# Machine Learning Theory

**Professor Jacob Abernethy**

**Organized by Yiwen Chen**

# Contents

# Preface

This is the organized course material of CS 7545 Machine Learning Theory from Georgia Institute of Technology. The material is based on the course in Fall 2019. The instructor of this course is professor Jacob Abernethy and the course website can be found via this link.

This course contains many theoretical aspects of machine learning, including decision making, online learning, boosting, regret minimization, etc. Students will get familiar with the frontiers of theoretical machine learning research after finishing this course. It is a totally theoretical course and no programming will be covered.

CS 7545 is a graduate level course, thus it has several prerequisites. Familiarity with Algorithm analysis, linear algebra, probability and statistics is a must to understand the contents of this course. Convex analysis background ~~is strongly recommended but not required~~ is also very important.

Before studying any algorithm and theory, we will first review some math aspects and point out some notations that we will use in this course material. Afterwards, we will talk about convex analysis and deviation bound. Online learning algorithms will be covered next. Finally, we will talk about statistical learning theory.

Thanks to professor Jacob Abernethy for his instruction.

# Chapter 1

# Math Review

In this chapter, we will briefly review several aspects of linear algebra, norm, etc. The following notations will be used throughout the course.

- $M$: Matrix (size $\mathbb{R}^{m\times n}$).
- $M^T$: Transpose Matrix of $M$.
- $\mathbf{x}$: Vector (size $\mathbb{R}^n$).
- $\mathbf{x}_i$: $i$-th element of vector $\mathbf{x}$ (size $\mathbb{R}$).

## 1.1 Positive Semidefinite and Positive Definite Matrices

We will first define Positive semidefinite (PSD) and positive definite matrices.

**Definition 1.1 (Positive Semidefinite Matrix).** *A matrix $M \in \mathbb{R}^{n\times n}$ is said to be positive semidefinite (PSD) if it satisfies the following two conditions.*

1. *$M$ is a symmetric matrix. That is $M^T = M$.*
2. *For all $\mathbf{x} \in \mathbb{R}^n$, we have that $\mathbf{x}^T M\mathbf{x} \geq 0$. This condition is equivalent to all eigenvalues of $M$ being non-negative.*

*We can denote matrix $M$ being positive semidefinite (PSD) as $M \succeq 0$.*

**Definition 1.2 (Positive Definite Matrix).** *Similar to above definition, a matrix $M$ is said to be positive definite (PD) if it satisfies the following two conditions:*

1. *$M$ is a symmetric matrix. That is $M^T = M$.*
2. *For all $\mathbf{x} \in \mathbb{R}^n$, we have that $\mathbf{x}^T M\mathbf{x} > 0$. This condition is equivalent to all eigenvalues of $M$ being positive.*

*We can denote matrix $M$ being positive definite (PD) as $M \succ 0$.*

Notice that the only difference between positive semidefinite matrices and positive definite matrices is whether the eigenvalues could be zero. In addition, all PD matrices are PSD, while PSD matrices may not be PD.

**Notation 1.3.** *We denote $M \preceq 0$ if $-M \succeq 0$ and $M \prec 0$ if $-M \succ 0$.*

## 1.2 Norm

**Definition 1.4 (Norm).** *Norm is defined as a function $||\cdot|| : \mathbb{R}^n \to [0, \infty]$ which satisfies the following conditions:*

1. ***Identity of Indiscernibles:*** $||\mathbf{x}|| = 0$ *if and only if* $\mathbf{x} = 0$.
2. ***Absolute Homogeneity:*** $||\alpha\mathbf{x}|| = |\alpha|||\mathbf{x}||$ *for all* $\mathbf{x} \in \mathbb{R}^n$ *and* $\alpha \in \mathbb{R}$.
3. ***Triangle Inequality:*** $||\mathbf{x}+\mathbf{y}|| \le ||\mathbf{x}|| + ||\mathbf{y}||$ *for all* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

Some famous norm functions can be seen following:

1. $\boldsymbol{\ell}_1$ norm: $||\mathbf{x}||_1 = \sum_{i=1}^n |\mathbf{x}_i|$.
2. $\boldsymbol{\ell}_2$ norm: $||\mathbf{x}||_2 = \sqrt[2]{\sum_{i=1}^n \mathbf{x}_i^2}$.
3. $\boldsymbol{\ell}_\infty$ norm: $||\mathbf{x}||_\infty = \max\limits_{1\le i\le n} |\mathbf{x}_i|$.
4. $\boldsymbol{\ell}_p$ norm: $||\mathbf{x}||_p = \sqrt[p]{\sum_{i=1}^n |\mathbf{x}_i|^p}$
5. $M$ norm (suppose $M \in \mathbb{R}^{n\times n}$): $||\mathbf{x}||_M = \sqrt[2]{\mathbf{x}^T M \mathbf{x}}$.

We can see that $\boldsymbol{\ell}_1$, $\boldsymbol{\ell}_2$ and $\boldsymbol{\ell}_\infty$ are special case of $\boldsymbol{\ell}_p$ norm.

We will next prove that $\boldsymbol{\ell}_\infty$ norm satisfies the three condition in norm definition.

*Proof.* We will just use the norm definition to finish the proof.

1. **Identity of Indiscernibles**:
   Since $||\mathbf{x}||_\infty = \max\limits_{1\le i\le n} |\mathbf{x}_i|$. $||\mathbf{x}||_\infty = 0$ if and only if $\mathbf{x}_i = 0$ for all $i \in [1, n]$.
2. **Absolute Homogeneity**:
   $||\alpha\mathbf{x}||_\infty = \max\limits_{1\le i\le n} |\alpha\mathbf{x}_i| = |\alpha| \max\limits_{1\le i\le n} |\mathbf{x}_i| = |\alpha|||\mathbf{x}||_\infty$.
3. **Triangle Inequality**:
   $||\mathbf{x}+\mathbf{y}||_\infty = \max\limits_{1\le i\le n} |\mathbf{x}_i + \mathbf{y}_i| \le \max\limits_{1\le i\le n} (|\mathbf{x}_i| + |\mathbf{y}_i|) \le \max\limits_{1\le i\le n} |\mathbf{x}_i| + \max\limits_{1\le j\le n} |\mathbf{y}_j| = ||\mathbf{x}||_\infty + ||\mathbf{y}||_\infty$.

□

**Definition 1.5 (Dual Norm).** *Given a norm $||\cdot||$, we can define its dual norm $||\cdot||_*$ as following:*

$$||\mathbf{y}||_* = \sup_{\mathbf{x}\in\mathbb{R}^n, ||\mathbf{x}||=1} \langle \mathbf{x}, \mathbf{y} \rangle \tag{1.1}$$

*where $\langle , \rangle$ is the inner product of two vectors.*

**Claim 1.6.** *The $\boldsymbol{\ell}_2$ norm's dual norm is $\boldsymbol{\ell}_2$ norm itself.*

*Proof.* For any vector $\mathbf{y} \in \mathbb{R}^n$, we have that

$$||\mathbf{y}||_{2,*} = \sup_{\mathbf{x} \in \mathbb{R}^n, ||\mathbf{x}||_2 = 1} \langle \mathbf{x}, \mathbf{y} \rangle = \sup_{\mathbf{x} \in \mathbb{R}^n} \langle \frac{\mathbf{x}}{||\mathbf{x}||_2}, \mathbf{y} \rangle = \frac{1}{||\mathbf{y}||_2} \langle \mathbf{y}, \mathbf{y} \rangle = ||\mathbf{y}||_2.$$

Notice that here the inner product is maximized when two vectors ($\frac{\mathbf{x}}{||\mathbf{x}||_2}$ and $\mathbf{y}$) take the same direction. □

**Claim 1.7.** *The $\boldsymbol{\ell}_p$ norm is dual to $\boldsymbol{\ell}_q$ norm if and only if the following equation holds:*

$$\frac{1}{p} + \frac{1}{q} = 1 \tag{1.2}$$

**Claim 1.8.** *For a PD Matrix $M$, the dual norm of $M$ norm is $M^{-1}$ norm, where $M^{-1}$ is the inverse matrix of $M$.*

**Theorem 1.9 (Hölders Inequality).** *Suppose $|| \cdot ||$ and $|| \cdot ||_*$ are a pair of dual norms. Then for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we can have that*

$$\langle \mathbf{x}, \mathbf{y} \rangle \leq ||\mathbf{x}|| ||\mathbf{y}||_*. \tag{1.3}$$

*Proof.* By definition,

$$||\mathbf{x}|| ||\mathbf{y}||_* = ||\mathbf{x}|| \sup_{\mathbf{z} \in \mathbb{R}^n, ||\mathbf{z}||_2 = 1} \langle \mathbf{z}, \mathbf{y} \rangle \geq ||\mathbf{x}|| \langle \frac{\mathbf{x}}{||\mathbf{x}||}, \mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle.$$

Here we use the fact that the supremum over $\mathbf{z}$ must be larger or equal than taking any vector $\mathbf{x}$. □

This theorem will be very useful in the upcoming chapters.

# Chapter 2

# Convex Analysis

## 2.1 Convex Set and Convex Function

In this section, we will review some concepts in convex analysis.

First, let us define some basic and useful notations.

**Definition 2.1 (Gradient).** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function and let $\mathbf{x} \in \mathbb{R}^n$. We can defined the gradient of $f$ at $\mathbf{x}$ as*

$$\nabla f(\mathbf{x}) = (\frac{\partial f}{\partial \mathbf{x}_1}(\mathbf{x}), \cdots, \frac{\partial f}{\partial \mathbf{x}_n}(\mathbf{x})). \tag{2.1}$$

**Remark 2.2.** *Notice that here we define gradient as a row vector ($\mathbb{R}^{1\times n}$). However, for convenience, we may abuse this notation and use it as a column vector ($\mathbb{R}^{n\times 1}$) in the following contents.*

**Definition 2.3 (Hessian).** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice differentiable function and let $\mathbf{x} \in \mathbb{R}^n$. Then the Hessian of $f$ at $\mathbf{x}$ is defined as*

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(\mathbf{x}) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(\mathbf{x}) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(\mathbf{x}). \end{pmatrix} \tag{2.2}$$

Notice that if $f$ is a twice differentiable function and $\mathbf{x} \in \mathbb{R}^n$, then $\nabla^2 f(\mathbf{x})$ is a symmetric matrix since for all $i, j \in [1, n]$, it has

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(\mathbf{x}) = \frac{\partial^2 f}{\partial x_j \partial x_i}(\mathbf{x}).$$

Next, we will talk about convexity, its definition and several useful facts.

**Definition 2.4 (Convex Set).** *Let $\mathcal{K} \subset \mathbb{R}^n$, set $\mathcal{K}$ is called a convex set if $\forall \mathbf{x}, \mathbf{y} \in \mathcal{K}$ and $\forall t \in [0, 1]$, we have*

$$t\mathbf{x} + (1-t)\mathbf{y} \in \mathcal{K}. \tag{2.3}$$

*The meaning of this definition is that for any two arbitrary points in the set $\mathcal{K}$, the straight line between these two points are all contained in $\mathcal{K}$.*

**Definition 2.5** (**Convex Function**)**.** *Let $\mathcal{K} \subset \mathbb{R}^n$ be a convex set and $f : \mathcal{K} \to \mathbb{R}^n$ be a differentiable function. Then function $f$ is convex if $\forall \mathbf{x}, \mathbf{y} \in \mathcal{K}$ and $\forall t \in [0, 1]$, we have that*

$$f((1-t)\mathbf{x} + t\mathbf{y}) \leq (1-t)f(\mathbf{x}) + tf(\mathbf{y}). \tag{2.4}$$

**Claim 2.6.** *Let $\mathcal{K} \subset \mathbb{R}^n$ be a convex set and $f : \mathcal{K} \to \mathbb{R}^n$ be a differentiable function. Then function $f$ is convex if and only if $\forall \mathbf{x}, \mathbf{y} \in \mathcal{K}$, we have that*

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle. \tag{2.5}$$

An intuition behind this claim is that $f$ is convex if and only if the first order approximation of $f$ at any point $\mathbf{y}$ is not larger than the function itself.

**Claim 2.7.** *Let $\mathcal{K} \subset \mathbb{R}^n$ be a convex set and $f : \mathcal{K} \to \mathbb{R}$ be a twice differentiable function. Then $f$ is convex if and only if $\forall \mathbf{x} \in \mathcal{K}$,*

$$\nabla^2 f(\mathbf{x}) \succeq 0. \tag{2.6}$$

**Remark 2.8.** *It can be seen that above three formulas* (2.4), (2.5) *and* (2.6) *are equivalent. However, in reality, if $f$ is twice differentiable, using* (2.6) *is much simpler than using* (2.4) *or* (2.5), *because it only contains one variable in $\mathcal{K}$ and we only need to determine if $\nabla^2 f(\mathbf{x})$ is PSD or not.*

*In facts, many functions satisfy this condition. Some examples are $f(\mathbf{x}) = \mathbf{x}^T\mathbf{x}$, $f(\mathbf{x}) = 1$, etc.*

**Proposition 2.9.**

1. *If $f$ is convex and $g$ is convex, then $f + g$ is convex.*
2. *If $f$ is convex, then $\forall \alpha \geq 0$, $\alpha f$ is convex.*
3. *If $f$ is convex and $g$ is convex, then $\max\{f, g\}$ is convex.*
4. *If $g(\mathbf{x}, \mathbf{y})$ is jointly convex in $\mathbf{x}$, $\mathbf{y}$, then $f(\mathbf{x}) = \inf_{\mathbf{y}} g(\mathbf{x}, \mathbf{y})$ is convex.*

**Definition 2.10** (**Concave Function**)**.** *Let $\mathcal{K} \subset \mathbb{R}^n$ be a convex set and $f : \mathcal{K} \to \mathbb{R}$, then $f$ is concave is $-f$ is convex.*

We can see that one example of concave function is *log* function.

**Definition 2.11** (Lipschitz)**.** *Let $||\cdot||$ be a norm, $c \geq 0$ and $f : \mathbb{R}^n \to \mathbb{R}$. Then $f$ is called $c-$ Lipschitz with respect to $||\cdot||$ if $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$,*

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq c||\mathbf{x} - \mathbf{y}||. \tag{2.7}$$

*Intuitively, it means that the difference between function value is smaller than the difference between the norm of difference between two points times some constant c.*

**Claim 2.12.** *If $f$ is a differentiable function, then $f$ is c-Lipschitz with respect to $||\cdot||$ if and only if $\forall \mathbf{x} \in \mathbb{R}^n$,*

$$||\nabla f(\mathbf{x})||_* \leq c \tag{2.8}$$

*where $||\cdot||_*$ is the dual norm of $||\cdot||$.*

*Proof.* We first prove the 'if' part.

Let $||\cdot||$ be a norm on $\mathbb{R}^n$ and $c \geq 0$, let $||\nabla f(\mathbf{x})||_* \leq c \; \forall \mathbf{x} \in \mathbb{R}^n$. Then, by the mean value theorem, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, there exists $t \in [0,1]$ such that

$$f(\mathbf{x}) = f(\mathbf{y}) + \langle \nabla f((1-t)\mathbf{x} + t\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle.$$

Then, by Hölder's inequality,

$$\begin{aligned} |f(\mathbf{x}) - f(\mathbf{y})| &= |\langle \nabla f((1-t)\mathbf{x} + t\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle| \\ &\leq ||\nabla f((1-t)\mathbf{x} + t\mathbf{y})||_* ||\mathbf{x} - \mathbf{y}|| \\ &\leq c||\mathbf{x} - \mathbf{y}||. \end{aligned}$$

That is, f is *c-Lipschitz* with respect to $||\cdot||$.

We will next prove the 'only if' part.

Suppose that $f$ is *c-Lipschitz* with respect to $||\cdot||$ and let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, then we have

$$\begin{aligned} \langle \nabla f(\mathbf{x}), \mathbf{y} \rangle &= \lim_{h \to 0} \frac{f(\mathbf{x} + h\mathbf{y}) - f(\mathbf{y})}{h} \\ &\leq \lim_{h \to 0} \frac{c||\mathbf{x} + h\mathbf{y} - \mathbf{x}||}{h} \\ &= c \lim_{h \to 0} \frac{h||\mathbf{y}||}{h} \\ &= c||\mathbf{y}||. \end{aligned}$$

Notice that $\langle \nabla f(\mathbf{x}), \mathbf{y} \rangle$ is the directional derivative of $f$ at $\mathbf{x}$ in the direction of $\mathbf{y}$. Then,

$$||\nabla f(\mathbf{x})||_* = \sup_{||\mathbf{y}|| \leq 1} \langle \nabla f(\mathbf{x}), \mathbf{y} \rangle \leq \sup_{||\mathbf{y}|| \leq 1} c||\mathbf{y}|| = c.$$

Therefore, $\forall \mathbf{x} \in \mathbb{R}^n$, $||\nabla f(\mathbf{x})|| \leq c$. □

**Theorem 2.13 (Jensen's Inequality).** *Let $\mathcal{K} \in \mathbb{R}^n$ be a convex set, $X$ be a random variable on $\mathcal{K}$ and $f : \mathcal{K} \to \mathbb{R}$ be a convex function, then*

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)] \tag{2.9}$$

*where $\mathbb{E}$ denotes the expectation of a random variable.*

**Theorem 2.14 (Young's Inequality).** *Let $p, q > 0$ and satisfy that $\frac{1}{p} + \frac{1}{q} = 1$. Then $\forall a, b > 0$,*

$$ab \leq \frac{a^p}{p} + \frac{b^q}{q}. \tag{2.10}$$

*Proof.* Since $-\log$ is a convex function, then $\forall a, b > 0$,

$$\begin{aligned} \log(ab) &= \log(a) + \log(b) \\ &= \frac{p}{p}\log(a) + \frac{q}{q}\log(b) \\ &= \frac{1}{p}\log(a^p) + \frac{1}{q}\log(b^q) \\ &\leq \log(\frac{a^p}{p} + \frac{b^q}{q}). \end{aligned}$$

Then apply exp to both side and since exp function is monotonically increasing, we can have that

$$ab \leq \frac{a^p}{p} + \frac{b^q}{q}.$$

□

**Definition 2.15 (Strongly Convex).** *Let $\mathcal{K}$ be a convex set, $f : \mathcal{K} \to \mathbb{R}$ be a differentiable function and $\alpha > 0$. Then $f$ os $\alpha$-strongly convex with respect to $||\cdot||$ if $\forall \mathbf{x}, \mathbf{y} \in \mathcal{K}$,*

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\alpha}{2}||\mathbf{x} - \mathbf{y}||^2. \tag{2.11}$$

Notice that if $f$ is strongly convex, then it must be a convex function. In addition, a strongly convex function grows at least quadratically.

**Claim 2.16.** *Let $f$: dom($f$) $\to \mathbb{R}$ be a twice differentiable function, then $f$ is $\alpha$-strongly convex if and only if $\forall \mathbf{x} \in$ dom($f$),*

$$\nabla^2 f(\mathbf{x}) - \alpha I \succeq 0.$$

**Definition 2.17 (Smooth).** *Let $f$: dom($f$) $\to \mathbb{R}$ be a differentiable function, let $\alpha > 0$, then $f$ is $\alpha$-smooth if $\forall \mathbf{x}, \mathbf{y} \in$ dom($f$),*

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\alpha}{2}||\mathbf{x} - \mathbf{y}||^2.$$

Notice that in contrast to strongly convex, a smooth function grows **at most** quadratically.

**Claim 2.18.** *We have a similar claim for smooth function to strongly convex function. That is, let $f$: dom($f$) $\to \mathbb{R}$ be a twice differentiable function, then $f$ is $\alpha$-smooth if and only if $\forall \mathbf{x} \in$ dom($f$), we have*

$$\nabla^2 f(\mathbf{x}) - \alpha I \preceq 0.$$

Notice that although smooth seems to be contrary to strongly convex, a function $f$ can be both $\alpha$-strongly convex and $\beta$-smooth for some $\alpha$ and $\beta$. We will give an example next.

**Example 2.18.1.** *Let $M \in \mathbb{R}^{n \times n}$ be a positive definite matrix and let $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T M \mathbf{x}$. Denote $\lambda_{min}$ as the smallest eigenvalue of $M$ and $\lambda_{max}$ as the largest eigenvalue of $M$. Then we can prove easily that $f$ is $\lambda_{min}$-strongly convex and $\lambda_{max}$-smooth.*

*Proof.* First, by taking derivative, it can be seen that $\nabla f(\mathbf{x}) = \mathbf{x}^T M$. Then, take the second derivative and we can get $\nabla^2 f(\mathbf{x}) = M$. By Claim 2.18 and Claim 2.16, we can show that $f$ is $\lambda_{min}$-strongly convex and $\lambda_{max}$-smooth. □

## 2.2 Bregman Divergence

In this section, we will define Bregman Divergence and see its relation with convexity analysis.

**Definition 2.19.** *Given a function $f$: $\mathcal{U} \to \mathbb{R}$, we can define its Bregman Divergence as*

$$D_f(\mathbf{x}, \mathbf{y}) := f(\mathbf{x}) - f(\mathbf{y}) - \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle. \tag{2.12}$$

**Example 2.19.1.** *Let $f(\mathbf{x}) = \frac{1}{2}||\mathbf{x}||_2^2$, then*

$$D_f(\mathbf{x}, \mathbf{y}) = \frac{1}{2}||\mathbf{x} - \mathbf{y}||_2^2.$$

*Notice that only in this situation is Bregman divergence a quadratic function.*

**Example 2.19.2.** *Let $\Delta^n = \{\mathbf{p} \in \mathbb{R}^n | \sum_{i=1}^n p_i = 1, p_i \geq 0\}$ be the simplex with dimension $n$. Let $f(\mathbf{p}) = \sum_{i=1}^n p_i \log p_i$ be the entropy function and suppose $f(0) = 0$. Then*

$$D_f(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^n p_i \log \frac{p_i}{q_i} = KL(\mathbf{p}||\mathbf{q}),$$

*where $KL$ is Kullback-Leibler divergence.*

We will next talk about some facts about Bregman divergence.

**Fact 2.19.1.**

1. *If $f$ is convex, then $D_f(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$.*
2. *$f$ is $\mu$-strongly convex if and only if $D_f(\mathbf{x}, \mathbf{y}) \geq \frac{\mu}{2}||\mathbf{x} - \mathbf{y}||^2$.*
3. *$f$ is $\beta$-smooth if and only if $D_f(\mathbf{x}, \mathbf{y}) \leq \frac{\beta}{2}||\mathbf{x} - \mathbf{y}||^2$*
4. *If $D_f(\mathbf{x}, \mathbf{y}) = D_f(\mathbf{y}, \mathbf{x})$, then $f$ is quadratic.*
5. *Pinsker's Inequality: $KL(\mathbf{p}||\mathbf{q}) \geq \frac{1}{2}||\mathbf{p} - \mathbf{q}||_1^2$. It means that entropy function is 1-strongly convex in $||\cdot||_1$.*

**Definition 2.20 (Fenchel Dual Conjugate).** *Let $f$ be convex, the Fenchel dual conjugate of $f$ is defined as*

$$f^*(\theta) = \sup_{\mathbf{x} \in dom(f)} \left[\langle \mathbf{x}, \theta \rangle - f(\mathbf{x})\right]. \tag{2.13}$$

**Claim 2.21.** *$f^*(\theta)$ is convex.*

*Proof.* It can be seen that $g_{\mathbf{x}}(\theta) = \langle \mathbf{x}, \theta \rangle - f(\mathbf{x})$ is a linear function. Linear function is convex. In addition, $f^*(\theta) = \sup_{\mathbf{x} \in \text{dom}(f)} g_{\mathbf{x}}(\theta)$ is the supreme of convex function, which means that $f$ is also convex. □

**Example 2.21.1.** *If $f(\mathbf{x}) = \frac{1}{2}||\mathbf{x}||_2^2$, then $f^*(\theta) = \frac{1}{2}||\theta||_2^2$.*

**Example 2.21.2.** *If $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T M \mathbf{x}$ and $M$ is a positive definite matrix,*

$$f^*(\theta) = \sup_{\mathbf{x} \in dom(f)} \left[\langle \mathbf{x}, \theta \rangle - \frac{1}{2}\mathbf{x}^T M \mathbf{x}\right].$$

*Let $g(\mathbf{x}) = \langle \mathbf{x}, \theta \rangle - \frac{1}{2}\mathbf{x}^T M \mathbf{x}$. Take the derivative over $\mathbf{x}$ and we can get that*

$$\begin{aligned} &\nabla_{\mathbf{x}} g(\mathbf{x}) = 0 \\ \Leftrightarrow\ &\theta - M\mathbf{x} = 0 \\ \Leftrightarrow\ &\mathbf{x} = M^{-1}\theta. \end{aligned}$$

*Substitute the value of* $\mathbf{x}$ *into* $f^*(\theta)$ *and we can get that*

$$\begin{aligned} f^*(\theta) &= \theta^T M^{-1}\theta - \frac{1}{2}(M^{-1}\theta)^T M M^{-1}\theta \\ &= \frac{1}{2}\theta^T M^{-1}\theta. \end{aligned}$$

**Example 2.21.3.** *Let* $f(x) = \frac{1}{p}||\mathbf{x}||_p^p = \frac{1}{p}\sum_{i=1}^{n} x_i^p$*, then*

$$f^*(\theta) = \frac{1}{q}||\theta||_q^q,$$

*where* $\frac{1}{p} + \frac{1}{q} = 1$*.*

There is one famous inequality for Fenchel dual conjugate called Fenchel-Yang Inequality.

**Theorem 2.22** (Fenchel-Yang Inequality)**.** $\forall \mathbf{x} \in$ *dom*$(f)$ *and* $\forall\theta \in$ *dom*$(f^*)$*, it holds that*

$$f(\mathbf{x}) + f^*(\theta) \geq \langle \mathbf{x}, \theta \rangle. \tag{2.14}$$

*Proof.* By the definition of Fenchel dual conjugate, $\forall\mathbf{x}$ and $\theta$,

$$\begin{aligned} f^*(\theta) + f(\mathbf{x}) &= \sup_{\mathbf{y}} [\langle \mathbf{y}, \theta \rangle - f(\mathbf{y})] + f(\mathbf{x}) \\ &\geq \langle \mathbf{x}, \theta \rangle - f(\mathbf{x}) + f(\mathbf{x}) \\ &= \langle \mathbf{x}, \theta \rangle. \end{aligned}$$

□

**Corollary 2.23.** *By the inequality above, we can get directly that*

$$\frac{1}{p}||\mathbf{x}||_p^p + \frac{1}{q}||\theta||_q^q \geq \langle \mathbf{x}, \theta \rangle.$$

I will next discuss some facts about Fenchel dual.

**Fact 2.23.1.**

1. *If* $f$ *is closed, then* $(f^*)^* = f$*.*
2. $\nabla f(\nabla f^*(\theta)) = \theta$ *and* $\nabla f^*(\nabla f(\mathbf{x})) = \mathbf{x}$*.*
3. *If* $f$ *is differentiable, then* $D_f(\mathbf{x}, \mathbf{y}) = D_{f^*}(\nabla f(\mathbf{y}), \nabla f(\mathbf{x}))$*.*
4. *If* $f$ *is* $\mu$*-strongly convex with respect to* $||\cdot||$*, then* $f^*$ *is* $\frac{1}{\mu}$*-smooth with respect to* $||\cdot||_*$*, where* $||\cdot||_*$ *is dual to* $||\cdot||$*.*

# Chapter 3

# Deviation Bound

In this chapter several famous deviation bounds are mentioned. There bounds will be heavily used in later chapters.

First of all, some simple definitions from measure theory will be discussed.

**Definition 3.1** (**Random Variable**). *A random variable $X$ is a measurable function from a sigma algebra $\Omega \to \mathbb{R}$.*

The definition of measurable function and sigma algebra is included in measure theory, which will not be discussed here.

**Definition 3.2** (**Cumulative Distribution Function**). *The CDF (cumulative distribution function) pf a random variable $X$ is defined as*

$$F(t) := \Pr(X \leq t). \tag{3.1}$$

**Definition 3.3** (**Probability Density Function**). *If $F(t)$ is differentiable, then the PDF (probability density function) is defined as*

$$f(t) = F'(t). \tag{3.2}$$

**Definition 3.4** (**Variance**). *The variance of a random variable $X$ is defined as*

$$Var(X) = \mathbb{E}[(X - \mu)^2], \tag{3.3}$$

*where $\mu = \mathbb{E}(X)$ is the expectation of $X$.*

After the above definitions, we will next introduction some useful deviation bounds.

**Theorem 3.5** (**Markov's Inequality**). *Let $X$ be a random variable and $X \geq 0$, then $\forall t \geq 0$, we have that*

$$\Pr(X \geq t) \leq \frac{\mathbb{E}[X]}{t}. \tag{3.4}$$

*Proof.* The proof of above theorem is not very complex.

First of all, let $Z_t := \mathbf{1}(X \geq t) \cdot t$, where $\mathbf{1}(\cdot)$ is the indicator function.

Notice that $X \geq Z_t$ for all $t$. The reason is that if $X < t$, then $Z_t = 0 \leq X$. On the contrary, if $X \geq t$, then $Z_t = t \leq X$.

Use this, we can know that

$$\begin{aligned}\mathbb{E}(X) &\geq \mathbb{E}(Z_t)\\ &= \mathbb{E}[\mathbf{1}(X \geq t)] \cdot t\\ &= \Pr(X \geq t) \cdot t\end{aligned}$$

This directly shows that $\Pr(X \geq t) \leq \frac{\mathbb{E}[X]}{t}$. □

Since $\Pr(X \geq t) = 1 - F(t)$, we can know from above theorem that $F(t) \geq 1 - \frac{\mathbb{E}(X)}{t}$.

**Theorem 3.6** (Chebyshev's Inequality)**.** *Let $\mathbb{E}(X) = \mu$ and $Var(X) = \sigma^2$, then $\Pr(|X - \mu| \geq t) \leq \frac{\sigma^2}{t^2}$.*

It turns out that this theorem can be easily proved with Markov Inequality.

*Proof.*

$$\begin{aligned}\Pr(|X - \mu| \geq t) &= \Pr(|X - \mu|^2 \geq t^2)\\ &\leq \frac{\mathbb{E}[(X - \mu)]^2}{t^2}\\ &= \frac{\sigma^2}{t^2}.\end{aligned}$$

□

Next, we will define gaussian distribution (also called normal distribution).

**Theorem 3.7** (Gaussian Distribution)**.** *Let's say $X \sim N(\mu, \sigma^2)$, it means that the pdf of $X$ is*

$$P(X = x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x - \mu)^2}{2\sigma^2}).$$

This distribution is heavily used in many different areas.

**Fact 3.7.1.** *If $X \sim N(0, \sigma)$, and $t > 0$, then $\mathbb{E}[\exp(tX)] = \exp(\frac{t^2 s^2}{2})$.*

This can be proved just by taking integral of this pdf function.

Next, we will define subgaussian distribution, which will be used later in some inequalities.

**Definition 3.8** (Subgaussian Distribution)**.** *A random variable $X$ with $\mathbb{E}(X) = 0$ is a subgaussian distribution with respect to variance proxy $\sigma^2$ if*

$$\mathbb{E}[\exp(tX)] \leq \exp(\frac{t^2\sigma^2}{2}). \tag{3.5}$$

We will then give an example of subgaussian distribution.

**Example 3.8.1.** *Let $X$ be a bounded random variable, that is, $a \leq X \leq b$ and $\mathbb{E}(X) = 0$. Then $X$ is subgaussian with respect to variance proxy $\frac{(b-a)^2}{4}$, which means*

$$\mathbb{E}[\exp(tX)] \leq \exp(\frac{t^2(b - a)^2}{8}).$$

This is also called Hoeffiding's Lemma.

**Claim 3.9.** *If a random variable $X$ is subgaussian with respect to variance proxy $\sigma^2$, then*

$$P(|X| > t) \leq 2\exp(-\frac{t^2}{2\sigma^2}). \tag{3.6}$$

Using subgaussian, we can introduce Hoeffding's inequality, which is extremely helpful in later material.

**Theorem 3.10 (Hoeffding's Inequality).** *Let $X_1, \cdots, X_n$ be independent random variables, with $\mathbb{E}(X_i) = \mu_i$ and $a_i \leq X_i \leq b_i$. Let $\bar{X_n} = \frac{1}{n}\sum\limits_{i=1}^{n} X_i$ and $\mu = \frac{1}{n}\sum\limits_{i=1}^{n} \mu_i$. It holds that*

$$\Pr(|\bar{X_n} - \mu| > t) \leq 2\exp(-\frac{2n^2t^2}{\sum\limits_{i=1}^{n}(a_i - b_i)^2}). \tag{3.7}$$

We will next give the proof of this theorem.

*Proof.* Notice that $\Pr(|\bar{X_n} - \mu| > t) \leq \Pr(\bar{X_n} - \mu > t) + \Pr(\bar{X_n} - \mu < -t)$. Then, we just need to prove that $\Pr(\bar{X_n} - \mu > t) \leq \exp(-\frac{2n^2t^2}{\sum\limits_{i=1}^{n}(a_i-b_i)^2})$.

$$\begin{aligned}
\Pr(\bar{X_n} - \mu > t) &= \Pr(\exp(s(\bar{X_n} - \mu)) > \exp(st)) \\
\text{(Markov Inequality)} &\leq \frac{\mathbb{E}[\exp(s(\bar{X_n} - \mu))]}{\exp(st)} \\
&= \exp(-st)\mathbb{E}[\exp(s(\bar{X_n} - \mu))] \\
&= \exp(-st)\mathbb{E}[\Pi_{i=1}^{n} \exp(\frac{s}{n}(X_i - \mu_i))] \\
\text{(Independence)} &= \exp(-st)\Pi_{i=1}^{n}\mathbb{E}[\exp(\frac{s}{n}(X_i - \mu_i))] \\
(X_i - \mu_i \text{ Subgaussian}) &\leq \exp(-st)\Pi_{i=1}^{n} \exp(\frac{s^2}{8n^2}(b_i - a_i)^2) \\
&= \exp(\frac{s^2}{8n^2}\sum_{i=1}^{n}(a_i - b_i)^2 - st) \\
(s \text{ takes } \frac{4tn^2}{\sum\limits_{i=1}^{n}(a_i - b_i)^2}) &= \exp(\frac{-2t^2n^2}{\sum\limits_{i=1}^{n}(a_i - b_i)^2})
\end{aligned}$$

Similarly, we can prove that $\Pr(\bar{X_n} - \mu < -t) \leq \exp(\frac{-2t^2n^2}{\sum\limits_{i=1}^{n}(a_i-b_i)^2})$. Combine the above two conclusions and we can get that $\Pr(|\bar{X_n} - \mu| > t) \leq 2\exp(\frac{-2t^2n^2}{\sum\limits_{i=1}^{n}(a_i-b_i)^2})$. □

Hoeffding's Inequality is very useful in proving other theorems. To see that, let's use it to prove the following claim.

**Claim 3.11.** *Let $X_1, \cdots, X_n$ be independent random variable and $0 \leq X_i \leq 1$, then with probability at least $1 - \delta$, it holds that*

$$|\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n} X_i]| \leq \sqrt{\frac{\log 2/\delta}{2n}}. \tag{3.8}$$

*Proof.* Let $\delta = 2\exp(-\frac{2n^2t^2}{\sum\limits_{i=1}^{n}(a_i-b_i)^2}) = 2\exp(-2nt^2)$ since $a_i = 0$ and $b_i = 1$. Then we can get that $t = \sqrt{\frac{\log 2/\delta}{2n}}$. By using Hoeffding's Inequality, we can get that

$$\Pr(|\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n} X_i]| \geq \frac{\log 2/\delta}{2n}) \leq \delta.$$

Therefore, we can get that $\Pr(|\frac{1}{n}\sum\limits_{i=1}^{n} X_i - \mathbb{E}[\frac{1}{n}\sum\limits_{i=1}^{n} X_i]| \leq \frac{\log 2/\delta}{2n}) \geq 1-\delta$. □

Afterwards, let's will introduction the concept of Martingale.

**Definition 3.12 (Martingale).** *A sequence of random variables $Z_0, Z_1, \cdots, Z_n$ is a Martingale sequence if the following two conditions are satisfied for any $i$*

1. $\mathbb{E}[|Z_i|] < \infty$
2. $\mathbb{E}[Z_i|Z_1, \cdots, Z_{i-1}] = Z_{i-1}$

Next let's see two examples of Martingale.

**Example 3.12.1.** *Let $X_1, \cdots, X_n$ be bounded i.i.d (independent and identically distributed) random variables, and $\mathbb{E}[X_i] = 0$. Let $Z_i = \sum\limits_{j=1}^{i} X_j$, and set $Z_0 = 0$, then $Z_0, \cdots, Z_n$ is a martingale.*

*Proof.*

$$\begin{aligned}
\mathbb{E}[Z_i|Z_1, \cdots Z_{i-1}] &= \mathbb{E}[Z_i|X_1, \cdots, X_{i-1}] \\
&= \mathbb{E}[X_1 + \cdots + X_i|X_1, \cdots, X_{i-1}] \\
&= \mathbb{E}[X_1 + \cdots, X_{i-1}|X_1, \cdots, X_{i-1}] + \mathbb{E}[X_i|X_1, \cdots, X_{i-1}] \\
&= X_1 + \cdots X_{i-1} + \mathbb{E}[X_i] \\
&= Z_{i-1} + 0 \\
&= Z_{i-1}.
\end{aligned}$$

□

**Example 3.12.2.** *Let $X_1, \cdots, X_n$ be i.i.d, with $\mathbb{E}[X_i] = 0$ and $Var(X_i) = \sigma^2$. Let $S_i = \sum\limits_{j=1}^{i} X_j$ and $Z_i = S_i^2 - i\sigma^2$. Then $Z_i$ is a martingale.*

*Proof.* Notice that

$$\begin{aligned}
\mathbb{E}[X_n^2] &= Var(X_n) + \mathbb{E}[X_n]^2 \\
&= Var(X_n) \\
&= \sigma^2.
\end{aligned}$$

In addition,

$$\begin{aligned}
\mathbb{E}[S_{n-1}X_n|X_1, \cdots, X_{n-1}] &= \mathbb{E}[X_n]\mathbb{E}[S_{n-1}|X_1, \cdots, X_{n-1}] \\
&= 0.
\end{aligned}$$

Therefore, we have that

$$\begin{aligned}\mathbb{E}[Z_n|X_1,\cdots,X_{n-1}] &= \mathbb{E}[S_n^2 - n\sigma^2|X_1,\cdots,X_{n-1}]\\ &= \mathbb{E}[(X_n + S_{n-1})^2 - n\sigma^2|X_1,\cdots,X_{n-1}]\\ &= \mathbb{E}[X_n^2 + S_{n-1}^2 + 2X_nS_{n-1} - n\sigma^2|X_1,\cdots,X_{n-1}]\\ &= \mathbb{E}[\sigma^2 + S_{n-1}^2 - n\sigma^2|X_1,\cdots,X_{n-1}]\\ &= S_{n-1}^2 - (n-1)\sigma^2\\ &= Z_{n-1}.\end{aligned}$$

□

Finally, we come to the last inequality of this chapter, which combines Hoeffding's Lemma and definition of Martingale.

**Theorem 3.13 (Azuma's Inequality).** *Let $Z_0, Z_1, \cdots, Z_n$ be a martingale, with $|Z_i - Z_{i-1}| \le c_i$ for some $c_1, \cdots, c_n$. Then we have that*

$$\Pr(Z_n - Z_0 > t) \le \exp(-\frac{t^2}{2\sum\limits_{i=1}^{n} c_i^2}). \tag{3.9}$$

*Proof.*

$$\begin{aligned}\Pr(Z_n - Z_0 > t) &= \Pr(\exp(s(Z_n - Z_0)) > \exp(st))\\ &\le \mathbb{E}[\exp(s(Z_n - Z_0))]\exp(-st)\\ &= \exp(-st)\mathbb{E}[\exp(s(Z_{n-1} - Z_0))\exp(s(Z_n - Z_{n-1}))]\\ &= \exp(-st)\mathbb{E}[\mathbb{E}[\exp(s(Z_{n-1} - Z_0))\exp(s(Z_n - Z_{n-1}))]|Z_0,\cdots,Z_{n-1}]\\ &= \exp(-st)\mathbb{E}[\exp(s(Z_{n-1} - Z_0))\mathbb{E}[\mathbb{E}[\exp(s(Z_n - Z_{n-1}))]|Z_0,\cdots,Z_{n-1}]\end{aligned}$$

Since $|Z_n - Z_{n-1}| \le c_n$, we can get that $Z_n - Z_{n-1} \in [-c_n, c_n]$. In addition, we know that $\mathbb{E}[Z_n - Z_{n-1}|Z_0,\cdots,Z_{n-1}] = Z_{n-1} - Z_{n-1} = 0$. Therefore, by using Hoeffding's Lemma, we know that

$$\begin{aligned}\Pr(Z_n - Z_0 > t) &\le \exp(-st)\mathbb{E}[\exp(s(Z_{n-1} - Z_0))\exp(\frac{s^2c_n^2}{2})\\ &= \exp(-st + \frac{s^2c_n^2}{2})\mathbb{E}[\exp(s(Z_{n-1} - Z_0))\\ &\le \exp(-st + \frac{s^2c_n^2}{2})\exp(\frac{s^2c_{n-1}^2}{2})\mathbb{E}[\exp(s(Z_{n-2} - Z_0))\\ &\cdots\\ &\le \exp(-st + \frac{s^2}{2}\sum_{i=1}^{n} c_i^2)\end{aligned}$$

Let $s = \frac{t^2}{\sum\limits_{i=1}^{n} c_i^2}$ and we can get that $\Pr(Z_n - Z_0 > t) \le \exp(-\frac{t^2}{2\sum\limits_{i=1}^{n} c_i^2})$. □

We can see that the above proof is a little bit similar to the proof of Hoeffding's Inequality in that they both take the exponential of both sides and then use Markov Inequality to transform probability into expectation. Afterwards, they perform some kind of relaxation for the expectation.

# Chapter 4

# Online Learning Algorithm

In this chapter we will talk about the basics of online learning algorithm step by step. We will first introduce the online learning setting and an algorithm called halving algorithm, which is simple but has strong assumption. Afterwards, we will relax the assumption and introduction other algorithms such like exponential weighted algorithm.

## 4.1 Online Learning and Halving Algorithm

First of all, let's introduce a simple setting of online learning. In this setting,

- Goal: predict rain/shine.
- Have a set of $N$ weather experts.
- On each day $t$, expert $i$ predicts $x_i^t \in \{0,1\}$.
- Based on the predictions of experts, the algorithm predicts $\hat{y^t} \in \{0,1\}$.
- Nature reveals $y^t \in \{0,1\}$.
- The number of mistakes increases by one if $\hat{y^t} \neq y^t$.
- Assume there exists an perfect expert $j$, such that $x_j^t = y^t$ for all $t$.

We hope to develop an algorithm $\mathcal{A}$ such that we can get an upper bound of total number of mistakes. In this setting, we can find a simple yet good algorithm called Halving algorithm such that the total number of mistakes is no larger than $\log N$.

**Algorithm 1** Halving Algorithm

```
Let a set $C_1 = \{1, 2, \cdots, N\}$.
For $t = 1, 2, \cdots$
    Observe $x_i^t$, $\forall i \in C_t$.
    $\hat{y^t} = round(\frac{1}{|C_t|} \sum_{i \in C_t} x_i^t)$
    Let $C_{t+1} = C_t$
    For all $i \in C_t$
        If $x_i^t \neq y_t$, remove $i$ from $C_{t+1}$.
```

Basically, the algorithm will output the round of the average over all experts' output in the set. Afterwards, it will remove all the experts whose output is not the same as ground truth and continue. For this algorithm, we can have the following claim.

**Claim 4.1.** *For halving algorithm, it satisfies that the total number of mistakes is no larger than* $\log_2 N$.

*Proof.* The proof is quite simple. Initially we have $|C_1| = N$. Each time the number of mistakes increases, we can have that $|C_{t+1}|/|C_t| \leq 1/2$ since more than half of experts produces the different prediction than ground truth.

Therefore, if the number of mistakes reaches $\log_2 N$, we will have that $|C_t| = 1$, which means the only expert in the set is the perfect expert and no more mistake will be made. □

Next, let's see an example of using halving algorithm to solve some problem.

**Example 4.1.1** (Betting on sports)**.**

*Problem setting: let's say there are* $n$ *sport teams. On each round* $t$*, two teams* $i_t$*,* $j_t$ *play a match. Algorithm* $\mathcal{A}$ *needs to predict if* $i_t$ *beats* $j_t$ *or vice versa. Then games happens, either* $i_t$ *or* $j_t$ *wins.*

*Assume there exists a permutation* $\pi^* \in S_n$*,* $i_t$ *beats* $j_t$ *if and only if* $\pi^*(i_t) > \pi^*(j_t)$*.*

*Now we need an algorithm that can minimize the number of mistakes to gain more revenue. We can design our algorithm by reducing to halving algorithm as following:*

**Algorithm 2** Betting Algorithm

```
  Treat every $\pi \in S_n$ as an expert.
  Let the prediction of expert $i$ at round $t$ be $x_i^t = \mathbf{1}[\pi(i_t) > \pi(j_t)]]$.
  Let the output of nature as $y_t = \mathbf{1}[\pi^*(i_t) > \pi^*(j_t)]$.
  Run halving algorithm to the set of experts.
```

*By halving algorithm, we know that the total number of mistakes is no larger than* $\log_2 |S_n|$*. In this case it is* $\log_2 n! = O(n \log n)$*.*

Halving algorithm is very intuitive and the upper bound is also very satisfying. However, the problem is that it assumes that there is a perfect experts that will not make any mistakes. This assumption is too strong and in most cases it is not satisfied. To handle this issue, in next section we introduction exponential weights algorithm. It removes the perfect expert assumption thus can generalize to more situations.

## 4.2 Exponential Weights Algorithm

In this section, we change the assumption. We no longer assume that there exists a perfect expert, and we introduce two new notations.

- $M_T(i) = \sum\limits_{i=1}^{T} \mathbf{1}[x_i^t \neq y^t]$
- $M_T(\mathcal{A}) = \sum\limits_{i=1}^{T} \mathbf{1}[\hat{y^t} \neq y^t]$

In the other words, $M_T(i)$ is the number of mistakes expert $i$ makes up to time $T$ while $M_T(\mathcal{A})$ is the number of mistakes the algorithm makes.

Since we have remove the assumption that there is an expert who will never make mistake, we need a smoother algorithm than halving algorithm because in halving algorithm we are zero-tolerant to any mistake. Before talking about exponential weighted algorithm, let's first talk about weighted majority algorithm.

**Algorithm 3** Weighted Majority Algorithm

```
Let $w_i^1 = 1$, for $i = 1, \cdots, N$.
Let $\epsilon \in (0,1)$ be a parameter we choose.
For $t = 1, 2, \cdots$
        Algorithm predicts $\hat{y^t} = round(\frac{\sum_{i=1}^{N} w_i^t x_i^t}{\sum_{i=1}^{N} w_i^t})$
        For $i = 1, \cdots, N$
            $w_i^{t+1} = w_i^t (1-\epsilon)^{\mathbf{1}[x_i^t \neq y^t]}$
```

Intuitively, different from halving algorithm, in weighted majority algorithm, if an expert makes a mistake, we will reduce its weight rather than remove it from the set. The expert will still have contribution to the algorithm but the weight is smaller. One can also see that halving algorithm is a special case of weighted majority algorithm when $\epsilon = 1$.

For this algorithm, we have the following theorem:

**Theorem 4.2.** *For any $\epsilon$ and expert $i$, no matter what the sequence $y^1, \cdots, y^T$ is, we can have that*

$$M_T(WMA) \leq \frac{2\log_e N}{\epsilon} + 2(1+\epsilon)M_T(i). \tag{4.1}$$

To prove this theorem, we will need the following lemma first:

**Lemma 4.3.** *The following inequalities are valid.*

1. $\log(1+x) \leq x$.
2. $1 + x \leq \exp(x)$.
3. $\exp(\alpha x) \leq 1 + (\exp(\alpha) - 1)x$, *for* $x \in [0,1]$.
4. $-\log(1+x) \leq -x + x^2$, *for* $x \in [-1, \frac{1}{2}]$.

There are many kinds of proof to this lemma online, thus we will omit the proof here. Next, we will go directly into the proof to this theorem.

*Proof.* Let $\Phi_t = \sum_{i=1}^{N} w_i^t$. Notice that $\Phi_1 = N$. Furthermore, since all weights are non-negative, we can have that $\Phi_{T+1} \geq w_i^{T+1} = (1-\epsilon)^{M_T(i)}$.

We now take a look at the case when $WMA$ makes a mistake at round $t$. In that case, we know that at least half of total weights are wrong. That is, at least $\frac{\sum_{i=1}^{N} w_i^t}{2}$ will shrink. By the updating formula of weights in WMA, we can have that

$$\begin{aligned}\Phi_{t+1} &\leq \frac{\sum\limits_{i=1}^{N} w_i^t}{2} + (1-\epsilon)\frac{\sum\limits_{i=1}^{N} w_i^t}{2} \\ &= (1-\frac{\epsilon}{2})\sum_{i=1}^{N} w_i^t \\ &= (1-\frac{\epsilon}{2})\Phi_t.\end{aligned}$$

Therefore, $\Phi_{T+1} \leq \Phi_0(1-\frac{\epsilon}{2})^{M_T(WMA)} = N(1-\frac{\epsilon}{2})^{M_T(WMA)}$.

Now, combine above inequalities and we can have that

$$(1-\epsilon)^{M_T(i)} \leq \Phi_{T+1} \leq N(1-\frac{\epsilon}{2})^{M_T(WMA)}.$$

Take log on both left side and right side we can get that

$$M_T(i)\log(1-\epsilon) \leq \log(N) + M_T(WMA)\log(1-\frac{\epsilon}{2}).$$

Use inequality 1 and 4 in the lemma and we get

$$M_T(i)(\epsilon+\epsilon^2) \geq -\log(N) + \frac{\epsilon}{2}M_T(WMA).$$

This directly indicates that

$$M_T(WMA) \leq \frac{2\log N}{\epsilon} + 2(1+\epsilon)M_T(i).$$

□

Since $i$ can take any value, we know that $M_T(WMA) \leq \frac{2\log N}{\epsilon} + 2(1+\epsilon)M_T(i^*)$ where $i^*$ is the best expert that makes least number of mistakes

Following this theorem, by setting $\epsilon = \sqrt{\frac{\log N}{M_T(i^*)}}$, we reach that

$$\begin{aligned}M_T(WMA) &\leq \frac{2\log N}{\sqrt{\frac{\log N}{M_T(i^*)}}} + 2(1+\sqrt{\frac{\log N}{M_T(i^*)}})M_T(i^*) \\ &= 2\sqrt{M_T(i^*)\log N} + 2M_T(i^*) + 2\sqrt{\frac{\log N}{M_T(i^*)}}M_T(i^*) \\ &= 2M_T(i^*) + 4\sqrt{M_T(i^*)\log N}.\end{aligned}$$

This concludes our discussion about weighted majority algorithm. We will now start discussing about exponential weighted algorithm. Before talking about the algorithm details, let's first introduce two new settings for this algorithm.

- Setting 1: Continuous Prediction
  - At time $t$, each expert $i$ predicts $x_i^t \in [0,1]$.

  - Algorithm predicts $\hat{y^t} \in [0,1]$.
  - Nature outcomes $y^t \in \{0,1\}$.
  - We have a convex loss function $l : [0,1] \times \{0,1\} \to \mathbb{R}$.
  - The loss of expert $i$ at $t$ is $l(x_i^t, y^t)$.
  - The loss of algorithm at $t$ is $l(\hat{y^t}, y^t)$.
  - Define $L_t(i) = \sum\limits_{s=1}^{t} l(x_i^s, y^s)$.
  - Define $L_t(Alg) = \sum\limits_{s=1}^{t} l(\hat{y}^s, y^s)$.

  We can see that $L_t(i) = L_{t-1}(i) + l(x_i^t, y^t)$. We can give some examples of loss functions.

  1. Absolute loss: $l(\hat{y}, y) = |\hat{y} - y|$.
  2. Square loss: $l(\hat{y}, y) = (\hat{y} - y)^2$.
  3. Log loss: $l(\hat{y}, y) = -y \log \hat{y} - (1-y)\log(1-\hat{y})$.

- Setting 2: Hedge/Action Setting
  - There are $N$ actions.
  - Algorithm must (randomly) select an action $i_t$ on day $t$.
  - This is equivalent that algorithm selects a distribution $\mathbf{p}^t \in \Delta_N$.
  - Nature chooses losses $l^t = [l_1^t, \cdots, l_N^t] \in [0,1]^N$, where $l_i^t$ means the cost of choosing $i$.
  - Then, the expected loss to this algorithm is defined as $p^t l^t = \mathbb{E}[l_i^t]$.
  - Define $L_t(i) = \sum\limits_{s=1}^{t} l_i^s$.
  - Define $L_t(Alg) = \sum\limits_{s=1}^{t} p^s l^s$.

To continue, we define the regret of an algorithm, which is the cost function that we want to minimize.

**Definition 4.4 (Regret).** *The regret of an algorithm is defined as*

$$Regret_T(Alg) = L_T(Alg) - \min_{i \in [N]} L_T(i). \tag{4.2}$$

Now we will give the algorithm details of Exponential Weights Algorithm:

**Algorithm 4** Exponential Weights Algorithm

```
Let $w_i^1 = 1$, for $i = 1, \cdots, N$. Choose $\eta > 0$.
For $t = 1, \cdots, T$
    For $i = 1, \cdots, N$
        $w_i^t = \exp(-\eta L_t(i))$
    Prediction Setting:
        Observe $x_1^t, \cdots, x_N^t$
        Output $\hat{y^t} = (\sum_{i=1}^{N} w_i^t x_i^t)/(\sum_{i=1}^{N} w_i^t)$
        Observed $y^t$.
    Hedge Setting:
        Output $\mathbf{p}^t = \frac{1}{\sum_{i=1}^{N} w_i^t}[w_1^t, \cdots, w_N^t]$
        Observed $l^t \in [0,1]^N$.
```

For prediction setting, we can see that it is the same as WMA algorithm except the way it updates the weights. We have the following theorem for Exponential Weights Algorithm:

**Theorem 4.5.** *For any sequence of inputs and any choice of* $\eta$*, we have that*

$$L_T(EWA) \leq \frac{\eta L_t(i) + \log N}{1 - \exp(-\eta)} \tag{4.3}$$

**Corollary 4.6.** *For an excellent choice of* $\eta$*, we have that*

$$L_T(EWA) - L_T(i) \leq \log N + 2\sqrt{L_T(i^*) \log N}. \tag{4.4}$$

*It means that the regret of algorithm is not larger than* $\log N + 2\sqrt{L_T(i^*) \log N}$*.*

Compare it with the result of weighted majority algorithm, we can see that EWA has much better performance when $M_T(i^*) \gg \log N$. Next, before proving the theorem, let's first prove a lemma that will be used.

**Lemma 4.7.** *Let* $X$ *be a random variable in* $[0, 1]$*, then* $\log(\mathbb{E}[\exp(sX)]) \leq (\exp(s) - 1)\mathbb{E}[X]$*.*

*Proof.* With the inequalities in Lemma 4.3, we can show that

$$\begin{aligned}
\log(\mathbb{E}[\exp(sX)]) &\leq \log(\mathbb{E}[1 + (\exp(s) - 1)X]) \\
&= \leq \log(1 + (\exp(s) - 1)\mathbb{E}[X]) \\
&\leq ((\exp(s) - 1)\mathbb{E}[X])
\end{aligned}$$

□

Next, let's prove theorem 4.5. We will prove the algorithm in Hedge setting, but the proof is almost the same for prediction setting.

*Proof.* First of all, let's define a random variable $X_t$ as $X_t = l(x_i^t, y^t)$ with probability $\frac{w_i^t}{\sum\limits_{i=1}^{N} w_j^t}$.

Then, like in the proof of WMA, we also define a function $\Phi_t = -\log(\sum\limits_{i=1}^{N} w_i^t)$. We can get that

$$
\begin{aligned}
\Phi_{t+1} - \Phi_t &= -\log(\frac{\sum\limits_{i=1}^{N} w_i^{t+1}}{\sum\limits_{i=1}^{N} w_i^t}) \\
&= -\log(\frac{\sum\limits_{i=1}^{N} w_i^t \exp(-\eta l(x_i^t, y^t))}{\sum\limits_{i=1}^{N} w_i^t}) \\
&= -\log(\sum_{i=1}^{N} \frac{w_i^t}{\sum\limits_{j=1}^{N} w_j^t} \exp(-\eta l(x_i^t, y^t))) \\
&= -\log(\mathbb{E}[\exp(-\eta X_t)]) \\
&\geq (1 - \exp(-\eta))\mathbb{E}[X_t] \\
&= (1 - \exp(-\eta))(\sum_{i=1}^{N} \frac{w_i^t}{\sum\limits_{j=1}^{N} w_j^t} l(x_i^t, y^t))) \\
(Jensen's\ Inequality) &\geq (1 - \exp(-\eta)) l(\sum_{i=1}^{N} \frac{w_i^t x_i^t}{\sum\limits_{j=1}^{N} w_j^t}, y^t) \\
&= (1 - \exp(-\eta)) l(\hat{y^t}, y^t).
\end{aligned}
$$

In addition, we know that $\Phi_1 = -\log N$ and $\Phi_{T+1} = -\log(\sum\limits_{i=1}^{N} \exp(-\eta L_T(i))) \leq \eta L_T(i)$ for any $i$. Therefore,

$$
\begin{aligned}
\log N + \eta L_T(i) &\geq \Phi_T - \Phi_1 \\
&= \sum_{t=1}^{T} (\Phi_{t+1} - \Phi_t) \\
&= \sum_{t=1}^{T} (1 - \exp(-\eta)) l(\hat{y^t}, y^t) \\
&= (1 - \exp(-\eta)) L_T(EWA).
\end{aligned}
$$

Therefore, we can have that for any $i$,

$$
L_T(EWA) \leq \frac{\log N + \eta L_T(i)}{1 - \exp(-\eta)}.
$$

□

From corollary 4.6, we know that by choosing $\eta$ carefully, we can have that $Regret_T(EWA) \leq \log N + 2\sqrt{L_T(i^*) \log N}$. If the loss function $l(x_i^s, y^s) \in [0, 1]$, it holds that

$$
\begin{aligned}
\frac{L_T(EWA) - L_T(i)}{T} &\leq \frac{1}{T}(\log N + 2\sqrt{L_T(i^*) \log N}) \\
&= O(\frac{1}{\sqrt{T}}).
\end{aligned}
$$

It shows that $\frac{Regret_T}{T}$ will goes to zero as $T \to \infty$.

Hedge setting and prediction setting belong to a category called full information setting. It means besides the choice algorithm makes, it also knows the loss of all alternative choices. Next we will talk about Perception Algorithms.

## 4.3 Perception

We now introduce a new setting called linear prediction setting:

- At time t, observe $x^t \in \mathbb{R}^d$ with $||x^t||_1 \leq 1$.
- Define linear predictor as a function $h_w(\cdot)$ parameterized by $w \in \mathbb{R}^d$ and $h_w(\cdot) = sign(w \cdot x)$.
- Predict $\hat{y^t} \in \{-1, 1\}$ using some linear predictor.
- Outcome $y^t \in \{-1, 1\}$.

We assume that for some $\gamma$, $\exists w$ such that $||w||_2 \leq 1$ and $(w \cdot x^t)y^t > \gamma$ for any $t$. This is equivalent to $\exists w$, such that $||w||_2^2 \leq \frac{1}{\gamma^2}$ and $(w \cdot x^t)y^t > 1$ for any $t$.

Intuitively, this assumption means that there exists a perfect linear predictor with respect to margin $\gamma$. Based on this assumption, we can start talking about perception algorithm.

**Algorithm 5** Perception

```
Let $w^1 = 0 \in \mathbb{R}^d$
For $t = 1, \cdots, T$
    $\hat{y^t} = sign(w^t \cdot x^t)$.
    Observe $y^t$.
    If $y^t(w^t \cdot x^t) > 0$, then $w^{t+1} = w^t$.
    Else, $w^{t+1} = w^t + x^t y^t$.
```

This algorithm is the same as gradient descent algorithm with loss function $l(w; (x, y)) = \max\{0, -(w \cdot x)y\}$, and

$$w^{t+1} = w^t - \nabla l(w^t; (x^t, y^t)).$$

For Perception Algorithm, the following theorem holds that

**Theorem 4.8.** *Let $M_T = \sum_{i=1}^{T} \mathbf{1}[y^t(w^t \cdot x^t) < 0]$. Assume $\exists w^*$ such that $||w^*|| \leq \frac{1}{\gamma}$ and $(w^* \cdot x^t)y^t \geq 1$, for any $t$. Then*

$$M_T \leq \frac{1}{\gamma^2}. \tag{4.5}$$

*Proof.* Suppose $w^*$ satisfies the assumption in theorem, then define $\Phi_t = ||w^* - w^t||^2$.

Notice that

$$\begin{aligned}\Phi_1 = ||w^* - w^1||^2 &= ||w^*|| \\ \leq \frac{1}{\gamma^2}.\end{aligned}$$

Therefore,

$$\begin{aligned}
\frac{1}{\gamma^2} &\geq \Phi_1 - \Phi_{T+1} \\
&= \sum_{t=1}^{T} (\Phi_t - \Phi_{t+1}) \\
&= \sum_{t=1}^{T} (||w^* - w^t||^2 - ||w^* - w^{t+1}||^2) \\
&= \sum_{t:\text{mistake}(t)} (||w^* - w^t||^2 - ||w^* - w^t - x^t y^t||^2) \\
&= \sum_{t:\text{mistake}(t)} 2(w^* - w^t)(x^t y^t) - (y^t)^2 ||x^t||^2 \\
&= \sum_{t:\text{mistake}(t)} 2(w^* - w^t)(x^t y^t) - ||x^t||^2 \\
&\geq \sum_{t:\text{mistake}(t)} 2(w^* - w^t)(x^t y^t) - 1 \\
(y^t(w^t \cdot x^t) < 0) &\geq \sum_{t:\text{mistake}(t)} 2(w^*)(x^t y^t) - 1 \\
(y^t(w^* \cdot x^t) \geq 1) &\geq \sum_{t:\text{mistake}(t)} 1 \\
&= M_T.
\end{aligned}$$

Therefore, we can get directly that $M_T \leq \frac{1}{\gamma^2}$. □

# Chapter 5

# Game Theory

In this chapter, we will talk about the basics of game theory. First of all, let's give some definitions in game theory.

**Definition 5.1 (Bimatrix Game).** *A two player bimatrix game is defined by matrix $M \in \mathbb{R}^{n\times m}$ and $N \in \mathbb{R}^{n\times m}$.*

*Each player (typically randomly) selects distribution $\mathbf{p} \in \Delta_n$, $\mathbf{q} \in \Delta_m$. If player 1 chooses $i \in [n]$ and player 2 chooses $j \in [m]$, then player 1 gets utility $U^1(i,j) = M_{ij}$ and player 2 gets utility $U^2(i,j) = N_{ij}$. Therefore, the utilities for their choices $\mathbf{p}$ and $\mathbf{q}$ is defined as an expectation*

$$\begin{aligned} U^1(p,q) &= \mathbb{E}[M_{ij}] = \mathbf{p}^T M\mathbf{q} \\ U^2(p,q) &= \mathbb{E}[N_{ij}] = \mathbf{p}^T N\mathbf{q} \end{aligned} \tag{5.1}$$

In addition, if $M + N = 0 \in \mathbb{R}^{n\times m}$, then the game is called zero-sum game.

Next, we will give a definition of Nash's equilibrium.

**Definition 5.2 (Nash's Equilibrium).** *Given $M$, $N \in \mathbb{R}^{n\times n}$, $\mathbf{p} \in \Delta_n$ and $\mathbf{q} \in \Delta_m$, $(\mathbf{p},\mathbf{q})$ is called a Nash equilibrium if the following two inequalities hold:*

$$\begin{aligned} \mathbf{p}^T M\mathbf{q} &\geq \tilde{\mathbf{p}}^T M\mathbf{q},\ \forall\tilde{\mathbf{p}} \in \Delta_n \\ \mathbf{p}^T N\mathbf{q} &\geq \mathbf{p}^T N\tilde{\mathbf{q}},\ \forall\tilde{\mathbf{q}} \in \Delta_m \end{aligned} \tag{5.2}$$

*Intuitively, it means that $(\mathbf{p},\mathbf{q})$ is a Nash equilibrium if and only if both players reach the best utility they can get at that point. That is, they cannot increase their utility by changing their strategy.*

We will next give two famous theorems in game theory.

**Theorem 5.3 (Nash's Theorem).** *For every bimatrix game $M$ and $N$, there always exists a Nash's equilibrium.*

*Proof.* The proof of this theorem relies on Brouwer's Fixed Pointer theorem.

In Brouwer's Fixed Point theorem, it states that if a function $f : \mathcal{K} \to \mathcal{K}$ is continuous, then $\exists\ x \in \mathcal{K}$ such that $f(x) = x$.

The proof of Nash's theorem is just to find correct $\mathcal{K}$ and $f$, then the fixed pointer $x$ is the equilibrium. □

**Theorem 5.4 (Maximum Theorem (Von Neumann)).** *The following equation holds for any zero sum game defined by $M \in \mathbb{R}^{n\times m}$*

$$\min_{\mathbf{p}\in\Delta_n} \max_{\mathbf{q}\in\Delta_m} \mathbf{p}^T M\mathbf{q} = \max_{\mathbf{q}\in\Delta_m} \min_{\mathbf{p}\in\Delta_n} \mathbf{p}^T M\mathbf{q}. \tag{5.3}$$

*Proof.* To prove this equality, we will prove that either side is no larger the other side. That is,

$$\begin{aligned}\min_{\mathbf{p}\in\Delta_n} \max_{\mathbf{q}\in\Delta_m} \mathbf{p}^T M\mathbf{q} &\le \max_{\mathbf{q}\in\Delta_m} \min_{\mathbf{p}\in\Delta_n} \mathbf{p}^T M\mathbf{q}\\ \min_{\mathbf{p}\in\Delta_n} \max_{\mathbf{q}\in\Delta_m} \mathbf{p}^T M\mathbf{q} &\ge \max_{\mathbf{q}\in\Delta_m} \min_{\mathbf{p}\in\Delta_n} \mathbf{p}^T M\mathbf{q}\end{aligned}$$

We will first prove $\ge$ which is easier than $\le$.

Let $\mathbf{p}_* = \arg\min_{\mathbf{p}\in\Delta_n} \max_{\mathbf{q}\in\Delta_m} \mathbf{p}^T M\mathbf{q}$ and $\mathbf{q}_* = \arg\max_{\mathbf{q}\in\Delta_m} \min_{\mathbf{p}\in\Delta_n} \mathbf{p}^T M\mathbf{q}$. Then we can have that

$$\begin{aligned}\min_{\mathbf{p}\in\Delta_n} \max_{\mathbf{q}\in\Delta_m} \mathbf{p}^T M\mathbf{q} &= \max_{\mathbf{q}\in\Delta_m} \mathbf{p}_*^T M\mathbf{q} \ge \mathbf{p}_*^T M\mathbf{q}_*\\ \max_{\mathbf{q}\in\Delta_m} \min_{\mathbf{p}\in\Delta_n} \mathbf{p}^T M\mathbf{q} &= \min_{\mathbf{p}\in\Delta_n} \mathbf{p}^T M\mathbf{q}_* \le \mathbf{p}_*^T M\mathbf{q}\end{aligned}$$

Combine the above two equations we can have that

$$\min_{\mathbf{p}\in\Delta_n} \max_{\mathbf{q}\in\Delta_m} \mathbf{p}^T M\mathbf{q} \ge \mathbf{p}_*^T M\mathbf{q}_* \ge \max_{\mathbf{q}\in\Delta_m} \min_{\mathbf{p}\in\Delta_n} \mathbf{p}^T M\mathbf{q}.$$

Next we continue to prove that $\min_{\mathbf{p}\in\Delta_n} \max_{\mathbf{q}\in\Delta_m} \mathbf{p}^T M\mathbf{q} \le \max_{\mathbf{q}\in\Delta_m} \min_{\mathbf{p}\in\Delta_n} \mathbf{p}^T M\mathbf{q}$.

To do so, we will utilize exponential weighted algorithm.

Imagine a repeated game:

```
For $t = 1, \cdots, T$
    $\mathbf{p}_t$ chosen by $P_1$.
    $\mathbf{q}_t$ chosen by $P_2$.
    $\mathbf{p}_t$ observes loss vector $l_t = M\mathbf{q}_t$.
    $\mathbf{q}_t$ observes loss vector $h_t = -M^T\mathbf{p}_t$.
    Players update to $p_{t+1}$, $q_{t+1}$ via EWA for these loss vectors.
```

Then, we can have

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T} \mathbf{p}_t^T M\mathbf{q}_t &= \frac{1}{T}\sum_{t=1}^{T} \mathbf{p}_t^T l_t\\ &= \min_{\mathbf{p}\in\Delta_n} \frac{1}{T}\sum_{t=1}^{T} \mathbf{p}^T l_t + \frac{Regret_T}{T}\\ &= \min_{\mathbf{p}\in\Delta_n} \frac{1}{T}\sum_{t=1}^{T} \mathbf{p}^T M\mathbf{q}_t + \epsilon_T^p\\ &= \min_{\mathbf{p}\in\Delta_n} \mathbf{p}^T M\bar{\mathbf{q}} + \epsilon_T^p\\ &\le \max_{\mathbf{q}\in\Delta_m} \min_{\mathbf{p}\in\Delta_n} \mathbf{p}^T M\mathbf{q} + \epsilon_T^p.\end{aligned}$$

Similarly,

$$\begin{aligned}
-\frac{1}{T}\sum_{t=1}^{T}\mathbf{p}_t^T M\mathbf{q}_t &= \frac{1}{T}\sum_{t=1}^{T}\mathbf{q}_t^T h_t \\
&= \min_{\mathbf{q}\in\Delta_m}\sum_{t=1}^{T}\frac{1}{T}\mathbf{q}^T h_t + \epsilon_T^q \\
&= -\max_{\mathbf{q}\in\Delta_m}\sum_{t=1}^{T}\frac{1}{T}\mathbf{p}^t M\mathbf{q} + \epsilon_T^q \\
&= -\max_{\mathbf{q}\in\Delta_m}\bar{\mathbf{p}}^T M\mathbf{q} + \epsilon_T^q \\
&\leq -\min_{\mathbf{p}\in\Delta_n}\max_{\mathbf{q}\in\Delta_m}\mathbf{p}^T M\mathbf{q} + \epsilon_T^q
\end{aligned}$$

Combine the above two inequalities we can get that

$$\min_{\mathbf{p}\in\Delta_n}\max_{\mathbf{q}\in\Delta_m}\mathbf{p}^T M\mathbf{q} - \epsilon_T^q \leq \max_{\mathbf{q}\in\Delta_m}\min_{\mathbf{p}\in\Delta_n}\mathbf{p}^T M\mathbf{q} + \epsilon_T^p$$

This indicates that $\min_{\mathbf{p}\in\Delta_n}\max_{\mathbf{q}\in\Delta_m}\mathbf{p}^T M\mathbf{q} \leq \max_{\mathbf{q}\in\Delta_m}\min_{\mathbf{p}\in\Delta_n}\mathbf{p}^T M\mathbf{q}$.

Therefore, we can conclude that $\min_{\mathbf{p}\in\Delta_n}\max_{\mathbf{q}\in\Delta_m}\mathbf{p}^T M\mathbf{q} = \max_{\mathbf{q}\in\Delta_m}\min_{\mathbf{p}\in\Delta_n}\mathbf{p}^T M\mathbf{q}$. □

**Corollary 5.5.** *$\bar{\mathbf{p}}_T$, $\bar{\mathbf{q}}_T$ are $\epsilon$ approximate to Nash equilibrium, where $\epsilon = \epsilon_T^q + \epsilon_T^p$. Therefore, the above proof also gives an algorithm to calculate Nash equilibrium.*

**Corollary 5.6.** *Minimax theorem can also directly proves Nash's theorem by setting the Nash equilibrium $(\mathbf{p}_*, \mathbf{q}_*)$ as following:*

$$\mathbf{p}_* = \arg\min_{\mathbf{p}\in\Delta_n}\max_{\mathbf{q}\in\Delta_m}\mathbf{p}^T M\mathbf{q}$$

$$\mathbf{q}_* = \arg\max_{\mathbf{q}\in\Delta_m}\min_{\mathbf{p}\in\Delta_n}\mathbf{p}^T M\mathbf{q}.$$

*We can then prove that*

$$\mathbf{p}_*^T(-M)\mathbf{q}_* \geq \tilde{\mathbf{p}}^T(-M)\mathbf{q}_*,\ \forall\tilde{\mathbf{p}}\in\Delta_n$$
$$\mathbf{p}_*^T M\mathbf{q}_* \geq \mathbf{p}_*^T M\tilde{\mathbf{q}},\ \forall\tilde{\mathbf{q}}\in\Delta_m.$$

That concludes our discussion about game theory.

# Chapter 6

# Boosting

In this chapter, we will talk about boosting, its definition and some theorems related to it. We will first give a definition of boosting.

**Definition 6.1** (**Boosting**)**.** *There are a set of data $\{(x_1, y_1), \cdots, (x_n, y_n)\} \subset X \times \{-1, 1\}$ and a class of weak learners $\mathcal{H}$. Each $h \in \mathcal{H}$ maps $X \to \{-1, 1\}$. These learners are "weak" means that typically, you should not expect to find a "good" $h \in \mathcal{H}$ for any particular problem.*

*The problem in boosting is that maybe we can combine many hs from $\mathcal{H}$ to get an ensemble function $F(x) = sign(\sum\limits_{i=1}^{n} q_i h_i(x))$ that has good performance on problem.*

To continue, let's define two conditions in boosting: Weak Lemma Condition and Strong Lemma Condition.

**Definition 6.2** (**Weak Lemma Condition**)**.** *The weak lemma condition states that for any distribution $\mathbf{p} \in \Delta_n$, these exists some $h \in \mathcal{H}$ such that*

$$\Pr_{(x_i, y_i) \sim p}[h(x_i) = y_i] = \sum_{i=1}^{n} p_i \mathbf{1}[h(x_i) = y_i] \geq \frac{1}{2} + \gamma. \tag{6.1}$$

In other words, it means that there exists a learner whose probability of being right is at least $\frac{1}{2} + \gamma$ for any distribution $\mathbf{p}$.

**Definition 6.3** (**Strong Lemma Condition**)**.** *The strong lemma condition states that $\exists \mathbf{q} \in \Delta(\mathcal{H})$ such that for every $(x, y)$ in sample,*

$$\sum_{h \in \mathcal{H}} \mathbf{q}(h) \cdot h(x_i) y_i > 0. \tag{6.2}$$

*That is, $F(x) = y$ for any data $(x, y)$ in sample.*

Intuitively, it means that there exists a way to combine all the weak learner, such that the ensemble function $F(x)$ can output the correct value for any data in the data set.

We will next use the theorems from game theory to prove a famous theorem in boosting:

**Theorem 6.4.** *Weak lemma condition is identical to strong lemma condition. That is, weak lemma condition is satisfied if and only if strong lemma condition is satisfied.*

Before going into the proof of this theorem, we will first try to transform WLC (Weak Lemma Condition) and SLC (Strong Lemma Condition) into a game theory format.

Let $M \in \mathbb{R}^{n\times m}$ be a matrix such that $M_{ij} = y_i h_j(x_i)$.

Then WLC means that $\forall \mathbf{p} \in \Delta_n$, $\exists i \in [m]$, such that $\mathbf{p}^T M e_i \geq 2\gamma$. That is

$$\min_{\mathbf{p}\in\Delta_n} \max_{j\in[m]} \mathbf{p}^T M e_j \geq 2\gamma.$$

Since $\mathbf{p}^T M$ is a vector, which means $\min\limits_{\mathbf{p}\in\Delta_n} \max\limits_{j\in[m]} \mathbf{p}^T M e_j = \min\limits_{\mathbf{p}\in\Delta_n} \max\limits_{\mathbf{q}\in\Delta_m} \mathbf{p}^T M \mathbf{q} \geq 2\gamma > 0$.

Similarly, SLC means that $\exists \mathbf{q} \in \Delta_m$, $\forall \mathbf{p} \in \Delta_n$, it has $\mathbf{p}^T M \mathbf{q} > 0$.

That is

$$\max_{\mathbf{q}\in\Delta_m} \min_{\mathbf{p}\in\Delta_n} \mathbf{p}^T M \mathbf{q} > 0.$$

By Minimax Theorem, we can know that these two statements are equivalent to each other.

Finally, in the last part of this chapter, we try to solve the problem that let $M$ be any payoff matrix bounded in $[0, 1] \in \mathbb{R}^{n\times m}$, we want to find an approximate min-max pair $\hat{\mathbf{p}}$, $\hat{\mathbf{q}}$ satisfying the following condition:

$$\max_{\mathbf{q}\in\Delta_m} \tilde{\mathbf{p}}^T M \mathbf{q} \leq V^* + \epsilon$$
$$\min_{\mathbf{p}\in\Delta_n} \mathbf{p}^T M \tilde{\mathbf{q}} \leq V^* - \epsilon$$

,

where $V^* = \min\limits_{\mathbf{p}\in\Delta_n} \max\limits_{\mathbf{q}\in\Delta_m} \mathbf{p}^T M \mathbf{q}$.

The reason we want to do so is the following theorem:

**Theorem 6.5.** *For boosting game, if we find a $\tilde{\mathbf{p}}$, $\tilde{\mathbf{q}}$, which are $\epsilon$-approximate Nash equilibrium, and $\epsilon < 2\gamma$, then $\forall$ $i$, it holds that*

$$F_{\tilde{\mathbf{q}}}(x_i) = y_i. \tag{6.3}$$

*Proof.* By assumption, we know that $\forall$ $i$,

$$e_i^T M \tilde{\mathbf{q}} \geq V^* - \epsilon.$$

By weak lemma condition, we can get that $V^* = \min\limits_{\mathbf{p}\in\Delta_n} \max\limits_{\mathbf{q}\in\Delta_m} \mathbf{p}^T M \mathbf{q} \geq 2\gamma$.

Therefore,

$$e_i^T M \tilde{\mathbf{q}} \geq 2\gamma - \epsilon > 0.$$

Equivalently, it means that $y_i \sum\limits_{j=1}^{m} \tilde{\mathbf{q}} h_j(x_i) > 0$ for any $(x_i, y_i)$.

Thus, $\forall$ $i$, $F_{\tilde{\mathbf{q}}}(x_i) = y_i$. □

After this, we try to solve this zero-sum game using exponential weights algorithm.

The following theorem holds for this algorithm:

```
For $t = 1, \cdots, T$
    For $i = 1, \cdots, n$
        $\mathbf{p}_t(i) = \exp(-\eta \sum_{s=1}^{t-1} e_i^T M\mathbf{q}_s)/Z.$
    For $j = 1, \cdots, m$
        $\mathbf{q}_t(j) = \exp(-\eta \sum_{s=1}^{t-1} \mathbf{p}_s^T Me_j)/Z'.$
Return $\mathbf{p}, \mathbf{q} = (\frac{1}{T}\sum_{t=1}^{T} \mathbf{p}_t, \frac{1}{T}\sum_{t=1}^{T} \mathbf{q}_t).$
```

**Theorem 6.6.** $\tilde{\mathbf{p}}, \tilde{\mathbf{q}}$ *are an $\epsilon$-approx Nash Equilibrium, where $\epsilon = \frac{Reg_T^{\mathbf{p}} + Reg_T^{\mathbf{q}}}{T}$.*

**Corollary 6.7.** *Consider new version with $\mathbf{q}_t = \arg\max_{\mathbf{q}\in\Delta_m} \mathbf{p}_t^T Mq$, which indicates that $Reg_T^{\mathbf{q}} \leq 0$. That means $\epsilon = \frac{Reg_T^{\mathbf{p}}}{T}$.*

Now we use this algorithm in boosting:

**Algorithm 6** In Boosting

```
For $t = 1, \cdots, T$
    For $i = 1, \cdots, n$
        $\mathbf{p}_t(i) = \exp(-\eta \sum_{s=1}^{t-1} y_i h_{js}(x_i))/Z.$
    $\mathbf{q}_t(j) = \arg\max_{e_j} \mathbf{p}^T Me_j.$
        $= \arg\max_{j} \sum_{i=1}^{n} \mathbf{p}_t(i) h_j(x_i) = e_{\text{Best Weak Learner at } t}.$
Return $\mathbf{p}, \mathbf{q} = (\frac{1}{T}\sum_{t=1}^{T} \mathbf{p}_t, \frac{1}{T}\sum_{t=1}^{T} \mathbf{q}_t).$
```

The last question in this chapter remained to be answered is how many rounds are needed to reach $\epsilon < 2\gamma$.

By theorem 4.5, we know that when it reaches $\epsilon < 2\gamma$, it has

$$\begin{aligned} \epsilon &= \frac{Reg_T^{\mathbf{p}}}{T} \\ &\leq \frac{\log N + \sqrt{T \log N}}{T} \\ &< 2\gamma. \end{aligned}$$

It means that it needs $T > \frac{\log N}{4\gamma^2}$ rounds.

That ends our discussion about boosting.

# Chapter 7

# Online Convex Optimization

In this chapter, we will talk about online convex optimization. We will focus on the definition of this problem, some algorithms to solve it and their complexities.

To begin with, let's talk about the framework used in online convex optimization.

## 7.1 Framework

- Given $\mathcal{K} \subset \mathbb{R}^d$, and $\mathcal{K}$ is bounded, convex and closed.
- At each round t, learner selects $x_t \in \mathcal{K}$.
- Nature reveals a convex function $f_t : \mathcal{K} \to \mathbb{R}$.
- Learner wants to minimize $Regret_T = \sum\limits_{t=1}^{T} f_t(x_t) - \min\limits_{x \in \mathcal{K}} \sum\limits_{t=1}^{T} f_t(x)$.

Let's see three examples about online convex optimization.

**Example 7.0.1 (Prediction with respect to expert advice).** *The prediction setting we talked about in online learning algorithm is an example of online convex optimization, where*

- $\mathcal{K} = \Delta_N$.
- $f_t(\tilde{w}) = l(\sum\limits_{i=1}^{N} \tilde{w}_i x_i^t, y^t)$.

**Example 7.0.2 (Online Portfolio Selection).** *In this setting, we have*

- $\mathcal{K} = \Delta_N$.
- $w^t$ *is the distribution of* $n$ *stocks fluctuations.*
- $r^t$ *is the vector of price, where* $r^t(i) = \frac{Price_t(stock_i)}{Price_{t-1}(stock_i)}$.
- $f_t(w^t) = -\log(w^t r^t)$.
- $\sum\limits_{t=1}^{T} f_t(w^t) = -\log(\Pi_{t=1}^{T}(w^t r^t))$.

- $\min\limits_{w\in\Delta_n} \sum\limits_{t=1}^{T} f_t(w) = -\log(\max\limits_{w\in\Delta_n} \Pi_{t=1}^{T}(wr^t))$ = *negative log wealth of best portfolio.*

*In this example, basically the loss function is minimized when money is put into the stock whose value will grow higher at time $t$.*

*In addition, there exists an algorithm such that for any sequence $r^1, \cdots, r^t$, we can have that*

$$\sum_{t=1}^{T} f_t(w^t) - \min_{w\in\Delta_n} \sum_{t=1}^{T} f_t(w) \leq n \log T.$$

**Example 7.0.3 (Online Regression).** *In this example, basically*

- $\mathcal{K}$ = *2-norm ball* $\Theta$.
- $f_t(\theta) = (\theta x_t - y_t)^2$, $(x_t, y_t) \in (\mathbb{R}^n \times \mathbb{R})$.

Online convex optimization is not an easy problem, it combines ideas from

- Optimization
- Statistical Learning

We have the following claim for (OCO) online convex optimization:

**Claim 7.1.** *OCO is harder than vanilla optimization.*

*Proof.* The goal of vanilla is to find the solution to $\arg\min\limits_{x\in\mathcal{K}} \Phi(x)$, where $\Phi(\cdot)$ is a convex function.

To prove this claim, let's take any OCO algorithm $\mathcal{A}$, and take the following procedures.

```
Initialize $x_1 \in \mathcal{K}$
For $t = 1, \cdots, T$:
        $x_t = \mathcal{A}(f_1, \cdots, f_{t-1})$
        Let $f_t(\cdot) = \Phi(\cdot)$ [or = $\langle \nabla\Phi(x_t), \cdot \rangle$]
Output $\frac{1}{T} \sum_{t=1}^{T} x_t = \bar{x_T}$.
```

By this algorithm, we can show that $\Phi(\bar{x_T}) - \min\limits_{x\in\mathcal{K}} \Phi(x) \leq \frac{Regret_T(\mathcal{A})}{T}$, since

$$\begin{aligned}
\Phi(\bar{x_T}) &\leq \frac{1}{T} \sum_{t=1}^{T} \Phi(x_t) \\
&= \frac{1}{T} \sum_{t=1}^{T} f_t(x_t) \\
&\leq \frac{1}{T} \min_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x_t) + \frac{Regret_T(\mathcal{A})}{T} \\
&\leq \frac{1}{T} \min_{x\in\mathcal{K}} \sum_{t=1}^{T} \Phi(x_t) + \frac{Regret_T(\mathcal{A})}{T} \\
&= \min_{x\in\mathcal{K}} \Phi(x_t) + \frac{Regret_T(\mathcal{A})}{T}.
\end{aligned}$$

□

Therefore, if we can solve online convex optimization problem, then we can also solve vanilla optimization.

Next, we will talk about Online-to-Batch Conversion, which use online optimization to solve batch minimization. The setting is following:

- $X$ - Data Space.
- $Y$ - Label Space.
- $D$ - Distribution over $X \times Y$.
- $\Theta \in \mathbb{R}^d$ - Bounded parameter space.
- $l : \Theta \times X \times Y \to \mathbb{R}$ - Loss function and convex in $\Theta$.

We can pick any OCO algorithm $\mathcal{A}$ and use it to generalize to batch minimization as following:

```
Initialize $\theta_1 \in \Theta$
For $t = 1, \cdots, T$:
    $\theta_t = \mathcal{A}(f_1, \cdots, f_{t-1})$
    Let $f_t(\cdot) = l(\cdot; (x_t, y_t))$
Output $\bar{\theta}_t = \frac{1}{T} \sum_{t=1}^{T} \theta_t$.
```

By this generalization, we can have that

**Theorem 7.2.** *Let risk* $R(\theta) = \mathbb{E}_{(x,y)\in D}[l(\theta; (x, y))]$, *then we can have*

$$\mathbb{E}[R(\bar{\theta_T})] - \min_{\theta\in\Theta} R(\theta) \leq \mathbb{E}[\frac{Regret_T(\mathcal{A})}{T}]. \tag{7.1}$$

Afterwards, we will introduce several algorithms in online gradient descent. The first one is called online gradient descent.

## 7.2 Online Convex Optimization Algorithm

```
Algorithm 7 Online Gradient Descent
Initialize $x_1 \in \mathcal{K}$
For $t = 1, \cdots, T$:
    $x_{t+1} = \Pi_{\mathcal{K}}(x_t - \eta_t \nabla f_t(x_t))$.
```

where $\Pi_{\mathcal{K}}(\cdot)$ is defined as the projection operator and $\Pi_{\mathcal{K}}(y) = \arg\min_{x\in\mathcal{K}} ||y - x||_2$.

This is a very simple algorithm similar to projected gradient algorithm, except that at each step, we use $f_t$ to calculate the gradient rather than a fixed $f$.

The following convergence theorem is associated with online gradient descent:

**Theorem 7.3.** *Assume that $f_t$ is G-Lipschitz ($||\nabla f_t|| \le G$), and $\forall x, y \in D$, $||x-y|| \le D$. Then, with $\eta_t = \frac{D}{G\sqrt{t}}$, we can get that*

$$Regret_T(OGD) \le \frac{3}{2} GD\sqrt{T}. \tag{7.2}$$

Before proving this theorem, we will first introduce a lemma about the projection operator.

**Lemma 7.4 (Pythagorean Theorem for Bregman Divergences).** *For any $x \in \mathcal{K}$,*

$$||x - \Pi_k(y)|| \le ||x-y||.$$

Then, let's start proving the convergence of OGD.

*Proof.* Let $x^* = \arg\min_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x)$, and $\nabla_t = \nabla f_t(x_t)$.

By the convexity of $f_t(\cdot)$, we know that

$$f_t(x_t) - f_t(x^*) \le \nabla_t^T (x_t - x^*).$$

By lemma 7.4 and the update function of OGD, we know that

$$\begin{aligned}
||x_{t+1} - x^*||^2 &= ||\Pi_{\mathcal{K}}(x_t - \eta_t \nabla f_t(x_t)) - x^*||^2 \\
&\le ||x_t - \eta_t \nabla f_t(x_t) - x^*||^2 \\
&= ||x_t - x^*||^2 + ||\eta_t \nabla f_t(x_t)||^2 - 2(\eta_t \nabla f_t(x_t))^T (x_t - x^*).
\end{aligned}$$

Therefore, we can get that

$$\begin{aligned}
f_t(x_t) - f_t(x^*) &\le \nabla_t^T (x_t - x^*) \\
&\le \frac{1}{2\eta_t}(||x_t - x^*||^2 - ||x_{t+1} - x^*||^2) + \frac{\eta_t}{2} |\nabla f_t(x_t)||^2 \\
&\le \frac{1}{2\eta_t}(||x_t - x^*||^2 - ||x_{t+1} - x^*||^2) + \frac{\eta_t G^2}{2}
\end{aligned}$$

Define $\frac{1}{\eta_0} = 0$, and by summing all terms we can have

$$\begin{aligned}
Reg_T &= \sum_{i=1}^{T} (f_t(x_t) - f_t(x^*)) \\
&\le \sum_{i=1}^{T} (\frac{1}{2\eta_t}(||x_t - x^*||^2 - ||x_{t+1} - x^*||^2) + \frac{\eta_t G^2}{2}) \\
&= \frac{1}{2} \sum_{i=1}^{T} ||x_t - x^*||^2 (\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) + \sum_{i=1}^{T} \frac{\eta_t G^2}{2} \\
&= \frac{||x_t - x^*||^2}{2} \sum_{i=1}^{T} (\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) + \frac{G^2}{2} \sum_{i=1}^{T} \eta_t \\
&= \frac{||x_t - x^*||^2}{2} \frac{1}{\eta_T} + \frac{G^2}{2} \sum_{i=1}^{T} \eta_t
\end{aligned}$$

By letting $\eta_t = \frac{D}{G\sqrt{t}}$, we can have that

$$\begin{aligned} Reg_T &\leq \frac{||x_t - x^*||^2}{2} \frac{1}{\frac{D}{G\sqrt{T}}} + \frac{G^2}{2} \sum_{i=1}^{T} \frac{D}{G\sqrt{t}} \\ &\leq \frac{DG\sqrt{T}}{2} + \frac{GD}{2} \sum_{i=1}^{T} \frac{1}{\sqrt{t}} \\ &\leq \frac{DG\sqrt{T}}{2} + GD\sqrt{T} \\ &= \frac{3}{2} GD\sqrt{T}. \end{aligned}$$

□

**Corollary 7.5.** *Gradient Descent algorithm for minimization one convex function $f(\cdot)$ with respect to averaging converge at a rate of $O(\frac{GD}{\sqrt{T}})$.*

It's surprise to find that the convergence rate of OGD is the same as normal gradient descent algorithm when they output the averaging.

Next, we will talk about Stochastic Gradient Descent, its setting and convergence rate. We will see that we can use OGD to prove its convergence rate easily.

In stochastic gradient descent, basically we want to minimize $\arg\min_{x \in \mathcal{K}} f(x)$, where $f(x) = \mathbb{E}_\xi[h(x;\xi)]$. The basic procedures in SGD is following:

- Sample $\xi_t$ from data set.
- $x_{t+1} \leftarrow x_t - \eta_t \nabla h(x_t; \xi_t)$.
- Output $\bar{x_t} = \frac{1}{T} \sum_{i=1}^{T} x_t$.

We can have the following claim about SDG:

**Claim 7.6.** *Stochastic Gradient Descent converges at $O(\frac{DG}{\sqrt{T}})$.*

*Proof.* Notice that

$$\begin{aligned} \mathbb{E}_{\xi_{1:t}}[f(\bar{x_T})] &\leq \frac{1}{T} \mathbb{E}_{\xi_{1:t}}[\sum_{t=1}^{T} f(x_t)] \\ &= \frac{1}{T} \mathbb{E}_{\xi_{1:t}}[\sum_{t=1}^{T} \mathbb{E}[f(x_t)|\xi_1, \cdots, \xi_{t-1}]] \\ &= \frac{1}{T} \mathbb{E}_{\xi_{1:t}}[\sum_{t=1}^{T} \mathbb{E}[\mathbb{E}_\xi[h(x_t;\xi_{1:t})]|\xi_1, \cdots, \xi_{t-1}]] \\ &= \frac{1}{T} \mathbb{E}_{\xi_{1:t}}[\sum_{t=1}^{T} \mathbb{E}[\mathbb{E}_\xi[h(x_t;\xi_t)]|\xi_1, \cdots, \xi_{t-1}]] \\ &= \frac{1}{T} \mathbb{E}_{\xi_{1:t}}[\sum_{t=1}^{T} \mathbb{E}[h(x_t;\xi_t)|\xi_1, \cdots, \xi_{t-1}]] \end{aligned}$$

Let $f_t(x) = h(x_t; \xi_t)$, then

$$\begin{aligned}
\mathbb{E}_{\xi_{1:t}}[f(\bar{x_T})] &\leq \frac{1}{T}\mathbb{E}_{\xi_{1:t}}[\sum_{t=1}^{T} \mathbb{E}[f_t(x_t)|\xi_1, \cdots, \xi_{t-1}]] \\
&= \mathbb{E}_{\xi_{1:t}}[\frac{1}{T}\sum_{t=1}^{T} f_t(x_t)] \\
&\leq \mathbb{E}_{\xi_{1:t}}[\frac{1}{T}\sum_{t=1}^{T} f_t(x^*)] + \frac{Regret_T(OGD)}{T} \\
&= f(x^*) + \frac{3DG}{2\sqrt{T}}.
\end{aligned}$$

□

Therefore, the convergence rate $\mathbb{E}_{\xi_{1:t}}[f(\bar{x_T})] - f(x^*) \leq \frac{3DG}{2\sqrt{T}}$.

We now conclude the three settings we talked about.

- $D$ - some distribution on data.
- $f(x, \xi)$ - a loss function of $x$ given data $\xi$.
- $F(x)$ - $\mathbb{E}_{\xi \in D}[f(x; \xi)]$.

Imagine now we have a sample $\hat{D} = \{\xi_{1:n}\}$ and the empirical loss function is defined as $\hat{F}(x) = \frac{1}{n}\sum_{i=1}^{n} f(x; \xi_i)$.

If the goal of learning is

$$\text{minimize } \hat{F}(x),$$

then its stochastic gradient descent.

If the goal of optimization is

$$\text{minimize } F(x),$$

then its online to batch minimization.

Their procedure is almost the same:

```
For $t = 1, \cdots, T$:
    Sample $\xi_t \sim D$ (or $\hat{D}$)
    $x_{t+1} = x_t - \eta_t \nabla f(x_t; \xi_t)$.
```

We have the following two claims for our algorithm.

**Claim 7.7.** *Let $\bar{x_T} = \frac{1}{T}\sum_{t=1}^{T} x_t$, then we have*

$$\mathbb{E}_{\xi_{1:n}}[F(\bar{x_T}) - F(x^*)] \leq O(\frac{GD}{\sqrt{T}}). \tag{7.3}$$

**Corollary 7.8.** *It is possible to get a high probability bound.*

**Claim 7.9.** *Let* $\bar{x_T} = \frac{1}{T}\sum\limits_{t=1}^{T} x_t$*, then we have*

$$\mathbb{E}_{\xi_{1:n}}[\hat{F}(\bar{x_T}) - \hat{F}(x^*)] \leq O(\frac{GD}{\sqrt{T}}). \tag{7.4}$$

We have proven claim 7.9 before, therefore, we will only prove claim 7.7 here.

*Proof.* We can see that

$$\begin{aligned}
\mathbb{E}_{\xi_{1:n}}[F(\bar{x_T}) - F(x^*)] &\leq \mathbb{E}_{\xi_{1:n}}[\frac{1}{T}\sum_{t=1}^{T} F(x_t) - F(x^*)] \\
&= \mathbb{E}_{\xi_{1:n}}[\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\xi\sim D}[f(x_t;\xi) - f(x^*;\xi)]] \\
&= \mathbb{E}_{\xi_{1:n}}[\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\xi_t}[f(x_t;\xi_t) - f(x^*;\xi_t)]] \\
&= \mathbb{E}_{\xi_{1:n}}[\frac{1}{T}\sum_{t=1}^{T} f(x_t;\xi_t) - f(x^*;\xi_t)] \\
&= \mathbb{E}[\frac{Regret_T}{T}] \\
&\leq \frac{3GD}{2\sqrt{T}}.
\end{aligned}$$

□

Note that the proof is not algorithm specific. That is, any low-regret algorithm can help us prove the theorem. The proof can guarantee the convergence of order $\mathbb{E}[\frac{Reg_T}{T}]$ for any algorithm, and $\mathbb{E}[\frac{Reg_T}{T}]$ will be small for a low-regret algorithm.

Another thing we need to consider is that we have shown that standard gradient descent converges at $O(\frac{GD}{\sqrt{T}})$, which is the same as SGD. Why is SGD better then?

The answer is that in each iteration, standard gradient descent will take $O(n)$ time while SGD will only use constant time. The time complexity in each iteration is the reason why SGD overtakes standard gradient descent.

Next, we will talk about a new online convex optimization algorithm, called Mirror Descent. The setting is following:

- Given $\mathcal{K} \subset \mathbb{R}^d$ and a regularizer $R(\cdot)$.
- $\mathcal{K} \subset int(dom(R))$
- $R$ is $\lambda$-strongly convex (w.r.t $||\cdot||$).
- Let $\nabla_t = \nabla f_t(x_t)$ at time $t$.

Then the algorithm is

**Algorithm 8** Mirror Descent

```
$x_1 \in \mathcal{K}$ arbitrary.
For $t = 1, \cdots, T$:
    $x_{t+1} = \arg\min_{x \in \mathcal{K}} \eta_t \langle \nabla_t, x \rangle + D_R(x, x_t)$
```

where $D_R(x, x_t)$ is the Bregman Divergence

$$D_R(x, y) = R(x) - R(y) - \langle \nabla R(y), x - y \rangle.$$

Since $R$ is $\lambda$-strongly convex, we know that $D_R(x, y) \geq \frac{\lambda}{2}||x - y||^2$.

Let's break down the update function of mirror descent. The term $\langle \nabla_t, x \rangle$ tends to move away from gradient. For example, if gradient $\nabla_t = [1, -1]$, then $x$ would be $[-\infty, \infty]$. To avoid infinity, we introduce $D_R(x, x_t)$, which will increase if $x$ moves too far away from $x_t$. Therefore, overall we can reach a balance of both parts.

Before proving its convergence, let's first introduce some lemmas.

**Lemma 7.10.** *For any $x, y, z \in \mathcal{K}$, it holds that*

$$D_R(z, x) + D_R(x, y) - D_R(z, y) = \langle \nabla R(y) - \nabla R(x), z - x \rangle. \tag{7.5}$$

*Proof.* By substituting $D_R(z, x)$ with the expression of Bregman Divergence, we can get the equality directly. □

**Lemma 7.11 (First-order Optimal Condition).** *If $x^* = \arg\min_{x \in \mathcal{K}} \Phi(x)$, then $\forall\ u \in \mathcal{K}$, we have*

$$\langle \nabla \Phi(x^*), u - x^* \rangle \geq 0. \tag{7.6}$$

Let's denote $\Phi(x) = \eta_t \langle \nabla_t, x \rangle + D_R(x, x_t)$ and apply to mirror descent. In mirror descent, we have $\nabla \Phi(x) = \eta_t \nabla_t + \nabla R(x) - \nabla R(x_t)$. Then by first-order optimal condition, we have that $\forall\ u \in \mathcal{K}$

$$\langle \eta_t \nabla_t + \nabla R(x_{t+1}) - \nabla R(x_t), u - x_{t+1} \rangle \geq 0.$$

**Lemma 7.12.** *Furthermore, notice that $\forall\ v, \gamma$, we have*

$$\langle v, \gamma \rangle \leq ||v|| \cdot ||\gamma||_* = (\frac{1}{\sqrt{\lambda}}||v||)(\sqrt{\lambda}||\gamma||_*) \leq \frac{||v||^2}{2\lambda} + \frac{||\gamma||_*^2 \lambda}{2}.$$

*The last inequality if Holder's inequality.*

**Lemma 7.13.** *The following inequality holds for any $u \in \mathcal{K}$,*

$$\eta_t(f_t(x_t) - f_t(u)) \leq \eta_t \langle \nabla_t, x_t - u \rangle \leq D_R(u, x_t) - D_R(u, x_{t+1}) + \frac{\eta_t^2}{2\lambda}||\nabla_t||_*^2.$$

*Proof.* The first inequality $\eta_t(f_t(x_t) - f_t(u)) \leq \eta_t \langle \nabla_t, x_t - u \rangle$ is indicated by the convexity of $f$. Now let's look at the second inequality.

$$\begin{aligned} \eta_t \langle \nabla_t, x_t - u \rangle =& \langle \nabla R(x_t) - \nabla R(x_{t+1}) - \eta_t \nabla_t, u - x_{t+1} \rangle \\ &- \langle \nabla R(x_t) - \nabla R(x_{t+1}), u - x_{t+1} \rangle \\ &+ \langle \eta_t \nabla_t, x_t - x_{t+1} \rangle. \end{aligned}$$

By Lemma 7.11, we know that $\langle \nabla R(x_t) - \nabla R(x_{t+1}) - \eta_t \nabla_t, u - x_{t+1}\rangle \leq 0$. Therefore, we have

$$\begin{aligned}
\eta_t \langle \nabla_t, x_t - u\rangle \leq & - \langle \nabla R(x_t) - \nabla R(x_{t+1}), u - x_{t+1}\rangle + \langle \eta_t \nabla_t, x_t - x_{t+1}\rangle \\
\text{(Lemma 7.10)} = & D_R(u, x_t) - D_R(u, x_{t+1}) - D_R(x_{t+1}, x_t) + \langle \eta_t \nabla_t, x_t - x_{t+1}\rangle \\
\text{(Lemma 7.12)} \leq & D_R(u, x_t) - D_R(u, x_{t+1}) - D_R(x_{t+1}, x_t) + \frac{||\eta_t \nabla_t||_*^2}{2\lambda} + \frac{||x_t - x_{t+1}||^2 \lambda}{2} \\
(\lambda - \text{Convexity of } D) \leq & D_R(u, x_t) - D_R(u, x_{t+1}) + \frac{\eta_t^2 ||\nabla_t||_*^2}{2\lambda}.
\end{aligned}$$

□

After all these lemma, we come to this theorem

**Theorem 7.14.** *For Mirror Descent, for any $u \in \mathcal{K}$, let $\eta_1, \cdots, \eta_t$ be decreasing, and let $d^2 = \max_{t=1,\cdots T} D_R(u, x_t)$, then we have*

$$\sum_{t=1}^{T} (f_t(x_t) - f_t(u)) \leq \frac{d^2}{\eta_T} + \frac{1}{2\lambda} \sum_{t=1}^{T} \eta_t ||\nabla_t||_*^2. \tag{7.7}$$

**Corollary 7.15.** *Let $\eta_t = \frac{d\sqrt{\lambda}}{\sqrt{\sum_{s=1}^{t} ||\nabla_s||_*^2}}$, then*

$$Regret_T(MD) \leq 2\frac{d}{\sqrt{\lambda}} \sqrt{\sum_{t=1}^{T} ||\nabla_t||_*^2}.$$

*Proof.* By Lemma 7.13, we have that

$$\begin{aligned}
\sum_{t=1}^{T} (f_t(x_t) - f_t(u)) \leq & \sum_{t=1}^{T} \frac{1}{\eta_t} (D_R(u, x_t) - D_R(u, x_{t+1}) + \frac{\eta_t^2 ||\nabla_t||_*^2}{2\lambda}) \\
= & \frac{1}{\eta_1} D_R(u, x_1) - \frac{1}{\eta_T} D_R(u, x_{t+1}) + \sum_{t=1}^{T-1} (\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}) D_R(u, x_{t+1}) + \sum_{t=1}^{T} \frac{\eta_t^2 ||\nabla_t||_*^2}{2\lambda} \\
\leq & \frac{1}{\eta_1} d^2 + \sum_{t=1}^{T-1} (\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}) D_R(u, x_{t+1}) + \sum_{t=1}^{T} \frac{\eta_t^2 ||\nabla_t||_*^2}{2\lambda} \\
\leq & \frac{1}{\eta_1} d^2 + d^2 \sum_{t=1}^{T-1} (\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}) + \sum_{t=1}^{T} \frac{\eta_t^2 ||\nabla_t||_*^2}{2\lambda} \\
\leq & \frac{d^2}{\eta_1} + \frac{d^2}{\eta_T} - \frac{d^2}{\eta_1} + \sum_{t=1}^{T} \frac{\eta_t^2 ||\nabla_t||_*^2}{2\lambda} \\
= & \frac{d^2}{\eta_T} + \sum_{t=1}^{T} \frac{\eta_t^2 ||\nabla_t||_*^2}{2\lambda}
\end{aligned}$$

□

Online mirror descent can extend to many other online convex optimization algorithms, let's see two examples.

**Example 7.15.1.** *Let $\mathcal{K}$ be any convex set, and $R(x) = \frac{1}{2}||x||^2$. By the update function of OMD, we can get that*

$$\begin{aligned} x_{t+1} &= \arg\min_{x \in \mathcal{K}} \eta_t \langle x, \nabla_t \rangle + D_R(x, x_t) \\ &= \Pi_{\mathcal{K}}(x_t - \eta \nabla_t) \end{aligned}$$

*which is online gradient descent.*

**Example 7.15.2.** *Let $K = \Delta_n$ and $R(x) = \sum\limits_{i=1}^{n} x_i \log x_i$. Then*

$$x_{t+1} = \arg\min_{x \in \mathcal{K}} \eta_t \langle x, \nabla_t \rangle + \sum_{i=1}^{n} x_i \log \frac{x_i}{x_t(i)}.$$

*By first order optimal condition, we can have that $(\eta_t \nabla_t)_i = \log \frac{x(i)}{x_t(i)}$, which means*

$$x(i) = x_t(i) \exp(-\eta_t \nabla_t(i))/Z.$$

*It turns out that it is just exponential weights algorithm.*

*Why do we choose this function $R$? Because entropy function is 1-strongly convex with respect to $|| \cdot ||_1$. Then $d^2 = \max\limits_t D_R(u, x_t) \leq \log n$ and*

$$Regret_T \leq \log n \sqrt{\sum_{t=1}^{T} ||\nabla_t||_\infty^2}.$$

Finally, we will talk about the last algorithm in online convex optimization: Follow the Regularized Leader algorithm.

**Algorithm 9** Follow the Regularized Leader

```
$x_1 \in \mathcal{K}$ arbitrary.
Regularizer $R(x)$
For $t = 1, \cdots, T$:
        (V1): $x_{t+1} = \arg\min_{x \in \mathcal{K}} \eta \sum_{s=1}^{t} f_s(x) + R(x)$
        (V2): $x_{t+1} = \arg\min_{x \in \mathcal{K}} \eta \sum_{s=1}^{t} \langle \nabla f_s(x_s), x \rangle + R(x)$
```

**Fact 7.15.1.**

1. *If $f_s(x)$ is linear in $x$, then FTL can have linear regret.*
2. *The second version of FTRL is just OMD al long as $x_t \in int(\mathcal{K})$.*

Next we will prove the second fact.

*Proof.* If $x_t \in int(\mathcal{K})$, then for FTRL we have

$$\nabla R(x_{t+1}) = -\eta \sum_{s=1}^{t} \nabla f_s(x_s).$$

For OMD, we have

$$\begin{aligned}\nabla R(x_{t+1}) &= \nabla R(x_t) - \eta \nabla_t f_t(x_t) \\ &= \nabla R(x_{t-1}) - \eta(\nabla_t f_t(x_t) + \nabla_t f_{t-1}(x_{t-1})) \\ &\cdots \\ &= -\eta \sum_{s=1}^{t} \nabla f_s(x_s).\end{aligned}$$

□

Therefore, OMD is just FTRL when $x \in \mathcal{K}$. This also shows that the convergence rate of FTRL is as good as that of OMD.

This concludes our discussion about online convex optimization. Next, we will discuss Multi-Armed Bandit, which tries to balance exploration and exploitation.

# Chapter 8

# Multi-Armed Bandit

In "full information" setting, in addition to the strategy the player takes, he can also computes the loss of alternative action. This is not the case in Multi-Armed Bandit setting. In Bandit, feedback is only limited to the selected action that user takes. The protocol is following:

```
Suppose that there are $n$ actions.
For $t = 1, \cdots, T$:
    Algorithm selects $p^t \in \Delta_n$.
    Nature chooses $l^t \in [0,1]^n$.
    Algorithm samples $i_t \sim p^t$.
    Observe only $l_{it}^t$ and update based on it.
```

The Regret is defined as $Regret_T = \sum_{t=1}^{T} l_{it}^t - \min_{i=1,\cdots,n} \sum_{t=1}^{T} l_i^t$. Since $i_t$ is a random variable sampled from distribution $p^t$, the regret is also a random variable. Therefore, in this setting we care about the expectation of regret as

$$\mathbb{E}[Regret_T] = \mathbb{E}[\sum_{t=1}^{T} l_{it}^t - \min_{i=1,\cdots,n} \sum_{t=1}^{T} l_i^t].$$

This expectation is taken over all randomness in the algorithm.

Next, we will describe an algorithm that could achieve good regret bound called EXP3 Algorithm.

**Algorithm 10** EXP 3

```
Initialize $w_1^1, \cdots, w_n^1 = 1$
For $t = 1, \cdots, T$:
    $p^t = \frac{w^t}{||w^t||_1}$
    Sample $i_t \sim p$, observe $l_{it}^t$.
    $\hat{l}^t = [0, 0, 0, \cdots, \frac{l_{it}^t}{p_i^t}, 0, \cdots, 0]$.
    For any $i$, $w_i^{t+1} = w_i^t \exp(-\eta \hat{l}_i^t)$.
```

We can see that intuitively, this algorithm looks very similar with Exponential Weights Algorithm in Hedge's setting, except that since we have no idea about the loss of alternative choice, we set these losses to zero. Therefore, we will only update the weight of our choice at one time. This is also the difference between full information setting and bandit setting.

One thing that should be noticed is that $\hat{l}^t$ is an unbiased estimator of $l^t$.

$$\begin{aligned}\mathbb{E}_{i_t \sim p^t}[\hat{l}^t] &= \sum_{i=1}^{n} p_i^t \cdot [0,0,0,\cdots,\frac{l_{it}^t}{p_i^t},0,\cdots,0] \\ &= \sum_{i=1}^{n}[0,0,0,\cdots,l_{it}^t,0,\cdots,0] \\ &= [l_{1t}^t,\cdots,l_{it}^t,\cdots,l_{nt}^t] \\ &= l^t.\end{aligned} \tag{8.1}$$

In addition, the covariance matrix of $\hat{l}^t$ is

$$CoVar(\hat{l}^t) = O(diag(\frac{1}{p_1^t},\cdots,\frac{1}{p_n^t})). \tag{8.2}$$

Before proving the theorem, lets first proving a lemma that would be used.

**Lemma 8.1.** *Let $X$ be a random variable and $X \in [0,1]$, then*

$$\log \mathbb{E}[exp(-sX)] \leq -s\mathbb{E}[X] + \frac{s^2}{2}\mathbb{E}[X]. \tag{8.3}$$

*Proof.* Let $X$ be a random variable $\geq 0$, then we have the following two inequalities.

1. $\log(1+X) \leq X$.
2. $\exp(-sX) \leq -sX + \frac{s^2}{2}X^2 + 1$

Therefore,

$$\begin{aligned}\log(\mathbb{E}[\exp(-sX)]) &\leq \log(\mathbb{E}[1 - sX + \frac{s^2}{2}X^2]) \\ &= \log(1 + \mathbb{E}[-sX + \frac{s^2}{2}X^2]) \\ &\leq \mathbb{E}[-sX + \frac{s^2}{2}X^2] \\ &= -s\mathbb{E}[X] + \frac{s^2}{2}\mathbb{E}[X^2].\end{aligned}$$

□

Then, we have the following theorem about the convergence of EXP3 Algorithm.

**Theorem 8.2.** *For any $i$, it has*

$$\mathbb{E}[\sum_{t=1}^{T} p^t \cdot l^t - \sum_{t=1}^{T} l_i^t] \leq \frac{\log n}{\eta} + \frac{\eta}{2}Tn. \tag{8.4}$$

Since we sample $i_t$ randomly from $p^t$, then the expectation of $l_{it}$ is

$$\mathbb{E}[l_{it}] = \sum_{i=1}^{n} p_i l_i^t = p^t \cdot l^t.$$

Therefore, for EXP 3, it means that $Regret_T \leq \frac{\log n}{\eta} + \frac{\eta}{2} Tn$.

**Corollary 8.3.** *For $\eta = \sqrt{\frac{2 \log n}{T}}$, we have that $\mathbb{E}[Regret_T] \leq \sqrt{2Tn \log n}$.*

This regret bound is almost the same as EWA algorithm where we have full information.

*Proof.* Similar to the proof of EWA algorithm, let $\Phi_t = -\frac{1}{\eta} \log(\sum\limits_{i=1}^{n} w_i^t)$.

Notice that

$$\begin{aligned}
\Phi_{t+1} - \Phi_t &= -\frac{1}{\eta} \log(\frac{\sum\limits_{i=1}^{n} w_i^{t+1}}{\sum\limits_{i=1}^{n} w_i^t}) \\
&= -\frac{1}{\eta} \log(\frac{\sum\limits_{i=1}^{n} w_i^t \exp(-\eta \hat{l}_i^t)}{\sum\limits_{i=1}^{n} w_i^t}).
\end{aligned}$$

Let $X$ be a random variable and $P(X = \hat{l}_i^t) = \frac{w_i^t}{\sum\limits_{i=1}^{n} w_i^t}$. Next, we can have that

$$\begin{aligned}
\Phi_{t+1} - \Phi_t &= -\frac{1}{\eta} \log(\mathbb{E}[\exp(-\eta X)]) \\
&\geq -\frac{1}{\eta}(-\eta \mathbb{E}[X] + \frac{\eta^2}{2} \mathbb{E}[X]) \\
&= \mathbb{E}[X] - \frac{\eta}{2} \mathbb{E}[X^2] \\
&= \sum_{i=1}^{n} \frac{w_i^t}{\sum\limits_{j=1}^{n} w_j^t} \hat{l}_i^t - \frac{\eta}{2} \sum_{i=1}^{n} \frac{w_i^t}{\sum\limits_{j=1}^{n} w_j^t} (\hat{l}_i^t)^2 \\
&= p^t \hat{l}^t - \frac{\eta}{2} \sum_{i=1}^{n} p_i^t (\hat{l}_i^t)^2.
\end{aligned}$$

The above steps are quite similar to that of EWA algorithm. Next, we will try to estimate $\mathbb{E}[\Phi_{t+1} - \Phi_t]$.

Following above deduction, we have that

$$\begin{aligned}
\mathbb{E}_{i_t \sim p_t}[\Phi_{t+1} - \Phi_t | i_1, \cdots, i_{t-1}] &\geq \mathbb{E}_{i_t \sim p_t}[p^t \hat{l}^t - \frac{\eta}{2} \sum_{i=1}^{n} p_i^t (\hat{l}_i^t)^2] \\
(p_i = 0 \text{ for } i \neq i_t) &\geq \mathbb{E}_{i_t \sim p_t}[p^t \hat{l}^t - \frac{\eta}{2} p_{it}^t (\hat{l_{it}^t})^2] \\
&\geq \mathbb{E}_{i_t \sim p_t}[p^t \hat{l}^t - \frac{\eta}{2} p_{it}^t (\frac{l_{it}^t}{p_{it}^t})^2] \\
(\text{Unbaised Estimator}) &\geq p^t \cdot l^t - \mathbb{E}_{i_t \sim p_t}[\frac{\eta}{2} p_{it}^t (\frac{l_{it}^t}{p_{it}^t})^2] \\
&\geq p^t \cdot l^t - \sum_{i_t=1}^{n} p_{it}^t [\frac{\eta}{2} p_{it}^t (\frac{l_{it}^t}{p_{it}^t})^2] \\
&\geq p^t \cdot l^t - \frac{\eta}{2} \sum_{i_t=1}^{n} (l_{it}^t)^2] \\
(l_{it}^t \leq 1) &\geq p^t \cdot l^t - \frac{\eta n T}{2}.
\end{aligned}$$

After all these steps, let consider

$$\begin{aligned}
\mathbb{E}[\Phi_{T+1} - \Phi_1] &= \mathbb{E}[\sum_{t=1}^{T} (\Phi_{t+1} - \Phi_t)] \\
&= \mathbb{E}[\sum_{t=1}^{T} \mathbb{E}_{i_t \sim p_t}[\Phi_{t+1} - \Phi_t | i_1, \cdots, i_{t-1}]] \\
&\geq \mathbb{E}[\sum_{t=1}^{T} p^t \cdot l^t - \frac{\eta n}{2}]
\end{aligned}$$

Then, we need to bound $\Phi_{T+1} - \Phi_1$. Notice that

$$\begin{aligned}
\Phi_{T+1} - \Phi_1 &= -\frac{1}{\eta} \log(\sum_{i=1}^{n} \exp(-\eta \sum_{t=1}^{T} \hat{l_i^t})) + \frac{1}{\eta} \log n \\
&\leq -\frac{1}{\eta} \log(\exp(-\eta \sum_{t=1}^{T} \hat{l_*})) + \frac{1}{\eta} \log n \\
&\leq \sum_{t=1}^{T} \hat{l_*} + \frac{1}{\eta} \log n.
\end{aligned}$$

Therefore,

$$\mathbb{E}[\Phi_{T+1} - \Phi_1] \leq \mathbb{E}[\sum_{t=1}^{T} \hat{l_*} + \frac{1}{\eta} \log n].$$

Combine above, we can then get that

$$\sum_{t=1}^{T} p^t \cdot l^t - \frac{\eta n T}{2} \leq \sum_{t=1}^{T} \hat{l_*} + \frac{1}{\eta} \log n.$$

Since $\sum_{t=1}^{T} p^t \cdot l^t = \mathbb{E}[l_{it}^t]$, then

$$\mathbb{E}[l_{it}^t] - \sum_{t=1}^{T} \hat{l_*} \leq \frac{\eta n T}{2} + \frac{1}{\eta} \log n.$$

Therefore, we can get that

$$\mathbb{E}[Regret_T] \leq \frac{\eta n T}{2} + \frac{1}{\eta} \log n.$$

□

**Remark 8.4.** *The bound $O(\sqrt{Tn \log n})$ is not the best possible bound for EXP 3. If we can combine Tsallis entropy with mirror descent, we can reach a smaller bound as $O(\sqrt{Tn})$.*

Now that we have finished the discussion about multi-armed bandit, let's go deeper into the setting of Stochastic Bandits Setting. The setting is following:

- We have $n$ distributions $D_1, \cdots, D_n$, each with $\mathbb{E}_{X \sim D_i}[X] = \mu_i$.
- Assume these distribution are subgaussian with proxy variance 1 and $\mu_i - \mu_j < 1$.
- For each round $t$, algorithm picks $i_t \in [n]$, and then observes gain $x_{it}^t \sim D_{it}$.
- Let $i^* = \arg\max_i \mu_i$.
- Define the regret at time $T$ as $Regret_T = \sum_{t=1}^{T} (\mu_{i*} - x_{it}^t)$.
- Assume without loss of generality, $i^* = 1$.
- Let $\Delta_i = \mu_1 - \mu_i$, for $i = 2, \cdots, n$.
- The expected regret of some algorithm choosing $i_1, \cdots, i_T$ is then

$$\begin{aligned}\mathbb{E}[Regret_T] &= \mathbb{E}[\sum_{t=1}^{T} (\mu_1 - \mu_{it})] \\ &= \mathbb{E}[\sum_{i=1}^{n} N_i^T \Delta_i]\end{aligned}$$

  where $N_i^t = \sum_{s=1}^{t} \mathbf{1}[i_s = i]$, which is the number of times bandit $i$ is chosen.

Notice that this setting is very similar to the previous multi-armed bandit setting. Actually, it is a special case of multi-armed bandit setting, where the Nature chooses the loss (in this case, gain) vector based on $n$ subgaussian distributions.

Let's see a simple algorithm in this setting first.

**Algorithm 11** Simple Algorithm in Stochastic Bandit Setting

```
Assume $\Delta_* = \min\limits_{i=2,\cdots,n} \Delta_i$ is known.
Let $K = \lceil \frac{4\log(nT)}{\Delta_*^2} \rceil$
For $t = 1, \cdots, T$:
        If $t \in [(i-1)K, iK]$
                $i_t = i$ (explore)
        Else
                $i_t = \arg\min\limits_i \hat{\mu_i}$, where $\hat{\mu_i} = \frac{1}{K} \sum\limits_{t=(i-1)K}^{iK} x_i^t$. (exploit)
```

Basically, what this algorithm does is that it will first try each bandit $K$ times (total $nK$ times). Afterwards, it will calculate the average of gain from each bandit, and stick to the best gain bandit afterwards.

One problem of this algorithm is that it assumed that $\Delta_* = \min\limits_{i=2,\cdots n} \Delta_i$ is known, which is not realistic in most cases. Before we go into a better algorithm that can remove this assumption, let's first give a claim about this algorithm.

**Claim 8.5.** *The expectation of regret of this simple algorithm is bounded.*

$$\mathbb{E}[Regret_T(Simple\ Algorithm)] \leq \sum_{i=1}^{n} \frac{4\Delta_i \log Tn}{\Delta_*^2} + O(1). \tag{8.5}$$

*Proof.* Let's consider two cases:

1. For $t > nK$, $i_t = 1$ (Find the best bandit after $nK$).
2. For $t > nK$, $i_t \neq 1$ (Not find the best bandit after $nK$).

Then the expectation of regret is bounded by

$$\begin{aligned}
\mathbb{E}[Regret_T] &= \mathbb{E}[\sum_{i=2}^{n} N_i^T \Delta_i] \\
&\leq \mathbb{E}[\sum_{i=2}^{n} N_i^T \Delta_i | Found] \Pr(Found) + \mathbb{E}[\sum_{i=2}^{n} N_i^T \Delta_i | Not\ Found] \Pr(Not\ Found) \\
&\leq (K \sum_{i=2}^{n} \Delta_i) \Pr(Found) + \mathbb{E}[\sum_{i=2}^{n} N_i^T \Delta_i | Not\ Found] \Pr(Not\ Found) \\
&\leq K \sum_{i=2}^{n} \Delta_i + \mathbb{E}[\sum_{i=2}^{n} N_i^T \Delta_i | Not\ Found] \Pr(Not\ Found) \\
&\leq K \sum_{i=2}^{n} \Delta_i + T \Pr(Not\ Found) \\
&\leq \sum_{i=2}^{n} \frac{4\Delta_i \log(nT)}{\Delta_*^2} + 1 + T \Pr(Not\ Found)
\end{aligned}$$

What remains to do is to bound $\Pr(Not\ Found)$. We will utilize Hoeffding Inequality in our proof. Notice that

$$
\begin{aligned}
\Pr(\textit{Not Found}) &= \Pr(\exists i \in [2, \cdots, n], \hat{\mu_i} > \hat{\mu_1}) \\
&\leq \sum_{i=2}^{n} \Pr(\hat{\mu_i} > \hat{\mu_1}) \\
&\leq \sum_{i=2}^{n} \Pr(\hat{\mu_i} - \mu_i \geq \frac{\Delta_i}{2} \textit{ or } \mu_1 - \hat{\mu_i} \geq \frac{\Delta_i}{2}) \\
&\leq \sum_{i=2}^{n} (\Pr(\hat{\mu_i} - \mu_i \geq \frac{\Delta_i}{2}) + \Pr(\mu_1 - \hat{\mu_i} \geq \frac{\Delta_i}{2})) \\
&\leq \sum_{i=2}^{n} (\Pr(\hat{\mu_i} - \mu_i \geq \frac{\Delta_*}{2}) + \Pr(\mu_1 - \hat{\mu_i} \geq \frac{\Delta_*}{2})) \\
&\leq \sum_{i=2}^{n} (\exp(-\frac{2K^2\Delta_*^2}{4K}) + \exp(-\frac{2K^2\Delta_*^2}{4K})) \\
&\leq \sum_{i=2}^{n} 2(\exp(-\frac{K\Delta_*^2}{2}) \\
&\leq 2(n-1)\exp(-\frac{K\Delta_*^2}{2}) \\
&\leq 2(n-1)\exp(-\frac{\frac{4\log(nT)}{\Delta_*^2}\Delta_*^2}{2}) \\
&\leq 2(n-1)\exp(\log(\frac{1}{n^2T^2})) \\
&= 2(n-1)(\frac{1}{n^2T^2}) \\
&\leq \frac{1}{T}.
\end{aligned}
$$

Therefore, we can have that

$$
\mathbb{E}[\textit{Regret}_T(\text{Simple Algorithm})] \leq \sum_{i=2}^{n} \frac{4\Delta_i \log(nT)}{\Delta_*^2} + 1 + T\Pr(\textit{Not Found}) \leq \sum_{i=1}^{n} \frac{4\Delta_i \log Tn}{\Delta_*^2} + O(1)
$$

□

This finishes our discussion about this simple algorithm. Again, it assumes that $\Delta_*$ is known, which cannot be reached in most cases. To remove this assumption. let's see an advanced algorithm called UCB (upper common bound algorithm).

**Algorithm 12** Upper Common Bound Algorithm

```
Let $N_i^t = \sum_{s=1}^{t-1} \mathbf{1}[I_s = i]$ and $\hat{\mu_i}^t = \frac{1}{N_i^t}[\sum_{s=1}^{t-1} \mathbf{1}[I_s = i] x_i^t]$ (empirical average reward of bandit $i$).
Let $UCB_i^t = \hat{\mu_i}^t + \sqrt{\frac{2\log 1/\delta}{N_i^t}}$ (empirical average reward + exploration bonus).
At each time $t$, select $I_t = \arg\max_{i=1\cdots n} UCB_i^t$.
```

We have the following claim for $UCB$.

**Claim 8.6.**

$$\Pr(\mu_i > UCB_i^t) \leq \delta. \tag{8.6}$$

*Proof.* The proof is again based on Hoeffding Inequality

$$\begin{aligned}
\Pr(\mu_i > UCB_i^t) &= \Pr(\mu_i > \hat{\mu_i}^t + \sqrt{\frac{2\log 1/\delta}{N_i^t}}) \\
&= \Pr(\mu_i - \hat{\mu_i}^t > \sqrt{\frac{2\log 1/\delta}{N_i^t}}) \\
(Hoeffding) &\leq \exp(-\frac{2(N_i^t)^2(\sqrt{\frac{2\log 1/\delta}{N_i^t}})^2}{4N_i^t}) \\
&= \exp(-\frac{N_i^t(\frac{2\log 1/\delta}{N_i^t})}{2}) \\
&= \exp(-\log 1/\delta) \\
&= \delta.
\end{aligned}$$

□

This claim shows that $\Pr(\mu_i < UCB_i^t) > 1 - \delta$. It means that $UCB$ has high probability $(1 - \delta)$ to overestimate $\mu_i$.

Let's define $K_i = \lceil \frac{8\log 1/\delta}{\Delta_i^2} \rceil$, we need to show that the probability of $N_i > K_i$ is very small. That is, only with small probability will bad arms be pulled very often.

To bound the probability, let's define a good case for arm $i$ as

$$G_i = \{\mu_1 > UCB_1^t, \forall t = 1, \cdots, T\} \wedge \{\hat{\mu_i}^{(K_i)} + \sqrt{\frac{2\log 1/\delta}{K_i}} < \mu_1\},$$

where

$$\hat{\mu_i}^{(K_i)} = \begin{cases} \frac{N_i^{T+1}\hat{\mu_i}^{T+1} + \sum\limits_{j=1=1}^{N_i^{T+1}-K} Y_j}{K}, & \text{where } Y_j \sim D_i \\ \hat{\mu_i}^t & \text{for first t such that } N_i^t = K. \end{cases}$$

Basically, $\hat{\mu_i}^{(K_i)}$ is the empirical average of reward from arm $i$ when it is sampled $K_i$ times. Therefore, the good case states that the UCB of the best arm estimates its true expectation and the UCB of arm $i$ when its sampled $K_i$ times is smaller than the true expectation of arm 1.

Using this, we can have the following claim.

**Claim 8.7.** *If $G_i$ is true, then $N_i^{T+1} \leq K_i$.*

*Proof.* Assume towards contradiction, there exists an arm $i$ such that $N_i^{T+1} > K_i$. Let be the final round such that $N_i^t = K_i$, which means $N_i^{t+1} = K_i + 1$ and $I_t = i$.

Then, by $G_i$, we know that

$$\begin{aligned} UCB_1^t > \mu_1 > \hat{\mu_i}^{(K_i)} + \sqrt{\frac{2\log 1/\delta}{K_i}} \\ = \hat{\mu_i}^t + \sqrt{\frac{2\log 1/\delta}{N_i^t}} \\ = UCB_i^t. \end{aligned}$$

By the update function, we know that at $t$, the player will not choose arm $i$ since $UCB_i^t$ is not the largest one. This reaches an contradiction that the player will pick arm $i$ at $t$. Thus $N_i^{T+1} \le K_i$. □

Using this, we can give the following theorem about UCB algorithm.

**Theorem 8.8.** *For $\delta^2 = \frac{1}{(T+1)^2}$, it holds that*

$$\mathbb{E}[Regret_T(UCB)] \le \sum_{i=2}^{n} \frac{16\log T}{\Delta_i} + O(\sum_{i=2}^{n}\Delta_i). \tag{8.7}$$

*Proof.* Notice that

$$\begin{aligned} \mathbb{E}[N_i^{T+1}] &= \mathbb{E}[N_i^{T+1}|G_i]\Pr(G_i) + \mathbb{E}[N_i^{T+1}|\neg G_i]\Pr(\neg G_i) \\ &\le K_i + \mathbb{E}[N_i^{T+1}|\neg G_i]\Pr(\neg G_i) \\ &\le K_i + T\Pr(\neg G_i). \end{aligned}$$

Then, let's try to bound $\Pr(\neg G_i)$. Notice that

$$\neg G_i = \{\mu_1 \le UCB_1^t, \exists t = 1, \cdots, T\} \vee \{\hat{\mu_i}^{(K_i)} + \sqrt{\frac{2\log 1/\delta}{K_i}} \ge \mu_1\}.$$

Then, we know that

$$\begin{aligned} \Pr(\neg G_i) &\le \Pr(\mu_1 \le UCB_1^t, \exists t = 1, \cdots, T) + \Pr(\hat{\mu_i}^{(K_i)} + \sqrt{\frac{2\log 1/\delta}{K_i}} \ge \mu_1) \\ &\le \sum_{s=1}^{t}\Pr(\mu_1 \le UCB_1^s) + \Pr(\hat{\mu_i}^{(K_i)} + \sqrt{\frac{2\log 1/\delta}{K_i}} \ge \mu_1) \\ \text{(Claim 8.6)} \quad &\le t\delta + \Pr(\hat{\mu_i}^{(K_i)} + \sqrt{\frac{2\log 1/\delta}{K_i}} \ge \mu_1) \\ &= t\delta + \Pr(\hat{\mu_i}^{(K_i)} + \sqrt{\frac{2\log 1/\delta}{K_i}} \ge \mu_i + \Delta_i) \\ &= t\delta + \Pr(\mu_i - \hat{\mu_i}^{(K_i)} \le \sqrt{\frac{2\log 1/\delta}{K_i}} - \Delta_i) \end{aligned}$$

Since $K_i = \lceil \frac{8\log 1/\delta}{\Delta_i^2} \rceil$, we can have that

$$
\begin{aligned}
\Pr(\mu_i - \hat{\mu_i}^{(K_i)} \le \sqrt{\frac{2\log 1/\delta}{K_i}} - \Delta_i) &= \Pr(\mu_i - \hat{\mu_i}^{(K_i)} \le \sqrt{\frac{2\log 1/\delta}{\frac{8\log 1/\delta}{\Delta_i^2}}} - \Delta_i) \\
&= \Pr(\mu_i - \hat{\mu_i}^{(K_i)} \le \sqrt{\frac{\Delta_i^2}{4}} - \Delta_i) \\
&= \Pr(\mu_i - \hat{\mu_i}^{(K_i)} \le -\frac{\Delta_i}{2}) \\
(Hoeffding) \ &\le \exp(-\frac{K_i(-\frac{\Delta_i}{2})^2}{2}) \\
&= \exp(-\frac{K_i\Delta_i^2}{8}) \\
&\le \exp(-\frac{\frac{8\log 1/\delta}{\Delta_i^2}\Delta_i^2}{8}) \\
&\le \exp(-\log 1/\delta) \\
&= \delta.
\end{aligned}
$$

Put all these together, we can have that

$$
\begin{aligned}
\mathbb{E}[Regret_T] &= \sum_{i=2}^{n} \Delta_i \mathbb{E}[N_i^{T+1}] \\
&\le \sum_{i=2}^{n} \Delta_i (K_i + T\Pr(\neg G_i)) \\
&\le \sum_{i=2}^{n} \Delta_i (K_i + T(T\delta + \delta)) \\
&\le \sum_{i=2}^{n} \Delta_i (\frac{8\log 1/\delta}{\Delta_i^2} + T(T\delta + \delta)) \\
&\le \sum_{i=2}^{n} (\frac{8\log 1/\delta}{\Delta_i} + T(T\delta + \delta)\Delta_i).
\end{aligned}
$$

With $\delta = \frac{1}{(T+1)^2}$, we can have that

$$
\begin{aligned}
\mathbb{E}[Regret_T] &\le \sum_{i=2}^{n} (\frac{8\log(T+1)^2}{\Delta_i} + \frac{T}{T+1}\Delta_i) \\
&\le \sum_{i=2}^{n} (\frac{16\log(T+1)}{\Delta_i}) + (n-1)\Delta_i
\end{aligned}
$$

□

This finishes the proof of the last algorithm in this chapter. UCB algorithm is much more realistic and generalizable in real life. It removes the assumption about knowing $\Delta_*$, the bound is not much worse compared with simple algorithm.

# Chapter 9

# Statistical Learning Theory

## 9.1 Statistical Learning Setting

In this chapter, we will talk about statistical learning theory. We will introduce several concept, like VC dimension, Rademacher Complexity. We will also talk about many lemmas to connect these definitions. Finally, we will apply our tools to models like Neural Network, etc.

To begin with, let's discuss the setting in statistical learning.

- Given observation space $X$.
- Given label space $Y$. For example,
  - $\{0, 1\}$: classification.
  - $[k]$: multi-class classification.
  - $\mathbb{R}$: regression.
- Prediction space $\hat{Y}$. In most cases $\hat{Y}$ is the same as $Y$, but sometimes they would be different.
- There is unknown distribution
$$D \in \Delta(X \times Y)$$
- We work with hypothesis space $\mathcal{H}$. For example,
  - Linear threshold functions: $h_{w,b}(x) = \mathbf{1}[wx + b > 0]$.
  - Decision stumps: $h_{i,c} = \mathbf{1}[x_i > c]$.
  - Neural Network: $h_{M_1,b_1,M_2,b_2,\cdots,M_k,b_k}(x) = \sigma(b_k + M_k\sigma(b_{k-1} + \cdots (x)))$, where $\sigma$ is the sigmoid function.
- Loss function $l$: $\hat{Y} \times Y \to \mathbb{R}$. For example,
  - Square Loss: $l(\hat{y}, y) = (y - \hat{y})^2$.
  - Hinge Loss: $l(\hat{y}, y) = \max(0, 1 - \hat{y}y)$.
  - 0-1 Loss: $l(\hat{y}, y) = \mathbf{1}[\hat{y} \neq y]$.

We now give a definition used in this setting.

**Definition 9.1** (**Risk**)**.** *Given distribution $D$, hypothesis $h \in \mathcal{H}$, we can define the risk of $h$ as*

$$R(h) = \mathbb{E}_{(x,y)\in D}[l(h(x), y)]. \tag{9.1}$$

Intuitively, the risk is just the expectation of loss function over samples in the distribution.

However, since $D$ is unknown to us, we cannot compute $R$ directly. Alternatively, we can compute empirical risk.

**Definition 9.2** (**Empirical Risk**)**.** *Given data $\{(x_1, y_1), \cdots, (x_n, y_n)\} \sim D$, the empirical risk is defined as*

$$\hat{R}_n(h) = \frac{1}{n}\sum_{i=1}^{n} l(h(x_i), y_i). \tag{9.2}$$

## 9.2 Empirical Risk Minimization

**Definition 9.3** (**Empirical Risk Minimization**)**.** *An ERM (Empirical Risk Minimization) algorithm tries to learn an*

$$\hat{h}^{ERM} = \arg\min_{h\in\mathcal{H}} \hat{R}_n(h). \tag{9.3}$$

Basically, it just finds an algorithm that could minimization empirical risk.

In addition to empirical risk minimization, we also hope to know how well it could generalize. Therefore, we introduce two kinds of error.

**Definition 9.4** (**Estimation Error**)**.**

$$R(\hat{h_n}^{ERM}) - \min_{h^*\in\mathcal{H}} R(h^*). \tag{9.4}$$

**Definition 9.5** (**Approximation Error**)**.**

$$R(h^*) - \min_{\textit{all functions } h^{**}} R(h^{**}), \tag{9.5}$$

*where $h^{**}(x) = \arg\min_{\hat{y}} \mathbb{E}_{y\sim D(\cdot|x)}[l(\hat{y}, y)]$.*

Basically, estimation error captures how well our model learned from the sample data can generalize to other data following the same distribution. Approximation Error captures how well our function class compared to all function classes.

For now, we just focus on bounding estimation error. Notice that

$$\begin{aligned} R(\hat{h_n}^{ERM}) - \min_{h^*\in\mathcal{H}} R(h^*) = {} & R(\hat{h_n}^{ERM}) - \hat{R}_n(\hat{h_n}^{ERM}) \\ & + \hat{R}_n(\hat{h_n}^{ERM}) - \hat{R}_n(h^*) \\ & + \hat{R}_n(h^*) - R(h^*). \end{aligned}$$

Since $\hat{h}_n^{ERM} = \arg\min_{h\in\mathcal{H}} \hat{R}_n(h)$, we know that $\hat{R}_n(\hat{h_n}^{ERM}) - \hat{R}_n(h^*) \le 0$. Then

$$\begin{aligned} R(\hat{h_n}^{ERM}) - \min_{h^*\in\mathcal{H}} R(h^*) & \le R(\hat{h_n}^{ERM}) - \hat{R}_n(\hat{h_n}^{ERM}) + \hat{R}_n(h^*) - R(h^*) \\ & \le 2\sup_{h\in\mathcal{H}} |R(h) - \hat{R}_n(h)|. \end{aligned}$$

Bounding $|R(h) - \hat{R}_n(h)|$ is known as uniform deviation bound.

Let's try to prove something wrong first.

***False Proof.*** Let $X_i = l(\hat{h}_n^{ERM}(x_i), y_i)$, where $(x_i, y_i)$ is the $i$-th element in the training set. Then by definition,

$$R(\hat{h}_n^{ERM}) = \mathbb{E}_{(x,y)\in D}[l(\hat{h}_n^{ERM}(x), y)] = \mu.$$

Therefore

$$\mathbb{E}_{x_i,y_i} = \mu.$$

Let's use Hoeffding, assume $l$ is bounded in $[0, 1]$, then

$$\hat{R}_n(\hat{h_n}^{ERM}) - R(\hat{h_n}^{ERM}) = \frac{1}{n}\sum_{i=1}^{n} X_i - \mu \quad \leq \sqrt{\frac{\log 1/\delta}{2n}}$$

with probability $1 - \delta$. □

The problem with this proof is that Hoeffding's inequality requires each random variable to be independent. However, in our case, $X_i$ is not independent with each other. The reason is that $\hat{h}_n^{ERM} = \arg\min_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} l(h(x_i), y_i)$. This makes these sample data correlated to each other. To be more specific, consider two points $(x_1, y_1)$ and $(x_2, y_2)$, if we change $(x_1, y_1)$, then $\hat{h}_n^{ERM}$ will change. In consequence, $X_2$ will also change. Therefore, we cannot use Hoeffding inequality here.

However, if we remove "ERM", everything will be fine. Changing $(x_1, y_1)$ will not affect $h$ and $X_2$. Therefore, alternatively we can do this,

$$\begin{aligned}
\Pr(\sup_{h\in\mathcal{H}} |\hat{R}_n(h) - R(h)| > t) &\leq \Pr(\exists h, |\hat{R}_n(h) - R(h)| > t) \\
&\leq \sum_{h\in\mathcal{H}} \Pr(|\hat{R}_n(h) - R(h)| > t) \\
&\leq \sum_{h\in\mathcal{H}} \exp(-2nt^2) \\
&\leq |\mathcal{H}| \exp(-2nt^2).
\end{aligned}$$

Following this, we can give the lemma about risk function.

**Lemma 9.6.** *If $H$ is finite, then*

$$\mathbb{E}[R(\hat{h}) - R(h^*)] = O(\sqrt{\frac{\log 1/\delta + \log|\mathcal{H}|}{n}})$$

*with probability larger than $1 - \delta$.*

*Proof.* From our previous deduction,

$$R(\hat{h}) - R(h^*) \leq 2 \sup_{h\in\mathcal{H}} |R(h) - \hat{R}_n(h)|$$

$$\Pr(\sup_{h\in\mathcal{H}} |\hat{R}_n(h) - R(h)| > t) \leq |\mathcal{H}| \exp(-2nt^2).$$

By setting $t = \sqrt{\frac{\log 2/\delta + \log|\mathcal{H}|}{2n}}$, we can get that

$$\begin{aligned}
\Pr(\sup_{h\in\mathcal{H}} |\hat{R}_n(h) - R(h)| > \sqrt{\frac{\log 2/\delta + \log|\mathcal{H}|}{2n}}) &\leq |\mathcal{H}| \exp(-(\log 2/\delta + \log|\mathcal{H}|)) \\
&\leq |\mathcal{H}| \exp(-(\log 2|\mathcal{H}|/\delta)) \\
&\leq \delta.
\end{aligned}$$

Therefore, with probability larger than $1-\delta$, we can have

$$\sup_{h\in\mathcal{H}} |\hat{R}_n(h) - R(h)| \leq \sqrt{\frac{\log 2/\delta + \log|\mathcal{H}|}{2n}}$$

.

In conclusion,

$$\begin{aligned}
\mathbb{E}[R(\hat{h}) - R(h^*)] &\leq 2 \sup_{h\in\mathcal{H}} |R(h) - \hat{R}_n(h)| \\
&\leq 2\sqrt{\frac{\log 2/\delta + \log|\mathcal{H}|}{2n}} \\
&= O(\sqrt{\frac{\log 2/\delta + \log|\mathcal{H}|}{2n}})
\end{aligned}$$

with probability at least $1-\delta$. □

This concludes our discussion about empirical risk minimization when $|\mathcal{H}|$ is finite. We can see that if $|\mathcal{H}|$ is finite, the model we learn from data compared with the best model in the function class has bounded error $(\sqrt{\frac{\log 2/\delta + \log|\mathcal{H}|}{2n}})$. This also shows that if we increase the number of samples, we could get closer to the best model in function class.

However, what if $|\mathcal{H}|$ is not finite? This is quite common for most statistical learning model. To handle this situation, we need to give some more definitions that will be used.

For now, assume that $\mathcal{H}$ is a class of binary functions.

**Definition 9.7 (Growth Function).** *The growth function of a function class $\mathcal{H}$ is defined as*

$$\Pi_{\mathcal{H}}(m) = \sup_{S=\{x_1,\cdots,x_m\}\subset X} |\{(h(x_1),\cdots,h(x_m))\} : h \in \mathcal{H}|.$$

Intuitively, it means that fixing a set of points in $X$, by changing the function $h \in \mathcal{H}$, how many different label vectors can we generate.

We have the following claim for growth function,

**Claim 9.8.**

$$\Pi_{\mathcal{H}}(m) \leq 2^m.$$

This is intuitive since the number of different vectors with length $m$ and each element can take only two values is definitely not larger than $2^m$.

Using growth function, we can give another important definition

**Definition 9.9 (VC dimension).** *The VC dimension of a function class $\mathcal{H}$ is the largest value d such that*

$$\Pi_{\mathcal{H}}(d) = 2^d. \tag{9.6}$$

*Basically, VC dimension is the size of largest set of X's that could be shattered.*

For VC dimension, we have the following claim.

**Claim 9.10.** *In $\mathbb{R}^d$, the class $\mathcal{H}$ of binary threshold functions has VC dimension $d + 1$.*

The reason that we introduce VC dimension is that we hope to show that VC dimension characterizes the "learnability" of a function class $\mathcal{H}$. It means

1. $\forall D \in \Delta(X, Y)$, it has
$$\sup_{h\in\mathcal{H}} |R(h) - \hat{R}_n(h)| = O(\sqrt{\frac{\log 2/\delta + VCdim(H)}{n}}).$$
2. This equation is tight up to log factors. That is, $\exists D \in \Delta(X, Y)$ such that no algorithm can guarantee
$$|R(h) - \hat{R}_n(h)| \leq O(\sqrt{\frac{VCdim(H)}{n}}).$$

Before proving this, we need some more definitions.

**Definition 9.11 (Rademacher Random Variable).** *We say that a random variable X is called Rademacher Random Variable if*

$$\Pr(X = 1) = \Pr(X = -1) = \frac{1}{2}. \tag{9.7}$$

**Definition 9.12 (Empirical Rademacher Complexity).** *Give samples $S = \{X_1, \cdots X_m\} \in X$, the Empirical Rademacher Complexity of a function class $\mathcal{H}$ is defined as*

$$\hat{\mathcal{R}}_S(\mathcal{H}) = \mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\frac{1}{m}\sup_{h\in\mathcal{H}} \sum_{i=1}^{n} \sigma_i h(x_i)], \tag{9.8}$$

*where $\sigma_1 \cdots \sigma_n$ is Rademacher random variables.*

Basically, Empirical Rademacher Complexity captures the richness of function class $\mathcal{H}$. It measures how well the functions can fit random noise.

In addition to Empirical Rademacher Complexity, let's also define Rademacher Complexity

**Definition 9.13 (Rademacher Complexity).** *Given distribution $D \in \Delta(X)$, the Rademacher Complexity of a function class $\mathcal{H}$ is defined as*

$$\mathcal{R}_n(G) = \mathbb{E}_{S\sim D^n}[\hat{\mathcal{R}}_S(\mathcal{H})]. \tag{9.9}$$

It represents the expectation of Empirical Rademacher Complexity if we randomly select $n$ samples from the distribution.

We have the following two facts for Empirical Rademacher Complexity

**Fact 9.13.1.**

1. *If $|G| = 1$, then $\hat{\mathcal{R}}_S(G) = 0$.*
2. *If $S = \{x_1, \cdots, x_n\}$ are all distinct, and $G = \{$all function $\in \{0,1\}\}$, then $\hat{\mathcal{R}}_s(G) = 1$.*

Before proceeding, let's give several notations for convenience.

**Notation 9.14.** *For a distribution $D \in \Delta(X)$, let's define*

$$\begin{aligned} \mathbb{E}h &= \mathbb{E}_{x \in D}[h(x)] \\ \hat{\mathbb{E}}_S h &= \frac{1}{m}\sum_{i=1}^{m} h(x_i),\ \ S = \{x_1, \cdots, x_n\} \end{aligned} \tag{9.10}$$

Basically, $\mathbb{E}h$ is the expectation value of function $h$ over a sample following the distribution, while $\hat{\mathbb{E}}_S h$ is the average function value for the samples.

By these, we have the following theorem.

**Theorem 9.15** (**Symmetrization Lemma**)**.** *Let $S = (x_1, \cdots, x_m) \sim D^m$, let $h \in \mathcal{H}$ be bounded in $[0,1]$, then $\forall h \in \mathcal{H}$, it holds that*

$$\hat{\mathbb{E}}_S h - \mathbb{E}h \leq 2\mathcal{R}_n(H) + \sqrt{\frac{\log 1/\delta}{2n}} \tag{9.11}$$

*with probability larger than $1 - \delta$.*

This theorem bounds the difference between average function value for sample and the expectation value.

To prove this, we need the following lemma,

**Lemma 9.16** (**McDiarmid's Inequality**)**.** *Let $f$ be a function that maps $S \to \mathbb{R}$. $\forall x_i \in S$, let $S'$ denote $(S - \{x_i\}) \cup x_i'$. Suppose that $\forall x_i'$, it has*

$$|f(S) - f(S')| \leq c$$

*Then it holds that*

$$\Pr(f(S) - \mathbb{E}_{S \sim D}[f(S)] > t) \leq \exp(-\frac{2t^2}{nc^2}). \tag{9.12}$$

This lemma states that if the function value will not change more than $c$ by exchanging only one element in the sample. Then the difference between function value and the expectation of function value can be bounded in probability.

*Proof.* Let's define $U_k$ as

$$U_k = \mathbb{E}_S[f(S)|x_1, \cdots x_k],$$

where $S = \{x_1, \cdots, x_n\}$.

We can say the following things for $U_k$

1. $U_0 = \mathbb{E}_S[f(S)]$.
2. $U_n = \mathbb{E}_S[f(S)|x_1, \cdots x_n] = f(S)$.
3. $\forall k$, $|U_k - U_{k-1}| \leq c$.
4. $U_0 \cdots U_n$ is a martingale.

Let's prove the last one.

We can see that

$$\begin{aligned}\mathbb{E}[U_k|x_1, \cdots x_{k-1}] &= \mathbb{E}[\mathbb{E}[f(S)|x_1, \cdots x_k]|x_1, \cdots, x_{k-1}] \\ &= \mathbb{E}[f(S)|x_1, \cdots, x_{k-1}] \\ &= U_{k-1}\end{aligned}$$

Therefore, by Azuma's inequality, we can have that

$$\Pr(U_n - U_0 > t) \leq \exp(-\frac{2t^2}{nc^2}).$$

□

This finishes the proof of the lemma, let's prove theorem 9.15 now.

*Proof.* To use lemma 9.16, let $\Phi(S) = \sup_{h \in \mathcal{H}}(\hat{\mathbb{E}}_S h - \mathbb{E}h)$. We first need to check that we can apply lemma 9.16 to $\Phi(S)$. Let $S'$ and $S$ differ by one element. Notice that

$$\begin{aligned}|\Phi(S) - \Phi(S')| &= |\sup_{h \in \mathcal{H}}(\hat{\mathbb{E}}_S h - \mathbb{E}h) - \sup_{h' \in \mathcal{H}}(\hat{\mathbb{E}}_{S'} h - \mathbb{E}h)| \\ &\leq \sup_{h \in \mathcal{H}}(\hat{\mathbb{E}}_{S'} h - \hat{\mathbb{E}}_S h) \\ &= \sup_{h \in \mathcal{H}} \frac{1}{n}(\sum_{i=1}^n h(x_i) - \sum_{i=1}^n h(x_i')) \\ &\leq \sup_{h \in \mathcal{H}} \frac{1}{n}(h(x_i) - h(x_i')) \\ &\leq \frac{1}{n}.\end{aligned}$$

Therefore, it satisfies the assumption in the lemma, and we can see that

$$\Pr(\Phi(S) - \mathbb{E}_{S\sim D}[\Phi(S)] > t) \leq \exp(-2t^2 n).$$

Let $\delta = -\exp(-2t^2 n)$ and we can get that $\Phi(S) - \mathbb{E}_{S\sim D}[\Phi(S)] \leq \sqrt{\frac{\log 1/\delta}{2n}}$ with probability larger than $1 - \delta$.

Now let's analysis $\mathbb{E}_{S\sim D}[\Phi(S)]$. Notice that

$$\begin{aligned}
\mathbb{E}_{S\sim D}[\Phi(S)] &= \mathbb{E}_{S\sim D}[\sup_{h\in\mathcal{H}}(\hat{\mathbb{E}}_S h - \mathbb{E}h)] \\
&= \mathbb{E}_{S\sim D}[\sup_{h\in\mathcal{H}}(\hat{\mathbb{E}}_S h - \mathbb{E}_{S'\sim D}[\hat{\mathbb{E}}_{S'} h])] \\
&\leq \mathbb{E}_{S\sim D}\mathbb{E}_{S'\sim D}[\sup_{h\in\mathcal{H}}(\hat{\mathbb{E}}_S h - \hat{\mathbb{E}}_{S'} h)] \\
&= \frac{1}{n}\mathbb{E}_{S\sim D}\mathbb{E}_{S'\sim D}\mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\sup_{h\in\mathcal{H}}\sum_{i=1}^n \sigma_i(h(x_i) - h(x_i'))] \\
&\leq \frac{1}{n}\mathbb{E}_{S\sim D}\mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\sup_{h\in\mathcal{H}}\sum_{i=1}^n \sigma_i(h(x_i))] + \frac{1}{n}\mathbb{E}_{S'\sim D}\mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\sup_{h'\in\mathcal{H}}\sum_{i=1}^n \sigma_i(h'(x_i'))] \\
&= \mathcal{R}_n(G) + \mathcal{R}_n(G) \\
&= 2\mathcal{R}_n(G).
\end{aligned}$$

Combine all above, we can see that

$$\begin{aligned}
\hat{\mathbb{E}}_S h - \mathbb{E}h &= \Phi(S) \\
&\leq \mathbb{E}[\Phi(S)] + \sqrt{\frac{\log 1/\delta}{2n}} \\
&\leq 2\mathcal{R}_n(S) + \sqrt{\frac{\log 1/\delta}{2n}}.
\end{aligned}$$

□

Before continuing, let's give another notation.

**Notation 9.17.** *Given $S = \{x_1, \cdots x_n\}$, let*

$$\mathcal{H}|S = \{h(x_1), \cdots, h(x_n) : h \in \mathcal{H}\}.$$

By this definition, we can see that the growth function $\Pi_{\mathcal{H}}(n) = \max_{|S|=n} |\mathcal{H}|S|$, here $\mathcal{H}$ is assumed to be a binary class of functions.

Let's talk about Massart's Lemma.

**Theorem 9.18 (Massart's Lemma).** *Let $A \subset \mathbb{R}^n$ be a finite set with $\max_{a\in A} ||a||_2 \leq r$, then we have*

$$\mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\sup_{a\in A}\sum_{i=1}^n a_i\sigma_i] \leq r\sqrt{2\log|A|}. \tag{9.13}$$

*Proof.* For any $\lambda > 0$, we know that

$$
\begin{aligned}
\exp(\lambda \mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\sup_{a\in A}\sum_{i=1}^n a_i\sigma_i]) &\le \mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\exp(\lambda \sup_{a\in A}\sum_{i=1}^n a_i\sigma_i)] \\
&= \mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\sup_{a\in A}\exp(\lambda \sum_{i=1}^n a_i\sigma_i)] \\
&\le \mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\sum_{a\in A}\exp(\lambda \sum_{i=1}^n a_i\sigma_i)] \\
&\le \sum_{a\in A}\mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\exp(\lambda \sum_{i=1}^n a_i\sigma_i)] \\
&\le \sum_{a\in A}\mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\Pi_{i=1}^n \exp(\lambda a_i\sigma_i)] \\
&\le \sum_{a\in A}\Pi_{i=1}^n \mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\exp(\lambda a_i\sigma_i)] \\
(Hoeffding) &\le \sum_{a\in A}\Pi_{i=1}^n \exp(\frac{\lambda^2 a_i^2}{2}) \\
&\le \sum_{a\in A}\exp(\frac{\lambda^2\sum_{i=1}^n a_i^2}{2}) \\
&\le \sum_{a\in A}\exp(\frac{\lambda^2 r^2}{2}) \\
&\le |A|\exp(\frac{\lambda^2 r^2}{2}).
\end{aligned}
$$

Then, just take log on both sides and we can get that

$$\mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\sup_{a\in A}\sum_{i=1}^n a_i\sigma_i]) \le \frac{\log|A|}{\lambda} + \frac{r^2\lambda}{2}.$$

Set $\lambda = \sqrt{\frac{2\log|A|}{r^2}}$ and we can get the result. □

The reason we introduce Massart's Lemma is that we want to give an upper bound on $\hat{\mathcal{R}_S}(\mathcal{H})$.

**Claim 9.19.** *The empirical Radamacher Complexity of a function class $\mathcal{H}$ is upper bounded by* $\sqrt{\frac{2\log\Pi_{\mathcal{H}}(n)}{n}}$.

*Proof.* By definition,

$$
\begin{aligned}
\hat{\mathcal{R}_S}(\mathcal{H}) &= \mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\frac{1}{n}\sup_{h\in\mathcal{H}}\sum_{i=1}^{n}\sigma_i h(x_i)] \\
&= \mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\frac{1}{n}\sup_{a\in\mathcal{H}|S}\sum_{i=1}^{n}\sigma_i a_i] \\
&\le \frac{1}{n} r\sqrt{2\log|\mathcal{H}|S|} \\
&= \sqrt{\frac{2\log|\mathcal{H}|S|}{n}} \\
&\le \sqrt{\frac{2\log\Pi_{\mathcal{H}}(n)}{n}}.
\end{aligned}
$$

□

Previously, we already known that

$$\Phi(S) \le \mathcal{R}_n(S) + \sqrt{\frac{\log 1/\delta}{2n}}.$$

Now with this lemma, we can replace Radamacher complexity with growth function and get

$$\Phi(S) \le \sqrt{\frac{2\log\Pi_{\mathcal{H}}(n)}{n}} + \sqrt{\frac{\log 1/\delta}{2n}}.$$

Next, we will try to express the growth function in term of VC-dimension. First of all, we will introduce Sauer's Lemma, which bounds growth function with VC dimension.

**Theorem 9.20 (Sauer's Lemma).** *Suppose that a binary function class $\mathcal{H}$ has VC-dimension $d$, then the growth function of $\mathcal{H}$ is bounded by*

$$\Pi_{\mathcal{H}}(n) \le \sum_{i=0}^{d}\binom{n}{i} \le n^d. \tag{9.14}$$

*Proof.* Let $M_{S,\mathcal{H}}$ be the matrix whose **unique** rows are $(h(x_1),\cdots,h(x_n))$, where $h\in\mathcal{H}$ and $S=\{x_1,\cdots,x_n\}$. The rows are unique by deleting repeated rows.

For this matrix, we can have the following two facts.

**Fact 9.20.1.** *1.* $\Pi_{\mathcal{H}}(n) = \max_{|S|=n} \#\ rows(M_{S,\mathcal{H}})$

*2. If all rows of $M_{S,\mathcal{H}}$ have $\le d$ 1s, then $\#\ rows(M_{S,\mathcal{H}}) \le \sum_{i=0}^{d}\binom{m}{i}$.*

Then the trick of the proof is to modify $M_{S,\mathcal{H}}$ such that every row has $\le d$ 1s and VC-dim will not increase and no rows are duplicate.

The way we modify the matrix is very simple.

For this procedure, we can have the following claim.

**Claim 9.21.** *After procedure finishes, if for some set of columns $U\subset[n]$, there exists row with all ones in $U$, then we are shattering $U$.*

```
For epoch $= 1, 2, \cdots$
    For col $j = 1, 2, \cdots n$
        For row $i = 1, \cdots$
            Let $M_{S,\mathcal{H}}(i,j) = 0$ if this does not duplicate another row.
```

Hence, the VC dimension of resulting matrix is just the largest number of 1s that we can find in the row.

What remains to be proved is that the VC-dimension will not increase under this procedure.

Assume towards contradiction that after shifting some column $j$, a new subset $U \subset [n]$ was suddenly shattered. This must mean that $\exists$ rows $i, k$ such that $M(i, U) = M(k, U)$ before shifting but $M'(i, U) \neq M'(k, U)$ after shifting.

Without loss of generality let $M(k, j) = 1$ and $M'(k, j) = 0$, thus $M(i, j) = 1$ and $M'(i, j) =$.

We do not change $M'(i, j)$ into 0 because there exists a row $l \neq k$ that $l$ would be duplicate if we change $M'(i, j)$ into zero.

This means that $M'(l, U) \neq M'(k, U)$, which indicates that the set $U$ is already shattered.

□

Now that we have all above tools, we want to give a generalization bound on estimation error

$$\sup_{h \in H} (\frac{1}{n} \sum_{i=1}^{n} l(h(x_i), y_i)) - \mathbb{E}_{(x,y)\sim D} l(h(x), y).$$

Notice that in our setting, we have a hypothesis class $\mathcal{H} : X \to \hat{Y}$ and a loss function $l : \hat{Y} \times Y \to \mathbb{R}$. Actually, the loss function itself is a set of functions

$$G_{\mathcal{H}} = \{g_h(x, y) = l(h(x), y), h \in \mathcal{H}\}.$$

The interesting thing is that we can use loss function class to represent estimation error.

Notice that

$$\sup_{h \in H} (\frac{1}{n} \sum_{i=1}^{n} l(h(x_i), y_i) - \mathbb{E}_{(x,y)\sim D} l(h(x), y)) = \sup_{g_n \in G_{\mathcal{H}}} \hat{\mathbb{E}}_S g_n - \mathbb{E} g_n.$$

To continue, we will give the following claim.

**Claim 9.22.** *Let $\mathcal{H}$ be a binary function class, and let $G_{\mathcal{H}}$ be the corresponding loss class for 0-1 loss. Then it has*

$$\hat{\mathcal{R}}_S(G_{\mathcal{H}}) = \hat{\mathcal{R}}_{S|x}(\mathcal{H}), \tag{9.15}$$

*where $S|x = \{x_1, \cdots, x_n\}$.*

This claim states that the Rademacher complexity for the loss function class is the same as that of the function class itself.

*Proof.*

$$\begin{aligned}
\hat{\mathcal{R}}_S(G_{\mathcal{H}}) &= \mathbb{E}_{\sigma_1 \cdots \sigma_n}[\frac{1}{n} \sup_{g_h \in G_{\mathcal{H}}} \sum_{i=1}^{n} \sigma_i g_h(x_i, y_i)] \\
&= \mathbb{E}_{\sigma_1 \cdots \sigma_n}[\frac{1}{n} \sup_{g_h \in G_{\mathcal{H}}} \sum_{i=1}^{n} \sigma_i \mathbf{1}[h(x_i) \neq y_i]] \\
&= \mathbb{E}_{\sigma_1 \cdots \sigma_n}[\frac{1}{2n} \sup_{g_h \in G_{\mathcal{H}}} \sum_{i=1}^{n} \sigma_i (2 \cdot \mathbf{1}[h(x_i) \neq y_i] - 1)] \\
&= \mathbb{E}_{\sigma_1 \cdots \sigma_n}[\frac{1}{2n} \sup_{g_h \in G_{\mathcal{H}}} \sum_{i=1}^{n} \sigma_i (2 \cdot \mathbf{1}[h(x_i) \neq 0] - 1)] \\
&= \mathbb{E}_{\sigma_1 \cdots \sigma_n}[\frac{1}{2n} \sup_{g_h \in G_{\mathcal{H}}} \sum_{i=1}^{n} \sigma_i (2 \cdot \mathbf{1}[h(x_i) \neq 0])] \\
&= \mathbb{E}_{\sigma_1 \cdots \sigma_n}[\frac{1}{2n} \sup_{g_h \in G_{\mathcal{H}}} \sum_{i=1}^{n} \sigma_i (2h(x_i))]
\end{aligned}$$

$$= \hat{\mathcal{R}}_{S|x}(\mathcal{H}).$$

□

In this proof, I can see three thicks used:

- We can add constant number into Rademacher Complexity without changing its value.
- Both $\mathcal{H}$ and loss function are binary function class.
- The reason we turn $\mathbf{1}(\cdot)$ into $2 \cdot \mathbf{1}(\cdot) - 1$ is to make the value symmetric for 0. In this way we can just replace $y_i$ as 0.

Finally, let's will give a bound for the estimation error using VC-dimension. By what we have deduced so far, the estimation error is bounded as

$$\begin{aligned}
R(\hat{h}^{ERM}) - \min_{h^* \in \mathcal{H}} R(h^*) &\leq c_1 \sup_{h \in \mathcal{H}} |R(h) - \hat{R}(h)| \\
&\leq c_1 \sup_{g_h \in G_{\mathcal{H}}} |\hat{\mathbb{E}}_S g_h - \mathbb{E} g_h| \\
&\leq c_2 \sup_{g_h \in G_{\mathcal{H}}} \hat{\mathbb{E}}_S g_h - \mathbb{E} g_h \\
&\leq c_3 (\mathcal{R}_n(G_{\mathcal{H}}) + \sqrt{\frac{\log 1/\delta}{2n}}) \\
&\leq c_3 (\mathcal{R}_n(\mathcal{H}) + \sqrt{\frac{\log 1/\delta}{2n}}) \\
&\leq c_4 (\sqrt{\frac{\log \Pi_H(n)}{n}} + \sqrt{\frac{\log 1/\delta}{2n}}) \\
&\leq c_4 (\sqrt{\frac{\log(n^d)}{n}} + \sqrt{\frac{\log 1/\delta}{2n}}) \\
&\leq c_4 (\sqrt{\frac{d \log(n)}{n}} + \sqrt{\frac{\log 1/\delta}{2n}})
\end{aligned}$$

The above shows that the bound of estimation error is approximately $O(\sqrt{\frac{d\log(n)}{n}})$. This is the upper bound of this error, we will next try to give a lower bound.

**Claim 9.23.** *Given a hyphothesis class $H$ with VC-dimension $d$, there exists a family of distribution $D_\sigma$, $\sigma \in \Gamma$, such that for any algorithm $\mathcal{A} : \{(x_1, y_1), \cdots (x_n, y_n)\} \to h$, it holds that*

$$\Pr(R(\hat{h}) - \min_{h\in\mathcal{H}} R(h)) > \sqrt{\frac{d}{320n}}) > \frac{1}{64}. \tag{9.16}$$

The proof is very complicate, we will only give the sketch here.

*Proof Sketch.* Sample $\sigma_1, \cdots, \sigma_d$ as i.i.d Rademacher distribution random variable.

Let $x_1, \cdots, x_d$ be the shattered set of $X$ and define $D_\sigma$ on $X \times \{0, 1\}$ as follows:

$$\begin{aligned} \Pr_{(x,y)\sim D_\sigma}((x, y) = (x_i, 1)) &= (\frac{1}{2} + c\sigma_i\frac{d}{m})\frac{1}{d} \\ \Pr_{(x,y)\sim D_\sigma}((x, y) = (x_i, 0)) &= (\frac{1}{2} - c\sigma_i\frac{d}{m})\frac{1}{d} \end{aligned} \tag{9.17}$$

**Lemma 9.24.** *If a coin has distribution Bernoulli($\frac{1}{2} \pm r$), then we need $O(\frac{1}{r^2})$ samples to get $\frac{2}{3}$ chance of correctly guessing the bias direction.*

Combine this lemma and the above proof sketch we can get a rough sense of the complete proof. □

## 9.3 Neural Network

Before we talk about neural network, let's first talk about two properties of growth function.

**Fact 9.24.1.** *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two function classes, then*

1. *If $\mathcal{H}_1$ and $\mathcal{H}_2$ both map $X \to Y$, then $\Pi_{\mathcal{H}_1\times\mathcal{H}_2}(m) \le \Pi_{\mathcal{H}_1}(m) \cdot \Pi_{\mathcal{H}_2}(m)$.*
2. *If $\mathcal{H}_1$ maps $X \to Y$ and $\mathcal{H}_2$ maps $Y \to Z$, then $\Pi_{\mathcal{H}_1\circ\mathcal{H}_2}(m) \le \Pi_{\mathcal{H}_1}(m) \cdot \Pi_{\mathcal{H}_2}(m)$.*

Next, we will talk about neural network.

**Definition 9.25 (Neural Network).** *Let $X = \mathbb{R}^{d_0}$. A neural network with respect to binary activation is a composition of $f_n$'s.:*

$$f_l \circ f_{l-1} \cdots f_2 \circ f_1 : X \to \{-1, 1\}, \tag{9.18}$$

*where $f_i : \mathbb{R}^{d_{i-1}} \to \{-1, 1\}^{d_i}$ for $i = 1, \cdots, l-1$ and $f_l : \mathbb{R}^{d_{l-1}} \to \{-1, 1\}$.*

*We can see that we have $l$ layers, and each layer has $d_i$ nodes. For a layer $i$ and input $u \in \mathbb{R}^{d_{i-1}}$ the activation function of the $j$th node is*

$$f_{i,j}(u) = sign(w^{i,j}u - \theta^{i,j}), \ j = 1, \cdots, d_i. \tag{9.19}$$

Now we start calculating the VC dimension of Neural Network.

Notice that $f_{i,j} \in \mathcal{H}_{i,j}$, where $\mathcal{H}_{i,j}$ is the linear thresholds on $d_{i-1}$ dimensions. We previous show that the VC dimension of linear thresholds in $d_{i-1}$ dimension is $d_{i-1}+1$.

Therefore,

$$\text{VC-dim}(\mathcal{H}_i, j) = d_{i-1} + 1.$$

Furthermore, since $\mathcal{H}_i = \mathcal{H}_{i,1} \times \mathcal{H}_{i,2} \times \cdots \times \mathcal{H}_{i,d_i}$, by 9.24.1, we can see that the VC dimension of the $i$th layer function class is

$$\Pi_{\mathcal{H}_i(m)} \leq \prod_{j=1}^{d_i} \Pi_{\mathcal{H}_{i,j}}(m).$$

In addition, since the entire class of Neural Network is $\mathcal{F} = \mathcal{H}_l \circ \mathcal{H}_{l-1} \circ \cdots \mathcal{H}_1$, by fact 9.24.1, we know that

$$\begin{aligned}\Pi_{\mathcal{F}} &\leq \prod_{i=1}^{l} \Pi_{\mathcal{H}_i}(m) \\ &\leq \prod_{i=1}^{l} \prod_{j=1}^{d_i} \Pi_{\mathcal{H}_{i,j}}(m) \\ &\leq \prod_{i=1}^{l} \prod_{j=1}^{d_i} m^{d_{i-1}-1} \\ &= m^{\sum_{i=1}^{l} \sum_{j=1}^{d_i} d_{i-1}-1}.\end{aligned}$$

We have the following claim for VC dimension and growth function.

**Claim 9.26.** *If $\Pi_{\mathcal{H}} \leq m^N$, then*

$$\textit{VC-dim}(\mathcal{H}) = O(N \log N). \tag{9.20}$$

Therefore, we can see that the VC dimension of Neural Network is

$$O((\sum_{i=1}^{l} \sum_{j=1}^{d_i} d_{i-1} - 1) \log(\sum_{i=1}^{l} \sum_{j=1}^{d_i} d_{i-1} - 1))$$

This is a large VC dimension and thus we will need a lot of training data to reduce estimation error. However, this is based on binary activation. When it comes to **Sigmoid** and **Relu** activation, we can prove that its VC dimension is $O(T^2)$, where $T$ is the number of operators.

## 9.4 Margin Theory

Finally, we will discuss about Margin Theory. First, let's define margin.

**Definition 9.27** (**Margin of Linear Classifier**)**.** *Given a linear classifier function $h_w(x) = sign(w \cdot x)$, given dataset $S = \{(x_i, y_i) : i = 1, \cdots, m\}$, we can define the margin of $h_w$ on $S$ as*

$$\rho(S) = \min_{i=1,\cdots,m} \frac{y_i(w \cdot x_i)}{||w||_2}. \tag{9.21}$$

The reason we introduce margin is that a classifier with a larger margin can works better and has better generalization. Intuitively, a large margin means the probability of wrongly classifying an input is very small.

We hope to find a lower bound on margin. To do that, let's introduce the following class of linear classifier with bounded norm:

**Definition 9.28** (**Linear Classifiers with Bounded Norm**)**.** *A linear classifier with bounded norm is defined as*

$$\mathcal{H}_\Lambda := \{h_w : \mathbb{R}^d \to \{-1, 1\}, ||w||_2 \leq \Lambda\}. \tag{9.22}$$

Let a loss function $l(\hat{y}, y) = \varphi_\rho(\hat{y}y)$, where $\varphi_\rho(\hat{y}y)$ is defined as

$$\varphi_\rho(z) = \begin{cases} 1, & z \leq 0 \\ 0, & z \geq \rho \\ 1 - \frac{z}{\rho}, & 0 \leq z \leq \rho \end{cases}$$

We have the following facts for $\rho$.

**Fact 9.28.1.** *$\varphi_\rho$ is $\frac{1}{\rho}$-Lipschitz.*

For the following content, let's assume without loss of generality that $\rho = 1$ and $||x_i||_2 \leq 1$. Next, we want to derive the relation between margin and Rademacher complexity, so that we can know how well can margin generalize. We can have the following lemma.

**Lemma 9.29** (**Talagrad**)**.** *When $l$ is c-Lipschitz, we can have that*

$$\hat{\mathcal{R}}_S(l \circ \mathcal{H}_\Lambda) \leq c\hat{\mathcal{R}}_{S|x}(\mathcal{H}_\Lambda). \tag{9.23}$$

The last theorem in this note then can be derived.

**Theorem 9.30.** *For any data set $S$ and any $\Lambda$, we have that*

$$\hat{\mathcal{R}}_S(\mathcal{H}_\Lambda) \leq \frac{\Lambda}{\sqrt{n}}. \tag{9.24}$$

*Proof.*

$$\begin{aligned}
\hat{\mathcal{R}}_S(\mathcal{H}_\Lambda) &= \mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\frac{1}{n}\sup_{h_w\in\mathcal{H}_\Lambda}\sum_{i=1}^{m}\sigma_i h_w(x_i)] \\
&= \mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\frac{1}{n}\sup_{||w||_2\le\Lambda} w(\sum_{i=1}^{m}\sigma_i x_i)] \\
&\le \mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\frac{1}{n}\sup_{||w||_2\le\Lambda} ||w||_2||\sum_{i=1}^{m}\sigma_i x_i||_2] \\
&\le \mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\frac{\Lambda}{n}||\sum_{i=1}^{m}\sigma_i x_i||_2] \\
&\le \frac{\Lambda}{n}\sqrt{\mathbb{E}_{\sigma_1,\cdots,\sigma_n}[||\sum_{i=1}^{m}\sigma_i x_i||_2^2]} \\
&\le \frac{\Lambda}{n}\sqrt{\mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\sum_{i,j}\sigma_i\sigma_j x_i x_j]} \\
&\le \frac{\Lambda}{n}\sqrt{\mathbb{E}_{\sigma_1,\cdots,\sigma_n}[\sum_{i}\sigma_i^2 x_i^2]} \\
&= \frac{\Lambda}{n}\sqrt{\sum_{i} x_i^2} \\
&= \frac{\Lambda}{\sqrt{n}}.
\end{aligned}$$

□