f_t(\mathbf{w}_{t+1}) \leq \sum_{t=1}^{T-1} f_t(\mathbf{u}). \tag{12}$$

Adding $f_T(\mathbf{w}_{T+1})$ to both sides, we get

$$\sum_{t=1}^{T} f_t(\mathbf{w}_{t+1}) \leq f_T(\mathbf{w}_{T+1}) + \sum_{t=1}^{T-1} f_t(\mathbf{u}). \tag{13}$$

The above holds for all $\mathbf{u}$ and in particular for $\mathbf{u} = \mathbf{w}_{T+1}$. Thus,

$$\sum_{t=1}^{T} f_t(\mathbf{w}_{t+1}) \leq \sum_{t=1}^{T} f_t(\mathbf{w}_{T+1}) = \min_{\mathbf{u}\in S} \sum_{t=1}^{T} f_t(\mathbf{u}). \tag{14}$$

Online Quadratic Optimization

Here we prove a regret bound for a subset of OCO in which $S = \mathbb{R}^d$ at each round t, we have $f_t(\mathbf{w}) = ||\mathbf{w} - \mathbf{z}_t||_2^2$ for some $\mathbf{z}_t$.

Online Quadratic Optimization

Here we prove a regret bound for a subset of OCO in which $S = \mathbb{R}^d$ at each round t, we have $f_t(\mathbf{w}) = ||\mathbf{w} - \mathbf{z}_t||_2^2$ for some $\mathbf{z}_t$.

- $\forall t, \quad \mathbf{w}_t = \text{argmin}_{\mathbf{w} \in S} \sum_{i=1}^{t-1} f_i(\mathbf{w}) \implies \mathbf{w}_t = \frac{1}{t-1} \sum_{i=1}^{t-1} \mathbf{z}_i.$

  Note that we can rewrite

$$\mathbf{w}_{t+1} = \frac{1}{t}\left(\mathbf{z}_t + (t-1)\mathbf{w}_t\right),$$

  which yields

$$\mathbf{w}_{t+1} - \mathbf{z}_t = \left(1 - \frac{1}{t}\right)(\mathbf{w}_t - \mathbf{z}_t).$$

Online Quadratic Optimization

Here we prove a regret bound for a subset of OCO in which $S = \mathbb{R}^d$ at each round t, we have $f_t(\mathbf{w}) = ||\mathbf{w} - \mathbf{z}_t||_2^2$ for some $\mathbf{z}_t$.

- $\forall t, \quad \mathbf{w}_t = \text{argmin}_{\mathbf{w} \in S} \sum_{i=1}^{t-1} f_i(\mathbf{w}) \implies \mathbf{w}_t = \frac{1}{t-1} \sum_{i=1}^{t-1} \mathbf{z}_i$.

  Note that we can rewrite

$$\mathbf{w}_{t+1} = \frac{1}{t} \left(\mathbf{z}_t + (t-1)\mathbf{w}_t \right),$$

  which yields

$$\mathbf{w}_{t+1} - \mathbf{z}_t = \left(1 - \frac{1}{t}\right)(\mathbf{w}_t - \mathbf{z}_t).$$

- Therefore,

$$\begin{aligned} f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1}) &= \frac{1}{2}||\mathbf{w}_t - \mathbf{z}_t||^2 - \frac{1}{2}||\mathbf{w}_{t+1} - \mathbf{z}_t||^2 \\ &= \frac{1}{2}\left(1 - (1 - \frac{1}{t})^2\right)||\mathbf{w}_t - \mathbf{z}_t||^2 \leq \frac{1}{t}||\mathbf{w}_t - \mathbf{z}_t||^2. \end{aligned}$$

For the quadratic OCO, we have

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1}) \leq \frac{1}{t}||\mathbf{w}_t - \mathbf{z}_t||^2.$$

For the quadratic OCO, we have

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1}) \leq \frac{1}{t}||\mathbf{w}_t - \mathbf{z}_t||^2.$$

Let $L = \max_t ||\mathbf{z}_t||$. Since $\mathbf{w}_t$ is the average of $\mathbf{z}_t$, it also holds that $\mathbf{w}_t \leq L$. By the triangle inequality $||\mathbf{w}_t - \mathbf{z}_t|| \leq 2L$. Hence,

$$\sum_{t=1}^{T}(f_t(\mathbf{w}_t) - f_t)(\mathbf{w}_{t+1})) \leq (2L)^2 \sum_{t=1}^{T} \frac{1}{t} \leq (2L)^2 \big(1 + \log(T)\big).$$

For the quadratic OCO, we have

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1}) \leq \frac{1}{t}||\mathbf{w}_t - \mathbf{z}_t||^2.$$

Let $L = \max_t ||\mathbf{z}_t||$. Since $\mathbf{w}_t$ is the average of $\mathbf{z}_t$, it also holds that $\mathbf{w}_t \leq L$. By the triangle inequality $||\mathbf{w}_t - \mathbf{z}_t|| \leq 2L$. Hence,

$$\sum_{t=1}^{T}(f_t(\mathbf{w}_t) - f_t)(\mathbf{w}_{t+1})) \leq (2L)^2 \sum_{t=1}^{T} \frac{1}{t} \leq (2L)^2 \big(1 + \log(T)\big).$$

Using Lemma 1, we have

$$\begin{aligned}\mathrm{Regret}_T(\mathbf{u}) = \sum_{t=1}^{T} \big(f_t(\mathbf{w}_t) - f_t(\mathbf{u})\big) &\leq \sum_{t=1}^{T} \big(f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1})\big) \\ &\leq (2L)^2 \big(1 + \log(T)\big).\end{aligned}$$

Follow the Leader

- One might wonder if the algorithm always works!

Follow the Leader

- One might wonder if the algorithm always works! The answer is negative. Consider a 1D online linear optimization: $f_t(w) = z_t w$.

Follow the Leader

- One might wonder if the algorithm always works! The answer is negative. Consider a 1D online linear optimization: $f_t(w) = z_t w$.
- Let $S = [-1, 1]$ and

$$
\begin{aligned}
z_1 &= -0.5 \\
z_t &= 1, \; t = 2, 4, \ldots \\
z_t &= -1, \; t = 3, 5, \ldots
\end{aligned}
$$

- The prediction of FTL will be set to $w_t = 1$ for t odd and $w_t = -1$ for t even.

Follow the Leader

- One might wonder if the algorithm always works! The answer is negative. Consider a 1D online linear optimization: $f_t(w) = z_t w$.
- Let $S = [-1, 1]$ and

$$z_1 = -0.5$$
$$z_t = 1, \ t = 2, 4, \dots$$
$$z_t = -1, \ t = 3, 5, \dots$$

- The prediction of FTL will be set to $w_t = 1$ for t odd and $w_t = -1$ for t even.
  - The cumulative loss of FTL: T.
  - The cumulative loss of the fixed solution $u = 0 \in S$ is 0.

Follow the Leader

- One might wonder if the algorithm always works! The answer is negative. Consider a 1D online linear optimization: $f_t(w) = z_t w$.
- Let $S = [-1, 1]$ and

$$z_1 = -0.5$$
$$z_t = 1, \; t = 2, 4, \ldots$$
$$z_t = -1, \; t = 3, 5, \ldots$$

- The prediction of FTL will be set to $w_t = 1$ for t odd and $w_t = -1$ for t even.
  - The cumulative loss of FTL: T.
  - The cumulative loss of the fixed solution $u = 0 \in S$ is 0.
- Hence, the regret is $O(T)$!

Follow the Leader

- One might wonder if the algorithm always works! The answer is negative. Consider a 1D online linear optimization: $f_t(w) = z_t w$.
- Let $S = [-1, 1]$ and

$$
\begin{aligned}
z_1 &= -0.5 \\
z_t &= 1,\ t = 2, 4, \ldots \\
z_t &= -1,\ t = 3, 5, \ldots
\end{aligned}
$$

- The prediction of FTL will be set to $w_t = 1$ for t odd and $w_t = -1$ for t even.
  - The cumulative loss of FTL: T.
  - The cumulative loss of the fixed solution $u = 0 \in S$ is 0.
- Hence, the regret is $O(T)$!
- Intuitively, FTL fails in the above example because its predictions are not stable.

Section 5

Follow the Regularized Leader

Follow the Regularized Leader

Follow-the-Regularized-Leader is a natural modification of the basic FTL algorithm.

$$\forall t, \quad \mathbf{w}_t = \operatorname*{argmin}_{\mathbf{w} \in S} \sum_{i=1}^{t-1} f_i(\mathbf{w}) + R(\mathbf{w})$$

- We now study the regret under strongly convex regularizes.

We will now analyze the regret:

$$\mathrm{Regret}_T(\mathbf{u}) = \sum_{t=1}^{T} \Big(f_t(\mathbf{w}_t) - f_t(\mathbf{u}) \Big)$$

Running FTRL on $f_1, \dots, f_t$ is equivalent to running FTL on $f_0, \dots, f_T$ where $f_0 = R$. Hence from the Difference Lemma, we have

$$\sum_{t=0}^{T} \Big(f_t(\mathbf{w}_t) - f_t(\mathbf{u}) \Big) \leq \sum_{t=0}^{T} \Big(f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1}) \Big)$$

We will now analyze the regret:

$$\mathrm{Regret}_T(\mathbf{u}) = \sum_{t=1}^{T} \Big(f_t(\mathbf{w}_t) - f_t(\mathbf{u}) \Big)$$

Running FTRL on $f_1, \ldots, f_t$ is equivalent to running FTL on $f_0, \ldots, f_T$ where $f_0 = R$. Hence from the Difference Lemma, we have

$$\sum_{t=0}^{T} \Big(f_t(\mathbf{w}_t) - f_t(\mathbf{u}) \Big) \leq \sum_{t=0}^{T} \Big(f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1}) \Big)$$

Rearranging terms, we arrive at:

$$\sum_{t=1}^{T} \Big(f_t(\mathbf{w}_t) - f_t(\mathbf{u}) \Big) \leq R(\mathbf{u}) - R(\mathbf{w}_1) + \sum_{t=1}^{T} \Big(f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1}) \Big)$$

Strongly Convex Regularizers

We will now analyze FTRL with strongly convex regularizers.

$$\mathrm{Regret}_T(\mathbf{u}) \leq R(\mathbf{u}) - R(\mathbf{w}_1) + \sum_{t=1}^{T} \Big(f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1}) \Big)$$
$$\leq R(\mathbf{u}) - R(\mathbf{w}_1) + \sum_{t=1}^{T} L||\mathbf{w}_t - \mathbf{w}_{t+1}||$$

So we need to ensure $||\mathbf{w}_t - \mathbf{w}_{t+1}||$ is small.

Let $F_t = \sum_{i=1}^{t-1} f_i(\mathbf{w}) + R(\mathbf{w})$ and note that $\mathbf{w}_t = \text{argmin}_{\mathbf{w} \in S} F_t(\mathbf{w})$.
Since $\mathbf{w}_t$ is the minimizer, by the strong convexity property we have

$$F_t(\mathbf{w}_{t+1}) \geq F_t(\mathbf{w}_t) + \frac{\sigma}{2}||\mathbf{w}_t - \mathbf{w}_{t+1}||^2$$

Reteating the same argument for F_{t+1} and minimizer $\mathbf{w}_{t+1}$:

$$F_{t+1}(\mathbf{w}_t) \geq F_{t+1}(\mathbf{w}_{t+1}) + \frac{\sigma}{2}||\mathbf{w}_t - \mathbf{w}_{t+1}||^2$$

Let $F_t = \sum_{i=1}^{t-1} f_i(\mathbf{w}) + R(\mathbf{w})$ and note that $\mathbf{w}_t = \operatorname{argmin}_{\mathbf{w} \in S} F_t(\mathbf{w})$. Since $\mathbf{w}_t$ is the minimizer, by the strong convexity property we have

$$F_t(\mathbf{w}_{t+1}) \geq F_t(\mathbf{w}_t) + \frac{\sigma}{2}||\mathbf{w}_t - \mathbf{w}_{t+1}||^2$$

Reteating the same argument for F_{t+1} and minimizer $\mathbf{w}_{t+1}$:

$$F_{t+1}(\mathbf{w}_t) \geq F_{t+1}(\mathbf{w}_{t+1}) + \frac{\sigma}{2}||\mathbf{w}_t - \mathbf{w}_{t+1}||^2$$

Summing the above inequalities and using Lipschitzness of f_t:

$$\sigma||\mathbf{w}_t - \mathbf{w}_{t+1}||^2 \leq f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1}) \leq L||\mathbf{w}_t - \mathbf{w}_{t+1}||$$

This implies

$$||\mathbf{w}_t - \mathbf{w}_{t+1}||^2 \leq \frac{L}{\sigma}.$$

$$
\begin{aligned}
\mathrm{Regret}_T(\mathbf{u}) &\leq R(\mathbf{u}) - R(\mathbf{w}_1) + \sum_{t=1}^{T} \Big(f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1}) \Big) \\
&\leq R(\mathbf{u}) - R(\mathbf{w}_1) + \sum_{t=1}^{T} L||\mathbf{w}_t - \mathbf{w}_{t+1}|| \\
&\leq R(\mathbf{u}) - \min R + TL^2/\sigma
\end{aligned}
$$

Euclidean Regularization

Corollary

Let $f_1, \dots, f_T$ be a sequence of convex and L-Lipschitz functions with respect to $||.||_2$. FTRL is run on the sequence with $R(\mathbf{w}) = \frac{1}{2\eta}||\mathbf{w}||_2^2$.

$$\forall \mathbf{u} : \quad \mathrm{Regret}_T(\mathbf{u}) \leq \frac{1}{2\eta}||\mathbf{u}||_2^2 + \eta T L^2.$$

In particular, if $U = \{\mathbf{u} : ||\mathbf{u}||_2 \leq B\}$ and $\eta = \frac{B}{L\sqrt{2T}}$, then

$$\mathrm{Regret}_T(U) \leq BL\sqrt{2T}.$$

Expert Advise

Corollary

Assume that the conditions of the previous corollary hold. Let S be a convex set. Define

$$R(\mathbf{w}) = \begin{cases} \frac{1}{2\eta}||\mathbf{w}||_2^2 & \mathbf{w} \in S \\ \infty & \mathbf{w} \notin S \end{cases}$$

Then $\forall \mathbf{u} \in S: \quad \mathrm{Regret}_T(\mathbf{u}) \leq \frac{1}{2\eta}||\mathbf{u}||_2^2 + \eta T L^2$.
In particular, if $B \geq \max_{\mathbf{u}\in S} ||\mathbf{u}||_2$ and $\eta = \frac{B}{L\sqrt{2T}}$, then

$$\mathrm{Regret}_T(S) \leq BL\sqrt{2T}.$$

Expert Advise

Corollary

Assume that the conditions of the previous corollary hold. Let S be a convex set. Define

$$R(\mathbf{w}) = \begin{cases} \frac{1}{2\eta}||\mathbf{w}||_2^2 & \mathbf{w} \in S \\ \infty & \mathbf{w} \notin S \end{cases}$$

Then $\forall \mathbf{u} \in S: \quad \mathrm{Regret}_T(\mathbf{u}) \leq \frac{1}{2\eta}||\mathbf{u}||_2^2 + \eta T L^2.$

In particular, if $B \geq \max_{\mathbf{u} \in S} ||\mathbf{u}||_2$ and $\eta = \frac{B}{L\sqrt{2T}}$, then

$$\mathrm{Regret}_T(S) \leq BL\sqrt{2T}.$$

In the expert advice setting, S is the probability simplex and $\mathbf{x}_t \in [0,1]^d$. We can set $L = \sqrt{d}$ and $B = 1$ which leads to a regret bound $\sqrt{2dT}$.

Expert Advise

Corollary

Assume that the conditions of the previous corollary hold. Let S be a convex set. Define

$$R(\mathbf{w}) = \begin{cases} \frac{1}{2\eta}||\mathbf{w}||_2^2 & \mathbf{w} \in S \\ \infty & \mathbf{w} \notin S \end{cases}$$

Then $\forall \mathbf{u} \in S: \quad \mathrm{Regret}_T(\mathbf{u}) \leq \frac{1}{2\eta}||\mathbf{u}||_2^2 + \eta T L^2$.
In particular, if $B \geq \max_{\mathbf{u}\in S} ||\mathbf{u}||_2$ and $\eta = \frac{B}{L\sqrt{2T}}$, then

$$\mathrm{Regret}_T(S) \leq BL\sqrt{2T}.$$

In the expert advice setting, S is the probability simplex and $\mathbf{x}_t \in [0,1]^d$. We can set $L = \sqrt{d}$ and $B = 1$ which leads to a regret bound $\sqrt{2dT}$. The Entropic Regularization leads to $\sqrt{2\log(d)T}$

Section 6

Online Mirror Descent

Online Mirror Descent

- FTRL involves solving an optimization in each round.
- We will show that Online Mirror Descent achieves the same regret bound as FTRL
- It is capable of introducing a variety of new algorithms
- Notation: $z_{1:t} = \sum_{i=1}^{t} \mathbf{z}_i$.

General OMD settings

Online Mirror Descent (OMD)

parameter: a link function $g : \mathbb{R}^d \to S$
initialize: $\boldsymbol{\theta}_1 = \mathbf{0}$
for $t = 1, 2, \ldots$
  predict $\mathbf{w}_t = g(\boldsymbol{\theta}_t)$
  update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mathbf{z}_t$ where $\mathbf{z}_t \in \partial f_t(\mathbf{w}_t)$

- Choosing different g's leads to differnet algorithms
- For instance, taking $g(x) = x$ results in OGD.
- θ is updated by subtracting the gradient out of it, but the actual prediction is "mirrored" or "linked" to the set S via the function g.

General OMD settings

```
Online Mirror Descent (OMD)
```

parameter: a link function $g : \mathbb{R}^d \rightarrow S$
initialize: $\boldsymbol{\theta}_1 = \mathbf{0}$
for $t = 1, 2, \ldots$
  predict $\mathbf{w}_t = g(\boldsymbol{\theta}_t)$
  update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mathbf{z}_t$ where $\mathbf{z}_t \in \partial f_t(\mathbf{w}_t)$

- Choosing different g's leads to differnet algorithms
- For instance, taking $g(x) = x$ results in OGD.
- θ is updated by subtracting the gradient out of it, but the actual prediction is "mirrored" or "linked" to the set S via the function g.
- We will show that it is **equivalent to FTRL** for some specific regularization.

If f_t are convex nonlinear functions, we have

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - f_t(\mathbf{u}) \leq \sum_{t=1}^{T} \langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t \rangle$$

So from now, we will consider the OLO problem.

If f_t are convex nonlinear functions, we have

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - f_t(\mathbf{u}) \leq \sum_{t=1}^{T} \langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t \rangle$$

So from now, we will consider the OLO problem.
Consider the FTRL update:

$$\begin{aligned}\mathbf{w}_{t+1} &= \text{argmin}\, R(\mathbf{w}) + \sum_{i=1}^{t} \langle \mathbf{w}, \mathbf{z}_t \rangle \\ &= \text{argmin}\, R(\mathbf{w}) + \langle \mathbf{w}, \mathbf{z}_{1:t} \rangle \\ &= \text{argmax} -R(\mathbf{w}) + \langle \mathbf{w}, -\mathbf{z}_{1:t} \rangle\end{aligned}$$

If f_t are convex nonlinear functions, we have

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - f_t(\mathbf{u}) \leq \sum_{t=1}^{T} \langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t \rangle$$

So from now, we will consider the OLO problem.
Consider the FTRL update:

$$\begin{aligned} \mathbf{w}_{t+1} &= \operatorname{argmin} R(\mathbf{w}) + \sum_{i=1}^{t} \langle \mathbf{w}, \mathbf{z}_t \rangle \\ &= \operatorname{argmin} R(\mathbf{w}) + \langle \mathbf{w}, \mathbf{z}_{1:t} \rangle \\ &= \operatorname{argmax} -R(\mathbf{w}) + \langle \mathbf{w}, -\mathbf{z}_{1:t} \rangle \end{aligned}$$

Let $g(\theta) = \operatorname{argmax}_{\mathbf{w}} \langle \mathbf{w}, \theta \rangle - R(\mathbf{w})$, we can write FTRL as the following recursive rule:

$$\begin{cases} \mathbf{w}_t = g(\theta_t) \\ \theta_{t+1} = \theta_t - \mathbf{z}_t \end{cases}$$

Preliminaries

Reminder

- **Conjugate function**:

$$f^*(\theta) = \max_{\mathbf{u}} \langle \mathbf{u}, \theta \rangle - f(\mathbf{u}).$$

- **Fenchel-Young's Inequality**:

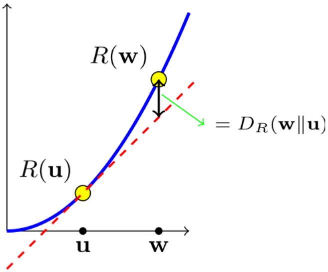
$$\forall \mathbf{u}, \quad f^*(\theta) + f(\mathbf{u}) \geq \langle \mathbf{u}, \theta \rangle$$

- **Bregman's Divergence**: A differentiable convex function R defines a Bregman divergence between two vectors as follows:

$$D_R(\mathbf{w}||\mathbf{u}) = R(\mathbf{w}) - \left(R(\mathbf{u}) + \langle \nabla R(\mathbf{u}), \mathbf{w} - \mathbf{u} \rangle\right) \geq 0$$

For example $R(w) = \frac{1}{2}||w||_2^2$ gives $D_R(\mathbf{w}||\mathbf{u}) = ||w - u||_2^2$ and $R(w) = \sum_i w[i] \log(w[i])$ gives KL-divergence.

Preliminaries



Preliminaries

- **Strong-Convexity**:

$$D_R(\mathbf{w}||\mathbf{u}) \geq \frac{\sigma}{2}||\mathbf{w} - \mathbf{u}||^2.$$

- **Strong-Smoothness**:

$$D_R(\mathbf{w}||\mathbf{u}) \leq \frac{\sigma}{2}||\mathbf{w} - \mathbf{u}||^2.$$

Preliminaries

- **Strong-Convexity**:

$$D_R(\mathbf{w}||\mathbf{u}) \geq \frac{\sigma}{2}||\mathbf{w} - \mathbf{u}||^2.$$

- **Strong-Smoothness**:

$$D_R(\mathbf{w}||\mathbf{u}) \leq \frac{\sigma}{2}||\mathbf{w} - \mathbf{u}||^2.$$

Lemma

(Strong/Smooth Duality) Assume that R is a closed and convex function. Then R is β-strongly convex with respect to a norm $||.||$ if and only if R^* is $\frac{1}{\beta}$-strongly smooth with respect to the dual norm $||.||_*$

Preliminaries

Lemma

It is possible to show that equality in Fenchel-Young Inequality holds if u is a sub-gradient of f^* at θ and in particular, if f^* is differentiable, equality holds when $\mathbf{u} = \nabla f^*(\theta)$. In the same way, $\theta = \nabla f(u)$

Preliminaries

Lemma

It is possible to show that equality in Fenchel-Young Inequality holds if u is a sub-gradient of f^* at θ and in particular, if f^* is differentiable, equality holds when $\mathbf{u} = \nabla f^*(\theta)$. In the same way, $\theta = \nabla f(u)$

Recall that $g(\theta) = \text{argmax}_{\mathbf{w}} \langle \mathbf{w}, \theta \rangle - R(\mathbf{w})$. then:

$$g(\theta) = \nabla R^*(\theta) \tag{15}$$

Analysis of OMD

Lemma

Suppose that OMD is run with a link function $g(\theta) = \nabla R^*(\theta)$ Then, its regret is upper bounded by:

$$\sum_{t=1}^{T} \langle w_t - \mathbf{u}, z_t \rangle \leq R(\mathbf{u}) - R(w_1) + \sum_{t=1}^{T} D_{R^*}(-z_{1:t}|| - z_{1:t-1}). \tag{16}$$

Analysis of OMD

Proof.

Using Fenchel–Young inequality we have:

$$R(\mathbf{u}) + \sum_{t=1}^{T} \langle \mathbf{u}, z_t \rangle = R(\mathbf{u}) - \langle \mathbf{u}, -z_{1:T} \rangle \geq -R^*(-z_{1:T})$$

Analysis of OMD

Proof.

Using Fenchel–Young inequality we have:

$$R(\mathbf{u}) + \sum_{t=1}^{T} \langle \mathbf{u}, z_t \rangle = R(\mathbf{u}) - \langle \mathbf{u}, -z_{1:T} \rangle \geq -R^*(-z_{1:T})$$

if we rewrite the RHS as:

$$-R^*(-z_{1:t}) = -R^*(0) - \sum_{t=1}^{T} (R^*(-z_{1:t}) - R^*(-z_{1:t-1}))$$

Analysis of OMD

Proof.

Using Fenchel–Young inequality we have:

$$R(\mathbf{u}) + \sum_{t=1}^{T} \langle \mathbf{u}, z_t \rangle = R(\mathbf{u}) - \langle \mathbf{u}, -z_{1:T} \rangle \geq -R^*(-z_{1:T})$$

if we rewrite the RHS as:

$$-R^*(-z_{1:t}) = -R^*(0) - \sum_{t=1}^{T} (R^*(-z_{1:t}) - R^*(-z_{1:t-1}))$$

knowing that $w_t = \nabla R^*(-z_{1:t-1})$:

$$= -R^*(0) + \sum_{t=1}^{T} (\langle w_t, z_t \rangle - D_{R^*}(-z_{1:t} || - z_{1:t-1}))$$

Note that $R^*(\mathbf{0}) = max_w\{\langle \mathbf{0}, w \rangle - R(w)\} = -min_w\{R(w)\} = -R(w_1)$
Combining all the above concludes the proof.

Analysis of OMD

Corollary

Let R be a $\frac{1}{\eta}$-strongly convex with respect to a norm $||.||$ and suppose the OMD algorithm is run with the link function $g = \nabla R^*$, Then:

$$\sum_{t=1}^{T} \langle w_t - \mathbf{u}, z_t \rangle \leq R(\mathbf{u}) - R(w_1) + \frac{\eta}{2} \sum_{t=1}^{T} ||z_t||_*^2$$

That is what we had for OGD, which is reassuring

Derived Algorithms

Normalized Exponentiated Gradient

Let S be the probability simplex and $g : \mathbb{R}^d \to \mathbb{R}^d$ be a vector valued function whose i'th component is

$$g_i(\theta) = \frac{\exp(\eta\,\theta[i])}{\sum_j \exp(\eta\,\theta[j])} \iff R(\mathbf{w}) = \frac{1}{\eta} \sum_i w[i] \log(w[i]) \text{ on } S$$

The update of OMD with this function is

$$w_{t+1}[i] = \frac{w_t[i] \exp(-\eta z_t[i])}{\sum_j w_t[j] \exp(-\eta z_t[j])}$$

Derived Algorithms

Normalized Exponentiated Gradient

Let S be the probability simplex and $g : \mathbb{R}^d \to \mathbb{R}^d$ be a vector valued function whose i'th component is

$$g_i(\theta) = \frac{\exp(\eta\,\theta[i])}{\sum_j \exp(\eta\,\theta[j])} \iff R(\mathbf{w}) = \frac{1}{\eta}\sum_i w[i]\log(w[i]) \text{ on } S$$

The update of OMD with this function is

$$w_{t+1}[i] = \frac{w_t[i]\exp(-\eta z_t[i])}{\sum_j w_t[j]\exp(-\eta z_t[j])}$$

Theorem

Assume that the normalized EG algorithm is run on a sequence of linear loss functions such that for all t, i we have $\eta z_t[i] \geq -1$. Then:

$$\sum_{t=1}^{T}\langle w_t - \mathbf{u}, z_t\rangle \leq \frac{\log(d)}{\eta} + \eta\sum_{t=1}^{T}\sum_{i} w_t[i]z_t[i]^2 \tag{17}$$

proof.

it suffices to show that:

$$D_{R^*}(-z_{1:t}||-z_{1:t-1}) \leq \eta \sum_i w_t[i] z_t[i]^2$$

proof.

it suffices to show that:

$$D_{R^*}(-z_{1:t}||-z_{1:t-1}) \leq \eta \sum_i w_t[i] z_t[i]^2$$

the conjugate function of $R(\mathbf{w}) = \frac{1}{\eta}\sum_i w[i]\log(w[i])$ is:

$$R^*(\theta) = \frac{1}{\eta}\log(\sum_i e^{\eta\theta[i]})$$

proof.

it suffices to show that:

$$D_{R^*}(-z_{1:t}||-z_{1:t-1}) \leq \eta \sum_i w_t[i] z_t[i]^2$$

the conjugate function of $R(\mathbf{w}) = \frac{1}{\eta}\sum_i w[i]\log(w[i])$ is:

$$R^*(\theta) = \frac{1}{\eta}\log(\sum_i e^{\eta\theta[i]})$$

then:

$$\begin{aligned} D_{R^*}(-z_{1:t}||-z_{1:t-1}) &= -R^*(-z_{1:t}) - R^*(-z_{1:t-1}) + \langle w_t, z_t \rangle \\ &= \frac{1}{\eta}\log(\frac{\sum_i e^{-\eta z_{1:t}[i]}}{\sum_i e^{-\eta z_{1:t-1}[i]}}) + \langle w_t, z_t \rangle \\ &= \frac{1}{\eta}\log(\sum_i w_t[i] e^{-\eta z_t[i]}) + \langle w_t, z_t \rangle \end{aligned}$$

proof.

Using numeric inequality: $e^{-a} \leq 1 - a + a^2 \; a \geq -1$, we obtain:

$$D_{R^*}(-z_{1:t}|| - z_{1:t-1}) \leq \frac{1}{\eta} \log(\sum_i w_t[i](1 - \eta z_t[i] + \eta^2 z_t[i]^2)) + \langle w_t, z_t \rangle$$

proof.

Using numeric inequality: $e^{-a} \leq 1 - a + a^2 \; a \geq -1$, we obtain:

$$D_{R^*}(-z_{1:t}|| - z_{1:t-1}) \leq \frac{1}{\eta} \log(\sum_i w_t[i](1 - \eta z_t[i] + \eta^2 z_t[i]^2)) + \langle w_t, z_t \rangle$$

and the inequality $\log(1 - a) \leq -a$,

$$D_{R^*}(-z_{1:t}|| - z_{1:t-1}) \leq \frac{1}{\eta} \sum_i w_t[i](-\eta z_t[i] + \eta^2 z_t[i]^2) + \langle w_t, z_t \rangle$$

$$= \eta \sum_i w_t[i] z_t[i]^2$$

Derived Algorithms

L_p Algorithm

Let $g : \mathbb{R}^d \to \mathbb{R}^d$ be a vector valued function with

$$g_i(\theta) = \eta \frac{\mathrm{sign}(\theta[i])\left|\theta[i]\right|^{p-1}}{||\theta||_p^{p-2}}$$

$g(\theta)$ is the update corresponding to $R(\mathbf{w}) = \frac{1}{2\eta(q-1)}||\mathbf{w}||_q^2$ where $\frac{1}{p} + \frac{1}{q} = 1$ and R is $\frac{1}{\eta}$-strongly convex with respect to l_q norm.

Corollary

Let $f_1, \ldots, f_T$ be a sequence of convex and L-Lipschitz function over $\mathbb{R}^d$ with respect to $||.||_q$. Then for all $\mathbf{u}$ for the L_p algorithm we have

$$\mathrm{Regret}_T(\mathbf{u}) \leq \frac{1}{2\eta(q-1)}||\mathbf{w}||_q^2 + \eta T L^2$$

Derived Algorithms

L_p Algorithm

Let $g : \mathbb{R}^d \to \mathbb{R}^d$ be a vector valued function with

$$g_i(\theta) = \eta \frac{\text{sign}(\theta[i]) \Big|\theta[i]\Big|^{p-1}}{||\theta||_p^{p-2}}$$

$g(\theta)$ is the update corresponding to $R(\mathbf{w}) = \frac{1}{2\eta(q-1)}||\mathbf{w}||_q^2$ where $\frac{1}{p} + \frac{1}{q} = 1$ and R is $\frac{1}{\eta}$-strongly convex with respect to l_q norm.

Corollary

Let $f_1, \ldots, f_T$ be a sequence of convex and L-Lipschitz function over $\mathbb{R}^d$ with respect to $||.||_q$. Then for all $\mathbf{u}$ for the L_p algorithm we have

$$\text{Regret}_T(\mathbf{u}) \leq \frac{1}{2\eta(q-1)}||\mathbf{w}||_q^2 + \eta T L^2$$

If $||\mathbf{u}||_q \leq B$ and $\eta = \frac{B}{L\sqrt{2T/(q-1)}}$ then $\text{Regret}_T(U) \leq BL\sqrt{\frac{2T}{q-1}}$.

Section 7

Bandits

In this section:

1. Review
2. Multi-Armed Bandits (Adversarial)
    1. Multi-Armed Bandits Algorithm
3. Multi-Armed Bandits (Stochastic)
    1. Explore-Then-Commit
    2. Upper Confidence Bound

Bandits: Introduction

What we have done so far:

Bandits: Introduction

What we have done so far:

- **Online Convex Optimization**

```
Online Convex Optimization (OCO)
```

input: A convex set S
for $t = 1, 2, \ldots$
  predict a vector $\mathbf{w}_t \in S$
  receive a convex loss function $f_t : S \to \mathbb{R}$
  suffer loss $f_t(\mathbf{w}_t)$

Limited Feedback (Bandits)

- Recall the OMD algorithm we described in last section.

```
Online Mirror Descent (OMD)
```

parameter: a link function $g : \mathbb{R}^d \to S$
initialize: $\boldsymbol{\theta}_1 = \mathbf{0}$
for $t = 1, 2, \ldots$
  predict $\mathbf{w}_t = g(\boldsymbol{\theta}_t)$
  update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mathbf{z}_t$ where $\mathbf{z}_t \in \partial f_t(\mathbf{w}_t)$

- What if we won't be given $\mathbf{z}_t$ after each step?

Limited Feedback (Bandits)

- Recall the OMD algorithm we described in last section.

```
Online Mirror Descent (OMD)
```

parameter: a link function $g : \mathbb{R}^d \to S$
initialize: $\boldsymbol{\theta}_1 = \mathbf{0}$
for $t = 1, 2, \ldots$
  predict $\mathbf{w}_t = g(\boldsymbol{\theta}_t)$
  update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mathbf{z}_t$ where $\mathbf{z}_t \in \partial f_t(\mathbf{w}_t)$

- What if we won't be given $\mathbf{z}_t$ after each step?
  Remember z_t was, For instance, in case of linear loss, vector constructed by expert's losses!

Limited Feedback (Bandits)

- Recall the OMD algorithm we described in last section.

```
Online Mirror Descent (OMD)
```

parameter: a link function $g : \mathbb{R}^d \to S$
initialize: $\boldsymbol{\theta}_1 = \mathbf{0}$
for $t = 1, 2, \dots$
  predict $\mathbf{w}_t = g(\boldsymbol{\theta}_t)$
  update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mathbf{z}_t$ where $\mathbf{z}_t \in \partial f_t(\mathbf{w}_t)$

- What if we won't be given $\mathbf{z}_t$ after each step?
Remember z_t was, For instance, in case of linear loss, vector constructed by expert's losses!
- Therefor, It's natural to assume that we are just given $\mathbf{z}_t[i]$ with probability $\mathbf{w}_t[i]$.

Bandit $\equiv$ Limited Feedback

Bandit $\equiv$ Limited Feedback

The learner knows $f_t(\mathbf{w}_t)$ but not the function f_t or its drrivative $\mathbf{z}_t \in \partial f_t(\mathbf{w}_t)$.

Limited Feedback (Bandits)

- An unbiased estimator of z_t might suffices.

Limited Feedback (Bandits)

- An unbiased estimator of z_t might suffices.

```
Online Mirror Descent with Estimated
Gradients
```

parameter: a link function $g : \mathbb{R}^d \to S$
initialize: $\boldsymbol{\theta}_1 = \mathbf{0}$
for $t = 1, 2, \ldots$
predict $\mathbf{w}_t = g(\boldsymbol{\theta}_t)$
pick $\mathbf{z}_t$ at random such that $\mathbb{E}[\mathbf{z}_t | \mathbf{z}_{t-1}, \ldots, \mathbf{z}_1] \in \partial f_t(\mathbf{w}_t)$
update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mathbf{z}_t$

Limited Feedback (Bandits)

Theorem

Suppose that the estimated sub-gradients are chosen such that with probability 1 we have:

$$\sum_{i=1}^{T}\langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t\rangle \leq B(\mathbf{u}) + \sum_{i=1}^{T} ||\mathbf{z}_t||_t^2$$

where B is some function, and for all round t the norm $||.||_t$ may depend on w_t. Then:

$$\mathbb{E}[\sum_{i=1}^{T} f_t(\mathbf{w}_t) - f_t(\mathbf{u})] \leq B(\mathbf{u}) + \sum_{i=1}^{T}\mathbb{E}[||\mathbf{z}_t||_t^2]$$

Where the expectation is with respect to the randomness in choosing $\mathbf{z}_1, \dots, \mathbf{z}_T$.

Limited Feedback (Bandits)

Proof.

Taking expectation of both sides with respect to the randomness in choosing $\mathbf{z}_t$:

$$\mathbb{E}\Big[\sum_{i=1}^{T}\langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t\rangle\Big] \leq B(\mathbf{u}) + \sum_{i=1}^{T}\mathbb{E}\big[||\mathbf{z}_t||_t^2\big]$$

By the law of total probability $\Big(\mathbf{v}_t = \mathbb{E}[\mathbf{z}_t|\mathbf{z}_{t-1}, \dots, \mathbf{z}_1] \in \partial f_t(\mathbf{w}_t)\Big)$:

$$\mathbb{E}\big[\sum_{i=1}^{T}\langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t\rangle\big] = \mathbb{E}\big[\sum_{i=1}^{T}\langle \mathbf{w}_t - \mathbf{u}, \mathbf{v}_t\rangle\big]$$

Limited Feedback (Bandits)

Proof.

Taking expectation of both sides with respect to the randomness in choosing $\mathbf{z}_t$:

$$\mathbb{E}\Big[\sum_{i=1}^{T}\langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t\rangle\Big] \leq B(\mathbf{u}) + \sum_{i=1}^{T}\mathbb{E}\big[||\mathbf{z}_t||_t^2\big]$$

By the law of total probability $\Big(\mathbf{v}_t = \mathbb{E}[\mathbf{z}_t|\mathbf{z}_{t-1},\ldots,\mathbf{z}_1] \in \partial f_t(\mathbf{w}_t)\Big)$:

$$\mathbb{E}\Big[\sum_{i=1}^{T}\langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t\rangle\Big] = \mathbb{E}\Big[\sum_{i=1}^{T}\langle \mathbf{w}_t - \mathbf{u}, \mathbf{v}_t\rangle\Big]$$

Due to the convexity we also know that:

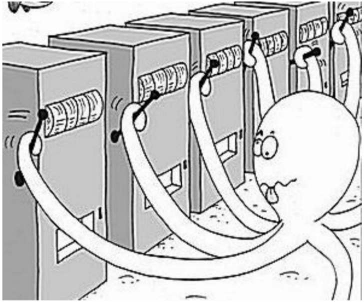
$$\langle \mathbf{w}_t - \mathbf{u}, \mathbf{v}_t\rangle \geq f_t(\mathbf{w}_t) - f_t(\mathbf{u})$$

Subsection 1

Multi-Armed Bandits

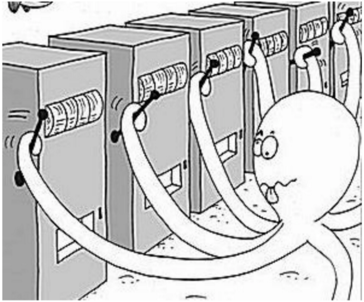
Multi-Armed Bandits (MAB)

A natural bandit version of Learning from Expert Advice (LEA):



Multi-Armed Bandits (MAB)

A natural bandit version of Learning from Expert Advice (LEA):



Exploration vs. Exploitation

Multi-Armed Bandit

- The vector $\mathbf{y}_t \in [0,1]^d$ associates a cost for each of the arms, but the learner only gets to see the cost of the arm it pulls.

Multi-Armed Bandit

- The vector $\mathbf{y}_t \in [0, 1]^d$ associates a cost for each of the arms, but the learner only gets to see the cost of the arm it pulls.
- The goal is to have low regret:

$$\mathbb{E}\Big[\sum_{t=1}^{T} \mathbf{y}_t[p_t]\Big] - \min_i \sum_{t=1}^{T} \mathbf{y}_t[i]$$

Multi-Armed Bandit

- Let S be the probability simplex.
- The learner picks an arm according to $\mathbb{P}[p_t = i] = \mathbf{w}_t[i]$ and therefore $f_t(\mathbf{w}) = \langle \mathbf{w}, \mathbf{y}_t \rangle$ is the expected cost of the chosen arm.
- To estimate the gradient:

$$\mathbf{z}_t[j] = \begin{cases} \frac{y_t[j]}{w_t[j]} & j = p_t \\ 0 & \text{else} \end{cases}$$

$$\mathbb{E}\left[\mathbf{z}_t^{(p_t)}[j] | \mathbf{z}_{t-1}, \ldots, z_1\right] = \sum_{i=1}^{d} \mathbb{P}[p_t = i] z_t^{(i)}[j] = w_t[j] \frac{y_t[j]}{w_t[j]} = y_t[j]$$

Multi-Armed Bandit

if we update $\mathbf{w}_t$ using the update rule of the normalized EG algorithm we saw before:

```
Multi-Armed Bandit Algorithm
```

parameter: $\eta \in (0,1)$
initialize: $\mathbf{w}_1 = (1/d, \ldots, 1/d)$
for $t = 1, 2, \ldots$
  choose $p_t \sim \mathbf{w}_t$ and pull the p_t'th arm
  receive cost of the arm $y_t[p_t] \in [0,1]$
  update
    $\tilde{w}[p_t] = w_t[p_t] e^{-\eta y_t[p_t]/w_t[p_t]}$
    for $i \neq p_t$, $\tilde{w}[i] = w_t[i]$
    $\forall i,\ w_{t+1}[i] = \frac{\tilde{w}[i]}{\sum_j \tilde{w}[j]}$

Multi-armed bandit

For the exponentiated gradient, we proved that:

$$\sum_{t=1}^{T} \langle w_t - \mathbf{u}, z_t \rangle \leq \frac{\log(d)}{\eta} + \eta \sum_{t=1}^{T} \sum_{i} \mathbf{w}_t[i] \mathbf{z}_t[i]^2$$

Multi-armed bandit

For the exponentiated gradient, we proved that:

$$\sum_{t=1}^{T} \langle w_t - \mathbf{u}, z_t \rangle \leq \frac{\log(d)}{\eta} + \eta \sum_{t=1}^{T} \sum_{i} \mathbf{w}_t[i] \mathbf{z}_t[i]^2$$

Thus,

$$\mathbb{E}\Big[\sum_{i=1}^{T} f_t(\mathbf{w}_t) - f_t(\mathbf{u}) \Big] \leq \frac{\log(d)}{\eta} + \eta \sum_{t=1}^{T} \sum_{i} \mathbb{E}[\mathbf{w}_t[i] \mathbf{z}_t[i]^2]$$

Multi-armed bandit

For the exponentiated gradient, we proved that:

$$\sum_{t=1}^{T} \langle w_t - \mathbf{u}, z_t \rangle \leq \frac{\log(d)}{\eta} + \eta \sum_{t=1}^{T} \sum_{i} \mathbf{w}_t[i] \mathbf{z}_t[i]^2$$

Thus,

$$\mathbb{E}\Big[\sum_{i=1}^{T} f_t(\mathbf{w}_t) - f_t(\mathbf{u})\Big] \leq \frac{\log(d)}{\eta} + \eta \sum_{t=1}^{T} \sum_{i} \mathbb{E}[\mathbf{w}_t[i] \mathbf{z}_t[i]^2]$$

The last term can be bounded as:

$$\begin{aligned}
\mathbb{E}\Big[\sum_{i} \mathbf{w}_t[\,i\,] \mathbf{z}_t^{(p_t)}[\,i\,]^2 \Big| \mathbf{z}_{t-1}, \ldots, \mathbf{z}_1\Big] &= \sum_{j} \mathbb{P}[\,p_t = j\,] \sum_{i} \mathbf{w}_t[\,i\,] \mathbf{z}_t^{(j)}[\,i\,]^2 \\
&= \sum_{j} \mathbf{w}_t[\,j\,] \mathbf{w}_t[\,j\,] \left(\frac{\mathbf{y}_t[\,j\,]}{\mathbf{w}_t[\,j\,]}\right)^2 \\
&= \sum_{j} \mathbf{y}_t[\,j\,]^2 \leq d
\end{aligned}$$

Multi-armed bandit

Corollary

The multi-armed bandit algorithm enjoys the bound

$$\mathbb{E}\left[\sum_{t=1}^{T} y_t[p_t]\right] \leq \min_i \sum_{t=1}^{T} y_t[i] + \frac{\log(d)}{\eta} + \eta d T.$$

In particular, $\eta = \sqrt{\frac{\log(d)}{dT}}$ gives the regret bound $2\sqrt{d \log(d) T}$.

Multi-armed bandit

Corollary

The multi-armed bandit algorithm enjoys the bound

$$\mathbb{E}\Big[\sum_{t=1}^{T} y_t[p_t]\Big] \leq \min_i \sum_{t=1}^{T} y_t[i] + \frac{\log(d)}{\eta} + \eta d T.$$

In particular, $\eta = \sqrt{\frac{\log(d)}{dT}}$ gives the regret bound $2\sqrt{d \log(d) T}$.

There exists a matching lower bound and $\sqrt{d \log(d)}$ is tight.

Subsection 2

Stochastic Bandits

Stochastic Bandits

- Each arm $i \in \{1, 2, \ldots, d\}$ is a probability distribution D_i.

Stochastic Bandits

- Each arm $i \in \{1, 2, \ldots, d\}$ is a probability distribution D_i.
- At time t, we select arm A_t and receive $\mathbf{g}_{t,A_t} \sim D_{A_t}$.

Stochastic Bandits

- Each arm $i \in \{1, 2, \ldots, d\}$ is a probability distribution D_i.
- At time t, we select arm A_t and receive $\mathbf{g}_{t,A_t} \sim D_{A_t}$.
- The Pseudo-Regret is defined as follows:

$$\mathrm{Regret}_T = \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{g}_{t,A_t}\right] - \min_i \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{g}_{t,i}\right]$$

Explore-Then-Commit Algorithm

The most basic algorithm:

Algorithm 10.4 Explore-Then-Commit Algorithm

Require: $T, m \in \mathbb{N}, 1 \leq m \leq \frac{T}{d}$

$S_{0,i} = 0, \hat{\mu}_{0,i} = 0, \ i = 1, \dots, d$

for $t = 1$ **to** T **do**

  Choose $A_t = \begin{cases} (t \mod d) + 1, & t \leq dm \\ \text{argmin}_i \ \hat{\mu}_{dm,i}, & t > dm \end{cases}$

  Observe g_{t,A_t} and pay it

  $S_{t,i} = S_{t-1,i} + \mathbf{1}[A_t = i]$

  $\hat{\mu}_{t,i} = \frac{1}{S_{t,i}} \sum_{j=1}^{t} g_{j,A_j} \mathbf{1}[A_j = i], \ i = 1, \dots, d$

end for

ETC: Analysis

- $S_{t,i} = \sum_{j=1}^{d} 1[A_t = i]$.
- $\Delta_i = \mu_i - \mu^*$.

Lemma

For any policy of selection of the arms,

$$\text{Regret}_T = \sum_{i=1}^{d} \mathbb{E}[S_{T,i}].\Delta_i \ .$$

ETC: Analysis

Proof.

$$\text{Regret}_T = \mathbb{E}\left[\sum_{t=1}^{T} g_{t,A_t}\right] - T\mu^\star = \mathbb{E}\left[\sum_{t=1}^{T}(g_{t,A_t} - \mu^\star)\right]$$

$$= \sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}\Big[\mathbf{1}[A_t = i](g_{t,i} - \mu^\star)\Big] = \sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}\Big[\mathbb{E}\big[\mathbf{1}[A_t = i](g_{t,i} - \mu^\star)|A_t\big]\Big]$$

$$= \sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}\Big[\mathbf{1}[A_t = i]\mathbb{E}\big[g_{t,i} - \mu^\star|A_t\big]\Big]$$

$$= \sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}\Big[\mathbf{1}[A_t = i](\mu_{A_t} - \mu^\star)\Big] = \sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}\big[\mathbf{1}[A_t = i]\big]\underbrace{(\mu_i - \mu^\star)}_{\Delta_i} .$$

ETC: Analysis

Proof.

$$\mathrm{Regret}_T = \mathbb{E}\left[\sum_{t=1}^{T} g_{t,A_t}\right] - T\mu^\star = \mathbb{E}\left[\sum_{t=1}^{T}(g_{t,A_t} - \mu^\star)\right]$$

$$= \sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}\Big[\mathbf{1}[A_t = i](g_{t,i} - \mu^\star)\Big] = \sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}\Big[\mathbb{E}\big[\mathbf{1}[A_t = i](g_{t,i} - \mu^\star)|A_t\big]\Big]$$

$$= \sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}\Big[\mathbf{1}[A_t = i]\mathbb{E}\big[g_{t,i} - \mu^\star|A_t\big]\Big]$$

$$= \sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}\Big[\mathbf{1}[A_t = i](\mu_{A_t} - \mu^\star)\Big] = \sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}\big[\mathbf{1}[A_t = i]\big]\underbrace{(\mu_i - \mu^\star)}_{\Delta_i}.$$

□

In order to have a small regret we have to select the suboptimal arms less often then the best one.

ETC: Analysis

Theorem

Assume that the losses of the arms minus their expectations are 1-subgaussian and $1 \leq m \leq T/d$. Then, ETC guarantees a regret of

$$\mathrm{Regret}_T \leq m \sum_{i=1}^{d} \Delta_i + (T - md) \sum_{i=1}^{d} \Delta_i \exp\left(-\frac{m\Delta_i^2}{4}\right) .$$

ETC: Proof

Proof.

Let's assume without loss of generality that the optimal arm is the first one. So, for $i \neq 1$, we have

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}[A_t = i]\right] &= m + (T - md)\mathbb{P}\left[\hat{\mu}_{md,i} \leq \min_{j \neq i} \ \hat{\mu}_{md,j}\right] \\ &\leq m + (T - md)\mathbb{P}\left[\hat{\mu}_{md,i} \leq \hat{\mu}_{md,1}\right] \\ &= m + (T - md)\mathbb{P}\left[\hat{\mu}_{md,1} - \mu_1 - (\hat{\mu}_{md,i} - \mu_i) \geq \Delta_i\right] \ .\end{aligned}$$

ETC: Proof

Proof.

Let's assume without loss of generality that the optimal arm is the first one. So, for $i \neq 1$, we have

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}[A_t = i]\right] &= m + (T - md)\mathbb{P}\left[\hat{\mu}_{md,i} \leq \min_{j \neq i} \hat{\mu}_{md,j}\right] \\ &\leq m + (T - md)\mathbb{P}\left[\hat{\mu}_{md,i} \leq \hat{\mu}_{md,1}\right] \\ &= m + (T - md)\mathbb{P}\left[\hat{\mu}_{md,1} - \mu_1 - (\hat{\mu}_{md,i} - \mu_i) \geq \Delta_i\right] .\end{aligned}$$

We also know that $\hat{\mu}_{md,1} - \mu_1 - (\hat{\mu}_{md,i} - \mu_i)$ is $\sqrt{2/m}$–subgaussian. Hence,

$$\mathbb{P}\left[\hat{\mu}_{md,1} - \mu_1 - (\hat{\mu}_{md,i} - \mu_i) \geq \Delta_i\right] \leq \exp\left(-\frac{m\Delta_i^2}{4}\right) .$$

□

ETC: Discussion

1. The main drawback of this algorithm is that its optimal tuning depends on the gaps.

ETC: Discussion

1. The main drawback of this algorithm is that its optimal tuning depends on the gaps.
2. The ETC algorithm has the disadvantage of requiring the knowledge of the gaps to tune the exploration phase.

ETC: Discussion

1. The main drawback of this algorithm is that its optimal tuning depends on the gaps.
2. The ETC algorithm has the disadvantage of requiring the knowledge of the gaps to tune the exploration phase.
3. It solves the exploration vs. exploitation trade-off in a bad way!

ETC: Discussion

1. The main drawback of this algorithm is that its optimal tuning depends on the gaps.
2. The ETC algorithm has the disadvantage of requiring the knowledge of the gaps to tune the exploration phase.
3. It solves the exploration vs. exploitation trade-off in a bad way!
4. It would be better to have an algorithm that smoothly transition from one phase into the other in a data-dependent way.

Upper Confidence Bound (UCB)

Algorithm 10.5 Upper Confidence Bound Algorithm

Require: $\alpha > 2, T \in \mathbb{N}$

$S_{0,i} = 0, \hat{\mu}_{0,i} = 0,\ i = 1, \ldots, d$

for $t = 1$ **to** T **do**

  Choose $A_t = \text{argmin}_{i=1,\ldots,d} \begin{cases} \mu_{t-1,i} - \sqrt{\frac{2\alpha \ln t}{S_{t-1,i}}}, & \text{if } S_{t-1,i} \neq 0 \\ -\infty, & \text{otherwise} \end{cases}$

  Observe g_{t,A_t} and pay it

  $S_{t,i} = S_{t-1,i} + \mathbf{1}[A_t = i]$

  $\hat{\mu}_{t,i} = \frac{1}{S_{t,i}} \sum_{j=1}^{t} g_{t,A_t} \mathbf{1}[A_t = i],\ i = 1, \ldots, d$

end for

Upper Confidence Bound (UCB)

Algorithm 10.5 Upper Confidence Bound Algorithm

Require: $\alpha > 2, T \in \mathbb{N}$

1: $S_{0,i} = 0, \hat{\mu}_{0,i} = 0,\ i = 1, \ldots, d$
2: **for** $t = 1$ **to** T **do**
3: Choose $A_t = \text{argmin}_{i=1,\ldots,d} \begin{cases} \mu_{t-1,i} - \sqrt{\frac{2\alpha \ln t}{S_{t-1,i}}}, & \text{if } S_{t-1,i} \neq 0 \\ -\infty, & \text{otherwise} \end{cases}$
4: Observe g_{t,A_t} and pay it
5: $S_{t,i} = S_{t-1,i} + \mathbf{1}[A_t = i]$
6: $\hat{\mu}_{t,i} = \frac{1}{S_{t,i}} \sum_{j=1}^{t} g_{t,A_t} \mathbf{1}[A_t = i],\ i = 1, \ldots, d$
7: **end for**

UCB works keeping an estimate of the expected loss of each arm and also a confidence interval at a certain probability.

UCB: Analysis

Theorem

Assume that the rewards of the arms are 1-subgaussian and let $\alpha > 2$. Then, UCB guarantees a regret of

$$\mathrm{Regret}_T \leq \frac{\alpha}{\alpha - 2} \sum_{i=1}^{d} \Delta_i + \sum_{i:\Delta_i>0} \frac{8\alpha \ln T}{\Delta_i} .$$

UCB Regret Proof

- Let $i = 1$ be the optimal arm.
- Note that $\mathrm{Regret}_T = \sum_{i=1}^{d} \Delta_i \mathbb{E}[S_{T,i}]$.
- For arm non optimal arm i, we want to prove that

$$\mathbb{E}[S_{T,i}] \leq \frac{8\alpha \ln T}{\Delta_i^2} + \frac{\alpha}{\alpha - 2}$$

.

- The proof is based on the fact that once I have sampled an arm enough times, the probability to take a suboptimal arm is small.
- Let $t^\star$ the biggest time index such that $S_{t^\star-1,i} \leq \frac{8\alpha \ln T}{\Delta_i^2}$. For $t > t^\star$, we have

$$S_{t-1,i} > \frac{8\alpha \ln T}{\Delta_i^2} \,. \tag{18}$$

UCB Regret Proof

- Consider $t > t^\star$ and such that $A_t = i \neq 1$, then we claim that at least one of the two following equations must be true:

$$\hat{\mu}_{t-1,1} - \sqrt{\frac{2\alpha \ln t}{S_{t-1,1}}} \geq \mu_1, \tag{19}$$

$$\hat{\mu}_{t-1,i} + \sqrt{\frac{2\alpha \ln t}{S_{t-1,i}}} < \mu_i \ . \tag{20}$$

UCB Regret Proof

Let's prove the claim: *if both the inequalities above are false*, $t > t^\star$, and $A_t = i$, we have

$$\begin{aligned}
\hat{\mu}_{t-1,1} - \sqrt{\frac{2\alpha \ln t}{S_{t-1,1}}} &< \mu_1 && \text{((19) false)} \\
&= \mu_i - \Delta_i \\
&< \mu_i - 2\sqrt{\frac{2\alpha \ln T}{S_{t-1,i}}} && \text{(for (18))} \\
&\leq \hat{\mu}_{t-1,i} - \sqrt{\frac{2\alpha \ln t}{S_{t-1,i}}} && \text{((20) false),}
\end{aligned}$$

that, by the selection strategy of the algorithm, would imply $A_t \neq i$.

UCB Regret Proof

Note that $S_{t^\star,i} \leq \frac{8\alpha \ln T}{\Delta_i^2} + 1$. Hence, we have

$$\begin{aligned}
\mathbb{E}\left[S_{T,i}\right] &= \mathbb{E}[S_{t^\star,i}] + \sum_{t=t^\star+1}^{T} \mathbb{E}[\mathbf{1}[A_t = i, (19) \text{ or } (20) \text{ true}]] \\
&\leq \frac{8\alpha \ln T}{\Delta_i^2} + 1 + \sum_{t=t^\star+1}^{T} \mathbb{E}[\mathbf{1}[(19) \text{ or } (20) \text{ true}]] \\
&\leq \frac{8\alpha \ln T}{\Delta_i^2} + 1 + \sum_{t=t^\star+1}^{T} \left(\Pr[(19) \text{ true}] + \Pr[(20) \text{ true}]\right) .
\end{aligned}$$

UCB Regret Proof

Now, we upper bound the probabilities in the sum. First, note that, given that the losses on the arms are i.i.d., we have

$$\left\{\hat{\mu}_{t-1,1} - \sqrt{\frac{2\alpha \ln t}{S_{t-1,1}}} \geq \mu_1\right\} \subset \left\{\max_{s=1,\dots,t-1} \frac{1}{s}\sum_{j=1}^{s} g_{j,1} - \sqrt{\frac{2\alpha \ln t}{s}} \geq \mu_1\right\}$$
$$= \bigcup_{s=1}^{t-1} \left\{\frac{1}{s}\sum_{j=1}^{s} g_{j,1} - \sqrt{\frac{2\alpha \ln t}{s}} \geq \mu_1\right\}$$

Hence, we have

$$\Pr[(19) \text{ true}] \leq \sum_{s=1}^{t-1} \Pr\left[\frac{1}{s}\sum_{j=1}^{s} g_{j,1} - \sqrt{\frac{2\alpha \ln t}{s}} \geq \mu_1\right] \qquad \text{(union bound)}$$

$$\leq \sum_{s=1}^{t-1} t^{-\alpha} = (t-1)t^{-\alpha} \; .$$

Given that the same bound holds for Pr[(20) true], we have

$$\begin{aligned}
\mathbb{E}\left[S_{T,i}\right] &\leq \frac{8\alpha \ln T}{\Delta_i^2} + 1 + \sum_{t=1}^{\infty} 2(t-1)t^{-\alpha} \\
&\leq \frac{8\alpha \ln T}{\Delta_i^2} + 1 + \sum_{t=2}^{\infty} 2t^{1-\alpha} \\
&\leq \frac{8\alpha \ln T}{\Delta_i^2} + 1 + 2\int_1^{\infty} x^{1-\alpha} \\
&= \frac{8\alpha \ln T}{\Delta_i^2} + \frac{\alpha}{\alpha-2}\ .
\end{aligned}$$

Using the decomposition of the regret we proved last time,

$$\mathrm{Regret}_T = \sum_{i=1}^{d} \Delta_i \mathbb{E}[S_{T,i}],$$

we have the stated bound.

Section 8

New Trends

Subsection 1

Bandits

New Trends

Exploration

- Now suppose this problem:
  1. Strategy chooses $a_t \in \mathcal{A} \subset \mathbb{R}^d$.
  2. Adversary chooses **linear** loss $l_t \in \mathcal{L} \subseteq [-1, 1]^{\mathcal{A}}$
  3. Strategy sees loss $l_t(a_t) = l_t^T a_t$

  We aim to minimize pseudo-regret:

$$R_n = \mathbb{E} \sum_{t=1}^{T} l_t(a_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} l_t(a) \tag{21}$$

New Trends

Exploration

- Now suppose this problem:
  1. Strategy chooses $a_t \in \mathcal{A} \subset \mathbb{R}^d$.
  2. Adversary chooses **linear** loss $l_t \in \mathcal{L} \subseteq [-1, 1]^{\mathcal{A}}$
  3. Strategy sees loss $l_t(a_t) = l_t^T a_t$

  We aim to minimize pseudo-regret:

$$R_n = \mathbb{E} \sum_{t=1}^{T} l_t(a_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} l_t(a) \tag{21}$$

problem falls to how to choose a_t's. And how to estimate $l_t^T a$

New Trends

Exploration

Given $\mathcal{A}$, distribution μ on $\mathcal{A}$, mixing coefficient $\gamma > 0$, learning rate $\eta > 0$,

set q_1 uniform on $\mathcal{A}$.

for $t = 1, 2, \ldots, n$,

1. $p_t = (1 - \gamma) q_t + \gamma \mu$
2. choose $a_t \sim p_t$
3. observe $\ell_t^T a_t$
4. update $q_{t+1}(a) \propto q_t(a) \exp(-\eta \tilde{\ell}_t^T a)$,

$$\text{where} \qquad \tilde{\ell}_t = \Sigma_t^{-1} a_t a_t^T \ell_t,$$

$$\Sigma_t = \mathbb{E}_{a \sim p_t} a a^T.$$

New trends

Exploration

- Strategy observes $a_t^T l_t$ and a_t, so it can compute:

$$\tilde{l}_t = \Sigma_t^{-1} a_t (a_t^T l_t)$$

- $\tilde{l}_t$ is unbiased:

$$\mathbb{E}[\tilde{l}_t | \mathcal{F}_{t-1}] = (\mathbb{E}_{a \sim p_t} a a^T)^{-1} (\mathbb{E}_{a \sim p_t} a a^T) l_t = l_t.$$

New trends

Exploration

- Strategy observes $a_t^T l_t$ and a_t, so it can compute:

$$\tilde{l}_t = \Sigma_t^{-1} a_t (a_t^T l_t)$$

- $\tilde{l}_t$ is unbiased:

$$\mathbb{E}[\tilde{l}_t | \mathcal{F}_{t-1}] = (\mathbb{E}_{a \sim p_t} a a^T)^{-1} (\mathbb{E}_{a \sim p_t} a a^T) l_t = l_t.$$

- Therefore:

$$\mathbb{E}[l_t^T a] = E[\tilde{l}_t{}^T a] \quad \forall a$$

New trends

Exploration

- Strategy observes $a_t^T l_t$ and a_t, so it can compute:

$$\tilde{l}_t = \Sigma_t^{-1} a_t (a_t^T l_t)$$

- $\tilde{l}_t$ is unbiased:

$$\mathbb{E}[\tilde{l}_t | \mathcal{F}_{t-1}] = (\mathbb{E}_{a \sim p_t} a a^T)^{-1} (\mathbb{E}_{a \sim p_t} a a^T) l_t = l_t.$$

- Therefore:

$$\mathbb{E}[l_t^T a] = E[\tilde{l}_t^{\,T} a] \quad \forall a$$

- and:

$$\mathbb{E}[l_t^T a_t] = \mathbb{E}[\sum_{a \in \mathcal{A}} p_t(a) \mathbb{E}[\tilde{l}_t | \mathcal{F}_{t-1}]^T a] = \mathbb{E}[\sum_{a \in \mathcal{A}} p_t(a) \tilde{l}_t^{\,T} a]$$

New Trends

Exploration

- So we can write the strategy's expected cumulative loss as:

$$\mathbb{E}\sum_{t=1}^{n} l_t^T a_t = \mathbb{E}\sum_{t=1}^{n}\sum_{a\in\mathcal{A}} p_t(a)\tilde{l}_t^{\,T} a.$$

New Trends

Exploration

- So we can write the strategy's expected cumulative loss as:

$$\mathbb{E}\sum_{t=1}^{n} l_t^T a_t = \mathbb{E}\sum_{t=1}^{n}\sum_{a\in\mathcal{A}} p_t(a)\tilde{l}_t^{\,T} a.$$

- Which can be written as:

$$\sum_{t=1}^{n}\sum_{a\in\mathcal{A}} p_t(a)\tilde{l}_t^{\,T} a = \sum_{t=1}^{n}\sum_{a\in\mathcal{A}}((1-\gamma)q_t(a)+\gamma\mu(a))\tilde{l}_t^{\,T} a$$

$$= (1-\gamma)(\sum_{t=1}^{n}\sum_{a\in\mathcal{A}} q_t(a)\tilde{l}_t^{\,T} a) + \gamma(\sum_{t=1}^{n}\sum_{a\in\mathcal{A}} \mu(a)\tilde{l}_t^{\,T} a)$$

New Trends

Exploration

- Note that the ditrbution changes as well:

$$\begin{aligned}\mathbb{E}\sum_{t=1}^{n}(l(\tilde{a}_t, z_t) - l(a, z_t)) &= \mathbb{E}\sum_{t=1}^{n}(l(\tilde{a}_t, z_t) - l(a_t, z_t) + l(a_t, z_t) - l(a, z_t)) \\ &\leq G\mathbb{E}\sum_{t=1}^{n}\|a_t - \tilde{a}_t\| + \mathbb{E}\sum_{t=1}^{n}\nabla l(a_t, z_t)^T(a_t - a) \\ &= G\mathbb{E}\sum_{t=1}^{n}\|a_t - \tilde{a}_t\| + \mathbb{E}\sum_{t=1}^{n}\tilde{l}_t^{\,T}(a_t - a)\end{aligned}$$

- In our case, if we assume $\|l_t\| \leq 1$:

$$\mathbb{E}\sum_{t=1}^{n}(l(\tilde{a}_t, z_t) - l(a, z_t)) \leq 2\gamma n + \mathbb{E}\sum_{t=1}^{n}\tilde{l}_t^{\,T}(a_t - a)$$

New Trends

Exploration

- Recall this theorem:

Theorem

Assume that the normalized EG algorithm is run on a sequence of linear loss functions such that for all t, i *we have* $\eta z_t[i] \geq -1$. *Then:*

$$\sum_{t=1}^{T} \langle w_t - \mathbf{u}, z_t \rangle \leq \frac{\log(d)}{\eta} + \eta \sum_{t=1}^{T} \sum_{i} w_t[i] z_t[i]^2$$

New Trends

Exploration

- Recall this theorem:

Theorem

Assume that the normalized EG algorithm is run on a sequence of linear loss functions such that for all t, i we have $\eta z_t[i] \geq -1$. Then:

$$\sum_{t=1}^{T} \langle w_t - \mathbf{u}, z_t \rangle \leq \frac{\log(d)}{\eta} + \eta \sum_{t=1}^{T} \sum_{i} w_t[i] z_t[i]^2$$

- $$\tilde{R}_n \leq 2\gamma n + (1-\gamma)\left(\frac{\log(N)}{\eta} + \eta \mathbb{E} \sum_{t=1}^{T} \sum_{a \in \mathcal{A}} q_t(a) ({\tilde{l}_t}^T a)^2\right)$$
$$\leq 2\gamma n + \frac{\log(N)}{\eta} + \eta \sum_{t=1}^{T} \sum_{a \in \mathcal{A}} p_t(a) ({\tilde{l}_t}^T a)^2$$

New Trends

Exploration

- $$\sum_{a\in\mathcal{A}} p_t(a)\langle\tilde{l}_t, a\rangle^2 = \sum_{a\in\mathcal{A}} p_t(a)\langle\tilde{l}_t, (aa^T)\tilde{l}_t\rangle$$
  $$= \langle\tilde{l}_t, \Sigma_t\tilde{l}_t\rangle$$
  $$= \langle a_t, l_t\rangle^2\langle\Sigma_t^{-1}a_t, \Sigma_t\Sigma_t^{-1}a_t\rangle$$

  where in last equality, we've used: $\tilde{l}_t = \Sigma_t^{-1}a_t\langle a_t, l_t\rangle$

New Trends

Exploration

- $$\sum_{a\in\mathcal{A}} p_t(a)\langle\tilde{l}_t, a\rangle^2 = \sum_{a\in\mathcal{A}} p_t(a)\langle\tilde{l}_t, (aa^T)\tilde{l}_t\rangle$$
  $$= \langle\tilde{l}_t, \Sigma_t\tilde{l}_t\rangle$$
  $$= \langle a_t, l_t\rangle^2\langle\Sigma_t^{-1}a_t, \Sigma_t\Sigma_t^{-1}a_t\rangle$$

  where in last equality, we've used: $\tilde{l}_t = \Sigma_t^{-1}a_t\langle a_t, l_t\rangle$
- if we assume $\|a\| \leq 1$, we will have:

  $$\leq \langle\Sigma_t^{-1}a_t, a_t\rangle$$

New Trends

Exploration

- $$\sum_{a\in\mathcal{A}} p_t(a)\langle\tilde{l}_t, a\rangle^2 = \sum_{a\in\mathcal{A}} p_t(a)\langle\tilde{l}_t, (aa^T)\tilde{l}_t\rangle$$
  $$= \langle\tilde{l}_t, \Sigma_t\tilde{l}_t\rangle$$
  $$= \langle a_t, l_t\rangle^2\langle\Sigma_t^{-1}a_t, \Sigma_t\Sigma_t^{-1}a_t\rangle$$

  where in last equality, we've used: $\tilde{l}_t = \Sigma_t^{-1}a_t\langle a_t, l_t\rangle$
- if we assume $\|a\| \leq 1$, we will have:
  $$\leq \langle\Sigma_t^{-1}a_t, a_t\rangle$$
- $$\mathbb{E}\langle\Sigma_t^{-1}a_t, a_t\rangle = d!$$

New Trends

Exploration

- We now turn to $\langle a, \tilde{l}_t \rangle$ (Why?)

$$\begin{aligned}\langle a, \tilde{l}_t \rangle &= \langle a_t, l_t \rangle \langle a_t, \Sigma_t^{-1} a_t \rangle \\ &= \langle a_t, \Sigma_t^{-1} a_t \rangle \\ &\leq \frac{1}{min_{1 \leq i \leq d} \lambda_i}\end{aligned}$$

New Trends

Exploration

- We now turn to $\langle a, \tilde{l}_t \rangle$ (Why?)

$$\begin{aligned}\langle a, \tilde{l}_t \rangle &= \langle a_t, l_t \rangle \langle a_t, \Sigma_t^{-1} a_t \rangle \\ &= \langle a_t, \Sigma_t^{-1} a_t \rangle \\ &\leq \frac{1}{min_{1 \leq i \leq d} \lambda_i}\end{aligned}$$

- We must have:

$$\eta z_t[i] \geq -1$$

to guarantee normalized EG algorithm.

New Trends

Exploration

Theorem

Assume that $\mathcal{L} \subset [-1,1]^{\mathcal{A}}$*, if:*

$$\underset{a,b\in\mathcal{A}}{a^t \Sigma_t^{-1} b} \leq \frac{c_d}{\gamma}$$

setting $\gamma = c_d\eta, \eta = \sqrt{\frac{\log(N)}{n(d+c_d)}}$*, wo will have:*

$$\tilde{R}_n \leq 2\sqrt{n(c_d + d)\log(N)} \tag{22}$$

New Trends

Exploration

Theorem

Assume that $\mathcal{L} \subset [-1,1]^{\mathcal{A}}$, *if:*

$$\underset{a,b \in \mathcal{A}}{a^t \Sigma_t^{-1} b} \leq \frac{c_d}{\gamma}$$

setting $\gamma = c_d \eta, \eta = \sqrt{\frac{\log(N)}{n(d+c_d)}}$, *wo will have:*

$$\tilde{R}_n \leq 2\sqrt{n(c_d + d)\log(N)} \tag{22}$$

Getting back to exploration term, what should we set for $\mu(a)$ to guarantee above bound?

New Trends

Exploration

(Dani, Hayes, Kakade, 2008):
For μ uniform over *barycentric spanner*,

$$\overline{R}_n = O\left(d\sqrt{n \log |\mathcal{A}|}\right) = \tilde{O}\left(d^{3/2}\sqrt{n}\right).$$

(Cesa-Bianchi and Lugosi, 2009):
For several combinatorial problems, $\mathcal{A} \subseteq \{0,1\}^d$, μ uniform over $\mathcal{A}$ gives

$$\frac{\sup_{a \in \mathcal{A}} \|a\|_2^2}{\lambda_{\min}\left(\mathbb{E}_{a \sim \mu}[aa^T]\right)} = O(d),$$

so

$$\overline{R}_n = O\left(\sqrt{dn \log |\mathcal{A}|}\right) = \tilde{O}\left(d\sqrt{n}\right).$$

(Bubeck, Cesa-Bianchi and Kakade, 2009): *John's Theorem*:
$\tilde{O}\left(d\sqrt{n}\right)$.

New Trends

barycentric spanner

- Suppose that $\mathcal{A} \subset \mathbb{R}^d$ spans $\mathbb{R}^d$, a barycentric spanner of $\mathcal{A}$ is a set $b_1, \ldots, b_d$ that spans $\mathbb{R}^d$ and satisfies:
  for all $a \in \mathcal{A}$ there is an $\alpha \in [-1, 1]^d$ such that $a = B\alpha$, where $B = (b_1, \ldots, b_d)$.
  it can be shown that:

New Trends

barycentric spanner

- Suppose that $\mathcal{A} \subset \mathbb{R}^d$ spans $\mathbb{R}^d$, a barycentric spanner of $\mathcal{A}$ is a set $b_1, \ldots, b_d$ that spans $\mathbb{R}^d$ and satisfies:
  for all $a \in \mathcal{A}$ there is an $\alpha \in [-1, 1]^d$ such that $a = B\alpha$, where $B = (b_1, \ldots, b_d)$.
  it can be shown that:
  Every compact $\mathcal{A}$ has a barycentric spanner.

New Trends

barycentric spanner

- Suppose that $\mathcal{A} \subset \mathbb{R}^d$ spans $\mathbb{R}^d$, a barycentric spanner of $\mathcal{A}$ is a set $b_1, \ldots, b_d$ that spans $\mathbb{R}^d$ and satisfies:
  for all $a \in \mathcal{A}$ there is an $\alpha \in [-1, 1]^d$ such that $a = B\alpha$, where $B = (b_1, \ldots, b_d)$.
  it can be shown that:
  Every compact $\mathcal{A}$ has a barycentric spanner.
  If linear functions can be efficiently optimized over A, then there is an efficient algorithm for finding an approximate barycentric spanner.
  That is: $|\alpha_i| < 1 + \delta$ then it needs $O(d^2 \log(d)/\delta)$

New Trends

barycentric spanner

Lemma

If $b_1, \ldots, b_d \subset A$ maximizes $det(B)$, then it is a barycentric spanner.

New Trends

barycentric spanner

Lemma

If $b_1, \ldots, b_d \subset A$ maximizes $det(B)$, then it is a barycentric spanner.

Proof.

For $a = B\alpha$:

$$|det(B)| \geq |det(a, b_2, \ldots, b_d)|$$
$$= |\sum_i \alpha_i det(b_i, b_2, \ldots, b_d)|$$
$$= |\alpha_1||det(B)|$$

□

New Trends

barycentric spanner

Theorem: For $\mathcal{A} \subseteq [-1,1]^d$ and μ uniform on a barycentric spanner of $\mathcal{A}$,

$$\sup_{a,b \in \mathcal{A}} a^T \Sigma_t^{-1} b \leq \frac{d^2}{\gamma}$$

(that is, $c_d \leq d^2$). Hence,

$$\overline{R}_n \leq 2d\sqrt{2n \log |\mathcal{A}|}.$$

Subsection 2

Parameter-Free Online Learning

New Trends

Parameter-Free Online Learning

- Using OGD with 1-Lipschitz losses and learning rate $\eta = \frac{\alpha}{T}$, we arrive at the following Regret bound:

$$\mathrm{Regret}_T(\mathbf{u}) \leq \frac{||\mathbf{u}||_2^2}{2\eta} + \frac{\eta T}{2} = \frac{1}{2}\sqrt{T}\Big(\frac{||\mathbf{u}||_2^2}{\alpha} + \alpha\Big) \tag{23}$$

- To get the best bound, we need to set $\alpha = ||\mathbf{u}||_2$.

New Trends

Parameter-Free Online Learning

- Using OGD with 1-Lipschitz losses and learning rate $\eta = \frac{\alpha}{T}$, we arrive at the following Regret bound:

$$\text{Regret}_T(\mathbf{u}) \leq \frac{||\mathbf{u}||_2^2}{2\eta} + \frac{\eta T}{2} = \frac{1}{2}\sqrt{T}\Big(\frac{||\mathbf{u}||_2^2}{\alpha} + \alpha\Big) \quad (23)$$

- To get the best bound, we need to set $\alpha = ||\mathbf{u}||_2$.
- **Goal**: Design OCO algorithms that will enjoy the optimal regret and will not require any parameter.
  - Doubling Trick — Sub-optimal.
  - **Coin-Betting**

New Trends

Parameter-Free Online Learning: Coin Betting

Imagine the following repeated game to maximize $\mathrm{Wealth_T}$:

- Set initial weight to ϵ: $\mathrm{Wealth}_0 = \epsilon$.
- In each round $t = 1, \ldots, T$:
  - You bet $x_t = \beta_t \mathrm{Wealth}_t$ where $|\beta_t| \leq 1$ on side on coin $\mathrm{sign}(\beta_t)$.
  - The adversary reveals coin $c_t \in \{-1, 1\}$.
  - $\mathrm{Wealth}_t = \mathrm{Wealth}_{t-1} + c_t x_t = (1 + \beta_t c_t)\mathrm{Wealth}_{t-1}$
- This is a special instance of OCO, and we can have algorithms guaranteeing high $\mathrm{Wealth_T}$.
  - **KT Betting**: $\beta_t = \frac{\sum_{i=1}^{t-1} c_i}{t}$.
  - Guarantee:

$$\ln(\mathrm{Wealth}_T) \geq \sum_{t=1}^{T} \frac{\left(\sum_{i=1}^{T} c_t\right)^2}{4T} - \frac{1}{2}\log(T)$$

New Trends

Parameter-Free Online Learning: Coin Betting

Theorem

Let ϕ be a proper closed convex function and let ϕ^* be its Fenchel conjugate. If an algorithm that generates $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ can guarantee

$$\forall \mathbf{g_1}, \mathbf{g_2}, \ldots, \mathbf{g_T} \in \mathbb{R}^d : \quad \epsilon - \sum_{t=1}^{T} \langle \mathbf{x}_t, \mathbf{g}_t \rangle \geq \epsilon - \sum_{t=1}^{T} \phi\Big(- \sum_{t=1}^{T} \mathbf{g}_t \Big),$$

Then it guarantees

$$\forall \mathbf{u} \in \mathbb{R}^d, \quad \sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{x}_t - \mathbf{u} \rangle \leq \phi^*(\mathbf{u}) + \epsilon$$

New Trends

Parameter-Free Online Learning: Coin Betting

- Assumption:
  $\forall \mathbf{g_1}, \mathbf{g_2}, \ldots, \mathbf{g_T} \in \mathbb{R}^d: \quad \epsilon - \sum_{t=1}^{T} \langle \mathbf{x}_t, \mathbf{g}_t \rangle \geq \epsilon - \sum_{t=1}^{T} \phi\big(- \sum_{t=1}^{T} \mathbf{g}_t \big)$

New Trends

Parameter-Free Online Learning: Coin Betting

- Assumption:
  $\forall \mathbf{g_1}, \mathbf{g_2}, \ldots, \mathbf{g_T} \in \mathbb{R}^d : \quad \epsilon - \sum_{t=1}^{T} \langle \mathbf{x}_t, \mathbf{g}_t \rangle \geq \epsilon - \sum_{t=1}^{T} \phi\big(- \sum_{t=1}^{T} \mathbf{g}_t \big)$

- The Regret can be bounded as follows:

$$\sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{x}_t - \mathbf{u} \rangle \leq - \sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{u} \rangle - \phi\big(- \sum_{t=1}^{T} \mathbf{g}_t \big) + \epsilon$$

$$\leq \sup_{\theta \in \mathbb{R}^d} \langle \theta, \mathbf{u} \rangle - \phi(\theta) + \epsilon = \phi^*(\mathbf{u}) + \epsilon$$

New Trends

Parameter-Free Online Learning: Coin Betting

- The regret guarantee of KT used a 1d OLO algorithm is upper bounded by

$$\text{Regret}_T(u) = \sum_{t=1}^{T} \ell_t(x_t) - \sum_{t=1}^{T} \ell_t(u) \leq |u| \sqrt{4T \ln\left(\frac{\sqrt{2}|u|KT}{\epsilon} + 1\right)} + \epsilon,\ \forall u \in \mathbb{R},$$

- To better appreciate this regret, compare this bound to the one of OMD with learning rate $\eta = \frac{\alpha}{\sqrt{T}}$:

$$\text{Regret}_T(u) = \sum_{t=1}^{T} \ell_t(x_t) - \sum_{t=1}^{T} \ell_t(u) \leq \frac{1}{2}\left(\frac{u^2}{\alpha} + \alpha\right)\sqrt{T},\ \forall u \in \mathbb{R}\ .$$

New Trends

Parameter-Free Online Learning: Coin Betting

- The regret guarantee of KT used a 1d OLO algorithm is upper bounded by

$$\text{Regret}_T(u) = \sum_{t=1}^{T} \ell_t(x_t) - \sum_{t=1}^{T} \ell_t(u) \leq |u| \sqrt{4T \ln\left(\frac{\sqrt{2}|u|KT}{\epsilon} + 1\right)} + \epsilon,\ \forall u \in \mathbb{R},$$

- To better appreciate this regret, compare this bound to the one of OMD with learning rate $\eta = \frac{\alpha}{\sqrt{T}}$:

$$\text{Regret}_T(u) = \sum_{t=1}^{T} \ell_t(x_t) - \sum_{t=1}^{T} \ell_t(u) \leq \frac{1}{2}\left(\frac{u^2}{\alpha} + \alpha\right)\sqrt{T},\ \forall u \in \mathbb{R}\ .$$

- How to generalize to the general setting? Magnitude and Direction Decomposition

New Trends

Parameter-free in Any Norm

How should we convert the 1D algorithm to the general case?

New Trends

Parameter-free in Any Norm

How should we convert the 1D algorithm to the general case?

Algorithm 9.4 Learning Magnitude and Direction Separately

Require: 1d Online learning algorithm $\mathcal{A}_{1\text{d}}$, Online learning algorithm $\mathcal{A}_{\mathcal{B}}$ with feasible set equal to the unit ball $B \subset \mathbb{R}^d$ w.r.t. $\|\cdot\|$

for $t = 1$ **to** T **do**
- Get point $z_t \in \mathbb{R}$ from $\mathcal{A}_{1\text{d}}$
- Get point $\tilde{\boldsymbol{x}}_t \in B$ from $\mathcal{A}_B$
- Play $\boldsymbol{x}_t = z_t \tilde{\boldsymbol{x}}_t \in \mathbb{R}^d$
- Receive $\ell_t : \mathbb{R}^d \to (-\infty, +\infty]$ and pay $\ell_t(\boldsymbol{x}_t)$
- Set $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$
- Set $s_t = \langle \boldsymbol{g}_t, \tilde{\boldsymbol{x}}_t \rangle$
- Send $\ell_t^{\mathcal{A}_{1\text{d}}}(x) = s_t x$ as the t-th linear loss to $\mathcal{A}_{1\text{d}}$
- Send $\ell_t^{\mathcal{A}_B}(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle$ as the t-th linear loss to $\mathcal{A}_B$

end for

New Trends

Parameter-free in Any Norm

Theorem

$$\text{Regret}_T(\mathbf{u}) \leq \sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{x}_t - \mathbf{u} \rangle = \text{Regret}_T^{\mathcal{A}_{1d}}(\|\mathbf{u}\|) + \|\mathbf{u}\| \text{Regret}_T^{\mathcal{A}_B}\left(\frac{\mathbf{u}}{\|\mathbf{u}\|}\right) .$$

Further, the subgradients s_t sent to $\mathcal{A}_{1d}$ satisfy $|s_t| \leq \|\mathbf{g}_t\|_\star$.

$$\begin{aligned}
\mathrm{Regret}_T(\mathbf{u}) &\leq \sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{x}_t - \mathbf{u} \rangle = \sum_{t=1}^{T} \langle \mathbf{g}_t, z_t \tilde{\mathbf{x}}_t \rangle - \langle \mathbf{g}_t, \mathbf{u} \rangle \\
&= \underbrace{\sum_{t=1}^{T} \left(\langle \mathbf{g}_t, \tilde{\mathbf{x}}_t \rangle z_t - \langle \mathbf{g}_t, \tilde{\mathbf{x}}_t \rangle \|\mathbf{u}\| \right)}_{\text{linear regret of } \mathcal{A}_{1\text{d}} \text{ at } \|\mathbf{u}\| \in \mathbb{R}} + \sum_{t=1}^{T} \left(\langle \mathbf{g}_t, \tilde{\mathbf{x}}_t \rangle \|\mathbf{u}\| - \langle \mathbf{g}_t, \mathbf{u} \rangle \right) \\
&= \mathrm{Regret}_T^{\mathcal{A}_{1\text{d}}}(\|\mathbf{u}\|) + \sum_{t=1}^{T} \left(\langle \mathbf{g}_t, \tilde{\mathbf{x}}_t \rangle \|\mathbf{u}\| - \langle \mathbf{g}_t, \mathbf{u} \rangle \right) \\
&= \mathrm{Regret}_T^{\mathcal{A}_{1\text{d}}}(\|\mathbf{u}\|) + \|\mathbf{u}\| \sum_{t=1}^{T} \left(\langle \mathbf{g}_t, \tilde{\mathbf{x}}_t \rangle - \left\langle \mathbf{g}_t, \frac{\mathbf{u}}{\|\mathbf{u}\|} \right\rangle \right) \\
&= \mathrm{Regret}_T^{\mathcal{A}_{1\text{d}}}(\|\mathbf{u}\|) + \|\mathbf{u}\| \mathrm{Regret}_T^{\mathcal{A}_B} \left(\frac{\mathbf{u}}{\|\mathbf{u}\|} \right) .
\end{aligned}$$

New Trends

Related Papers

- Orabona and Pál. *Open Problem: Parameter-Free and Scale Free Online Learning*, COLT Open Problems, 2016.
- Orabona and Pál. *Coin Betting and Parameter-free Online Learning*, NIPS 2016.
- Kwang-Sung and Orabona. *Parameter-Free Online Convex Optimization with Sub-Exponential Noise*, COLT 2019.
- Chen, Langford, and Orabona. *Better Paratemer-free Stochastic Optimization with ODE Updates for Coin Betting*, Arxiv 2020.
- Cutkosky, and Orabona; *Black-Box Reductions for Parameter-Free Online Learning in Banach Spaces*, COLT 2018.

Subsection 3

Combining Online Learning Guarantees

New Trends

Combining Online Learning Guarantees

Theorem

Let $\mathcal{A}_1$ and $\mathcal{A}_2$ two OLO algorithms that produces the predictions $\mathbf{x}_{t,1}$ and $\mathbf{x}_{t,2}$ respectively. Then, predicting with $\mathbf{x}_t = \mathbf{x}_{t,1} + \mathbf{x}_{t,2}$, guarantees:

$$\sum_{t=1}^{T}\langle \mathbf{g}_t, \mathbf{x}_t\rangle - \sum_{t=1}^{T}\langle \mathbf{g}_t, \mathbf{u}\rangle = \min_{\mathbf{u}=\mathbf{u}_1+\mathbf{u}_2} \mathrm{Regret}_T^{\mathcal{A}_1}(\mathbf{u}_1) + \mathrm{Regret}_T^{\mathcal{A}_2}(\mathbf{u}_2) \ .$$

Proof.

Set $\mathbf{u}_1 + \mathbf{u}_2 = \mathbf{u}$. Then,

$$\sum_{t=1}^{T}\langle \mathbf{g}_t, \mathbf{x}_t\rangle - \sum_{t=1}^{T}\langle \mathbf{g}_t, \mathbf{u}\rangle = \sum_{t=1}^{T}\langle \mathbf{g}_t, \mathbf{x}_{t,1}\rangle - \sum_{t=1}^{T}\langle \mathbf{g}_t, \mathbf{u}_1\rangle + \sum_{t=1}^{T}\langle \mathbf{g}_t, \mathbf{x}_{t,2}\rangle - \sum_{t=1}^{T}\langle \mathbf{g}_t, \mathbf{u}_2\rangle \ . \qquad \square$$

- Cutkosky; *Combining Online Learning Guarantees*, COLT 2020.

Subsection 4

Predictable Sequences (a.k.a. Hints)

New Trends

Predictable Sequences (a.k.a. Hints)

- Regret guarantees can be loose if the sequence being encountered is not “worst-case”.
- We have a hint or predict of what the adversary is going to play next.

New Trends

Predictable Sequences (a.k.a. Hints)

- Regret guarantees can be loose if the sequence being encountered is not "worst-case".
- We have a hint or predict of what the adversary is going to play next.
- **Online Learning with Predictable Gradient Sequences**:

$$f_t \in \operatorname*{argmin}_{f \in \mathcal{F}} \eta \langle f, M_t \rangle + D_R(f, g_{t-1})$$

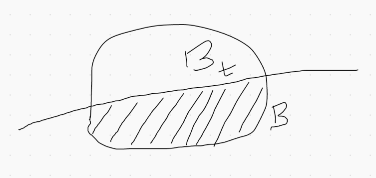
$$g_t \in \operatorname*{argmin}_{g \in \mathcal{F}} \eta \langle g, \nabla l_t \rangle + D_R(g, g_{t-1})$$

- $\mathrm{Regret}_T(\mathbf{u}) \leq \eta^{-1} R^2 + \frac{\eta}{2} \sum_{t=1}^{T} ||\nabla l_t - M_t||_*^2$

New Trends

Predictable Sequences (a.k.a. Hints)

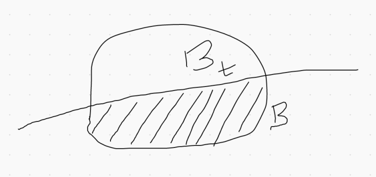
- At time t, adversary can play in B_t.
- Regret bounds available for $B_t = \{\mathbf{w} : \angle(\mathbf{w}, \mathbf{M}_t) \leq \alpha\}$
- What about a more general case?



New Trends

Predictable Sequences (a.k.a. Hints)

- At time t, adversary can play in B_t.
- Regret bounds available for $B_t = \{\mathbf{w} : \angle(\mathbf{w}, \mathbf{M}_t) \leq \alpha\}$
- What about a more general case?



- **Some Papers**:
  - Rakhlin, Sridharan; *Optimization, Learning and Games with Predictable Sequences*, NIPS 2013.
  - Rakhlin, Sridharan; *Online Learning with Predictable Sequences*, COLT 2013.
  - Dekel, Flajolet, Haghtalab, Jaillet; *Online Learning with a Hint*, NIPS 2017.
  - Bhaskara, Cutkosky, Kumar and Purohit; *Online Learning with Imperfect Hints*, ICML 2020.

Section 9

Bibliography

Bibliography

There are several good texts for a start:

1. Shai Shalev-Shwartz, *Online Learning and Online Convex Optimization*, Foundations and Trends in Machine Learning, 2011.
2. Francesco Orabona, *A Modern Introduction to Online Learning*, ArXiv, 2020.
3. Tor Lattimore and Csaba Szepesvári, *Bandit Algorithms*, Cambridge University Press, 2020.