1 Introduction

This course will be divided into 2 parts. In each part we will make different assumptions about the data generating process:

**Online Learning** No assumptions about data generating process. Worst case analysis. Fundamental connections to Game Theory.

**Statistical Learning** Assume data consists of independently and identically distributed examples drawn according to some fixed but unknown distribution.

Our examples will come from some space $X \times Y$. Given a data set
\[
\{(x_t, y_t)\}_{t=1}^{T} \in (X \times Y)^T,
\]
our goal is to predict $y_{T+1}$ for a new point $x_{T+1}$. A hypothesis is simply a function $h : X \rightarrow Y$. Sometimes, a hypothesis will map to a set $D$ (for decision space) larger than $Y$. Depending on the nature of the set $Y$, we get special cases of the general prediction problem.

**Binary classification** $Y = \{-1, +1\}$

**Multiclass classification** $Y = \{1, 2, \ldots, K\} =: [K]$ for $K \geq 3$

**Regression** $Y = [-B, B]$ or $Y = \mathbb{R}$

A set of hypotheses is often called a hypotheses class. If the range of a hypothesis is $\{-1, +1\}$ (or $\{0, 1\}$) then it also called a concept. A concept can be identified with the subset of $X$ on which it is 1.

2 Mistake Bound Model

In this model, learning proceeds in rounds, as we see examples one by one. Suppose $Y = \{-1, +1\}$. At the beginning of round $t$, the learning algorithm $A$ has the hypothesis $h_t$. In round $t$, we see $x_t$ and predict $h_t(x_t)$. At the end of the round, $y_t$ is revealed and $A$ makes a mistake if $h_t(x_t) \neq y_t$. The algorithm then updates its hypothesis to $h_{t+1}$ and this continues till time $T$.

Suppose the labels were actually produced by some function $f$ in a given concept class $C$. Then it is natural to bound the total number of mistakes the learner commits, no matter how long the sequence. To this end, define
\[
\text{mistake}(A, C) := \max_{f \in C, x_1:T} \sum_{t=1}^{T} 1[h_t(x_t) \neq f(x_t)].
\]

We can now define what it means for an algorithm to learn a class in the mistake bound model.

**Definition 2.1.** An algorithm $A$ learns a class $C$ with mistake bound $M$ iff
\[
\text{mistake}(A, C) \leq M.
\]
Note that we are ignoring efficiency issues here. We have not said anything about the amount of computation A has to do in each round in order to update its hypothesis from \( h_t \) to \( h_{t+1} \). Setting this issue aside for a moment, we have a remarkably simple algorithm \( \text{HALVING}(C) \) that has a mistake bound of \( \lg(|C|) \) for any finite concept class \( C \).

For a finite set \( \mathcal{H} \) of hypotheses, define the hypothesis majority \( \text{majority}(\mathcal{H}) \) as follows,

\[
\text{majority}(\mathcal{H}) (x) := \begin{cases} 
  +1 & \{|h \in \mathcal{H} \mid h(x) = +1\} \geq |\mathcal{H}|/2, \\
  -1 & \text{otherwise}.
\end{cases}
\]

**Algorithm 1** \( \text{HALVING}(C) \)

\[
\begin{align*}
C_1 &\leftarrow C \\
h_1 &\leftarrow \text{majority}(C_1) \\
\text{for } t = 1 \text{ to } T \text{ do} \\
&\quad \text{Receive } x_t \\
&\quad \text{Predict } h_t(x_t) \\
&\quad \text{Receive } y_t \\
&\quad C_{t+1} \leftarrow \{f \in C_t \mid f(x_t) = y_t\} \\
&\quad h_{t+1} \leftarrow \text{majority}(C_{t+1}) \\
\text{end for}
\end{align*}
\]

**Theorem 2.2.** For any finite concept class \( C \), we have

\[
\text{mistake}(\text{HALVING}(C), C) \leq \lg|C|.
\]

**Proof.** The key idea is that if the algorithm makes a mistake then at least half of the hypothesis in \( C_t \) are eliminated. Formally,

\[
h_t(x_t) \neq y_t \Rightarrow |C_{t+1}| \leq |C_t|/2.
\]

Therefore, denoting the number of mistakes up to time \( t \) by \( M_t \),

\[
M_t := \sum_{t=1}^{T} 1[h_t(x_t) \neq y_t],
\]

we have

\[
|C_{t+1}| \leq \frac{|C_1|}{2^M_t} = \frac{|C|}{2^M_t}. \tag{1}
\]

Since there is an \( f \in C \) which perfectly classifies all \( x_t \), we also have

\[
1 \leq |C_{t+1}|. \tag{2}
\]

Combining (1) and (2), we have

\[
1 \leq \frac{|C|}{2^M_t},
\]

which gives \( M_t \leq \lg(|C|) \).

\[ \square \]

### 3 Linear Classifiers and Margin

Let us now look at a concrete example of a concept class. Suppose \( \mathcal{X} = \mathbb{R}^d \) and we have a vector \( w \in \mathbb{R}^d \). We define the hypothesis,

\[
h_w(x) = \text{sgn}(w \cdot x),
\]
where $\text{sgn}(z) = 2 \cdot 1 [z \geq 0] - 1$ gives the sign of $z$. With some abuse of terminology, we will often speak of “the hypothesis $w$” when we actually mean “the hypothesis $h_w$”. The class of linear classifiers in the (uncountable) concept class

$$C_{\text{lin}} := \{h_w \mid w \in \mathbb{R}^d\}.$$  

Note that $w$ and $cw$ yield the same linear classifier for any $c > 0$.

Suppose we have a data set that is linearly separable. That is, there is a $w^*$ such that,

$$\forall t \in [T], \ y_t = \text{sgn}(w^* \cdot x_t).$$  

(3)

Separability means that $y_t(w^* \cdot x_t) > 0$ for all $t$. The minimum value of this quantity over the data set is referred to as the margin. Let us make the assumption that the margin is at least $\gamma$ for some $\gamma > 0$.

**Assumption M.** There exists a $w^* \in \mathbb{R}^d$ for which (3) holds. Further assume that

$$\min_{t \in [T]} y_t(w^* \cdot x_t) \geq \gamma,$$  

(4)

for some $\gamma > 0$.

Define

$$\|x_{1:t}\| := \max_{t \in [T]} \|x_t\|.$$  

We now show that under Assumption M, our simple halving algorithm can be used with a suitable finite subset of $C_{\text{lin}}$ to derive a mistake bound. Let $W_{\gamma}$ be those $w$ such that $w_i$ is of the form $m\gamma/2\|x_{1:T}\|d$ for some

$$m \in \{-2\|x_{1:T}\||w^*|d/\gamma, \ldots, -1, 0, +1, \ldots, 2\|x_{1:T}\||w^*|d/\gamma\}.$$  

In other words, since each coordinate of $w^*$ is in the range $[-\|w^*\|, \|w^*\|]$, we have discretized that interval at a scale of $\gamma/2\|x_{1:T}\|d$. We want to run the halving algorithm on the (finite) concept class,

$$C^\gamma_{\text{lin}} := \{h_w \mid w \in W_{\gamma}\}.$$  

The size of this class is $\left[\frac{4\|x_{1:T}\||w^*|d}{\gamma} + 1\right]^d$. Note that there exists a $\tilde{w} \in W_{\gamma}$ such that,

$$\forall i \in [d], \ |w^*_i - \tilde{w}_i| \leq \gamma/2\|x_{1:T}\|d.$$  

Thus, we have, for any $t \in [T],$

$$|y_t(\tilde{w} \cdot x_t) - y_t(w^* \cdot x_t)| = |\tilde{w} \cdot x_t - w^* \cdot x_t|$$

$$\leq \sum_{i=1}^d |\tilde{w}_i - w^*_i| \cdot |x_{t,i}|$$

$$\leq \sum_{i=1}^d \frac{\gamma}{2\|x_{1:T}\|d} \cdot \|x_t\|$$

$$\leq \gamma/2.$$  

This, together with Assumption R, implies that $y_t(\tilde{w} \cdot x_t) \geq \gamma/2 > 0$. Thus, there exists a hypothesis in $C^\gamma_{\text{lin}}$ that classifies the data set perfectly. Theorem 2.2 immediately gives the following corollary.

**Corollary 3.1.** Under Assumption M, HALVING $(C^\gamma_{\text{lin}})$ makes at most

$$d \lg \left(\left[\frac{4d\|x_{1:T}\||w^*|}{\gamma} + 1\right] \right)$$

mistakes.
This bound is nice because even though we had an uncountable concept class to begin with, the margin assumption allowed us to work with a finite subset of the concept class and we were able to derive a mistake bound. However, the result is unsatisfactory because running the halving algorithm on $\mathcal{C}_{\text{lin}}$ is extremely inefficient. One might wonder if one can use the special structure of the space of linear classifiers to implement the halving algorithm more efficiently. Indeed, it possible to implement a variant of the halving algorithm efficiently using the ellipsoid method developed for the linear programming feasibility problem.

Note that the mistake bound depends explicitly on the dimension $d$ of the problem. We would also like to be able to give a dimension independent mistake bound. Indeed, a classic algorithm called PERCEPTRON has such a mistake bound.