1 Announcement

Recently, journal of Foundations and Trends in Machine Learning published a paper by Shai Shalev-Shwartz on “Online Learning and Online Convex Optimization”, [1], which can be considered as a good reference for this course.

2 Introduction

As a summary of the previous lecture, consider a finite class of experts called $H$. Consider two following cases

- if $\exists h^* \in H$ which is perfect, Halving algorithms claims $M \leq \log_2(|H|)$ where $M$ is the number of mistakes
- if $H$ is a class of linear classifiers and $\exists h^* \in H$ perfect with a margin $\gamma > 0$, Perceptron algorithm says $M \leq \left(\frac{2}{\gamma}\right)^2$.

The concern with these algorithms is that the realizability assumption is unrealistic. Now let us define a new game.

3 The online optimization game (between a player and an adversary)

For $t = 1, 2, \ldots$, here is the game

1. player chooses $w_t \in W$ where $W \subseteq \mathbb{R}^n$ is a feasible set of actions.
2. the adversary has infinite power and it chooses a loss function $f_t : W \rightarrow \mathbb{R}$.
3. the player suffers the loss function $f_t(w_t)$
4. the player updates $w_{t+1}$

Let us start with the following definition

**Definition 1.** The cumulative loss after $T$ rounds of play is $\sum_{t=1}^{T} f_t(w_t)$.

**What is the goal?**

The goal is to

- minimize the cumulative loss: the adversary can make the loss arbitrary large.
- we need to compare to a benchmark.

**Definition 2.** The player’s regret after $T$ rounds is defined $\sum_{t=1}^{T} f_t(w_t) - \min_{w \in W} \sum_{t=1}^{T} f_t(w)$. 
In words, we compare our cumulative loss to the “best” fixed point in hindsight.

**Goal:**
- prove regret bounds $R(T)$-function that upper bounds the regret $\forall T, \forall f_1, f_2, \ldots$
- $R(T)$-sublinear, $R(T) = O(\sqrt{T}, T^{2/3}, \ldots)$
- later we will relax “fixed” in various ways

- $W$ plays two roles
  1. set of actions for the player
  2. the “comparison” or the “competitor” class
- the adversary and the competitor class are two different entities

### 4 Online binary prediction

The online binary prediction is a special case of online optimization.

<table>
<thead>
<tr>
<th>online binary prediction</th>
<th>online optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>for $t = 1, 2, \ldots$ choose $h_t \in H$</td>
<td>for $t = 1, 2, \ldots$ choose $w_t \in W$</td>
</tr>
<tr>
<td>the adversary chooses $(x_t, y_t)$</td>
<td>the adversary chooses loss $f_t(w_t)$</td>
</tr>
<tr>
<td>loss at time $t$ is equal to $1_{h_t(x_t) \neq y_t}$</td>
<td>we suffer the loss function $f_t(w_t)$</td>
</tr>
</tbody>
</table>

We can think of $h_t$ as parametrized by $w_t$ and the loss function $f_t(w)$, in the online optimization, is $f_t(w) = 1_{h_t(x_t) \neq y_t}$.

### 5 Online convex optimization

In this problem

1. $W$ is a convex subset of $\mathbb{R}^n$
2. $f_t$ is convex for all $t$

Following is couple examples

- **Online linear regression** $w_t \in W = \mathbb{R}^n$, $x_t \in \mathbb{R}^n$, $y_t \in \mathbb{R}$, $f_t(w_t) = (w_t \cdot x_t - y_t)^2$
- **deg-2 linear regression** $(w_t, A_t) \in (\mathbb{R}^n \times \mathbb{R}^{n \times n})$, $x_t \in \mathbb{R}^n$, $y_t \in \mathbb{R}$, $f_t(w_t, A_t) = (w_t \cdot x_t + x_t^T A_t x_t - y_t)^2$
- **Air-dropping supplies** $W = \mathbb{R}^2$, $x_t \in \mathbb{R}^2$, $f_t(w_t) = ||w_t - x_t||_2$ or other norms such as $f_t(w_t) = ||w_t - x_t||_1 = \sum_{j=1}^n |w_{tj} - x_{tj}|$ like the Manhattan police example
- online portfolio management
- online to Batch conversion: a way to use an online learning algorithm for an offline learning algorithm
6 Follow the leader (FTL) algorithm

The update strategy in this algorithm is as follows

\[ w_{t+1} = \arg \min_{w \in W} \sum_{s=1}^{t} f_s(w) \]

\[ = \arg \min_{w \in W} f_{1:t}(w) \]

where the notation \( f_{1:t}(w) = \sum_{s=1}^{t} f_s(w) \) captures a batch of all data from time 1 to \( t \).

Example 1 Consider \( W \in [-1,1] \), \( f_t(w) = g_t w \) where \( g_t \in [-1,1] \). At time \( t \), we take the action \( w_t \) and the adversary chooses \( g_t \) and we suffer the loss. Then, we have the following table of couple steps

<table>
<thead>
<tr>
<th>( t )</th>
<th>( w_t )</th>
<th>( g_t )</th>
<th>loss</th>
<th>( g_{1:t} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>-0.5</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>1</td>
<td>-0.5</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

For example \( w_2 = \arg \min_{w \in [-1,1]} 0.5 w \). Therefore, the regret is of order \( T, R(T) = O(T) \).

References