L15 – Week 8
Introduction to Statistical Learning Theory

CS 295 Optimization for Machine Learning
Ioannis Panageas
Linear Prediction

• Goal. Compute a vector $w$ that separates the two classes.
The Perceptron Algorithm

Given \((x_1, y_1), \ldots, (x_T, y_T) \in X \times \{\pm 1\}\) where we assume \(\|x_t\| = 1\) for all \(t\).

Formally \(\gamma\) is defined

\[
\gamma := \max_{w: \|w\| = 1} \min_{i \in [T]} (y_i w^\top x_i)_+,
\]

where \((a)_+ = \max(a, 0)\).
The Perceptron Algorithm

Given \((x_1, y_1), \ldots, (x_T, y_T) \in X \times \{\pm 1\}\) where we assume \(||x_t|| = 1\) for all \(t\).

Formally \(\gamma\) is defined

\[
\gamma := \max_{w: ||w||=1} \min_{i \in [T]} (y_i w^\top x_i)_+,
\]

where \((a)_+ = \max(a, 0)\).

**Definition (Perceptron).** Consider the following iterative algorithm:

1. Initialize \(w_1 = 0\) (hypothesis)
2. On round \(t=1 \ldots T\)
3. Consider \((x_t, y_t)\) and form prediction \(\hat{y}_t = \text{sign}(w_t^\top x_t)\).
4. If \(\hat{y}_t \neq y_t\)
5. \(w_{t+1} = w_t + y_t x_t\).
6. Else \(w_{t+1} = w_t\).
Analysis of Perceptron

Theorem (# Corrections). Perceptron makes at most $1/\gamma^2$ mistakes and corrections on any sequence with margin $\gamma$.

Proof. Let $m$ the number of mistakes after $T$ iterations. If a mistake is made at round $t$ then

$$\|w_{t+1}\|_2^2 = \|w_t + y_t x_t\|_2^2$$
Analysis of Perceptron

**Theorem (# Corrections).** Perceptron makes at most $1/\gamma^2$ mistakes and corrections on any sequence with margin $\gamma$.

**Proof.** Let $m$ the number of mistakes after $T$ iterations. If a mistake is made at round $t$ then

$$
\|w_{t+1}\|_2^2 = \|w_t + y_t x_t\|_2^2 = \|w_t\|_2^2 + \|x_t\|_2^2 + 2y_t x_t^T w_t
$$

$$
\text{negative}
$$
Analysis of Perceptron

**Theorem (# Corrections).** Perceptron makes at most $1/\gamma^2$ mistakes and corrections on any sequence with margin $\gamma$.

**Proof.** Let $m$ the number of mistakes after $T$ iterations. If a mistake is made at round $t$ then

$$
\|w_{t+1}\|_2^2 = \|w_t + y_t x_t\|_2^2 \\
= \|w_t\|_2^2 + \|x_t\|_2^2 + 2y_t x_t^\top w_t \\
\leq \|w_t\|_2^2 + 1
$$

Optimization for Machine Learning
Analysis of Perceptron

Theorem (# Corrections). Perceptron makes at most $1/\gamma^2$ mistakes and corrections on any sequence with margin $\gamma$.

Proof. Let $m$ the number of mistakes after $T$ iterations. If a mistake is made at round $t$ then

$$\|w_{t+1}\|_2^2 = \|w_t + y_t x_t\|_2^2$$

$$= \|w_t\|_2^2 + \|x_t\|_2^2 + 2y_t x_t^\top w_t$$

$$\leq \|w_t\|_2^2 + 1$$

Therefore $\|w_T\|_2^2 \leq m$. 
Analysis of Perceptron

Proof cont. Consider a vector $w^*$ with margin $\gamma$.

By definition of $\gamma$ for all $t$ that there is a mistake

$$\gamma \leq y_t w^* \top x_t = w^* \top (w_{t+1} - w_t).$$
Analysis of Perceptron

Proof cont. Consider a vector $w^*$ with margin $\gamma$.

By definition of $\gamma$ for all $t$ that there is a mistake

$$\gamma \leq y_t w^* \top x_t = w^* \top (w_{t+1} - w_t).$$

By adding the above we also have that

$$m\gamma \leq w^* \top (w_T - w_1) = w^* \top w_T,$$

$$\leq \|w_T\|_2.$$
Analysis of Perceptron

Proof cont. Consider a vector \( w^* \) with margin \( \gamma \).

By definition of \( \gamma \) for all \( t \) that there is a mistake

\[
\gamma \leq y_t w^* \top x_t = w^* \top (w_{t+1} - w_t).
\]

By adding the above we also have that

\[
m\gamma \leq w^* \top (w_T - w_1) = w^* \top w_T,
\]

\[
\leq \|w_T\|_2.
\]

Therefore \( m\gamma \leq \|w_T\|_2 \leq \sqrt{m}. \)
What we really showed is that given \((x_1, y_1), \ldots, (x_T, y_T) \in X \times \{\pm 1\}\) where we assume \(|x_t| = 1\) for all \(t\) it holds

\[
\sum_{t=1}^{T} 1_{y_t w_t^\top x_t \leq 0} \leq \frac{1}{\gamma^2}.
\]

Given \((x_1, y_1), \ldots, (x_n, y_n) \in X \times \{\pm 1\}\) IID from some distribution \(P\).

Run perceptron algorithm and consider \(w_1, \ldots, w_n\). Then choose \(w\) uniformly at random from \(\{w_1, \ldots, w_n\}\). This is good enough...
Random Data and 0-1 loss function

What we really showed is that given \((x_1, y_1), \ldots, (x_T, y_T) \in X \times \{\pm 1\}\)
where we assume \(||x_t|| = 1\) for all \(t\) it holds

\[
\sum_{t=1}^{T} 1_{y_t w_t^\top x_t \leq 0} \leq \frac{1}{\gamma^2}.
\]

Given \((x_1, y_1), \ldots, (x_n, y_n) \in X \times \{\pm 1\}\) IID from some distribution \(P\).

Run perceptron algorithm and consider \(w_1, \ldots, w_n\). Then choose \(w\) uniformly at random from \(\{w_1, \ldots, w_n\}\). This is good enough...

**Theorem (IID Data).** Let \(w\) be the choice of the algorithm. It holds that

\[
\mathbb{E} \left[ \frac{1}{n} \sum_{i=1}^{n} 1_{y_i w^\top x_i \leq 0} \right] \leq \frac{1}{n} \mathbb{E} \left[ \frac{1}{\gamma^2} \right].
\]

Optimization for Machine Learning
Random Data and 0-1 loss function

*Proof.* We have proved from before that (and taking expectations)

\[
\mathbb{E} \left[ \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{y_{i}w_{i}^{\top} x_{i} \leq 0} \right] \leq \mathbb{E} \left[ \frac{1}{n \gamma^2} \right].
\]

Let \( S = ((x_1, y_1), ..., (x_n, y_n)) \). The LHS can be expressed as

\[
\mathbb{E}_{\tau} \mathbb{E}_{S} \left[ \mathbf{1}_{y_{\tau}w_{\tau}^{\top} x_{\tau} \leq 0} \right] = \mathbb{E}_{S} \mathbb{E}_{\tau} \left[ \mathbf{1}_{y_{\tau}w_{\tau}^{\top} x_{\tau} \leq 0} \right].
\]
Random Data and 0-1 loss function

Proof. We have proved from before that (and taking expectations)

\[ \mathbb{E} \left[ \frac{1}{n} \sum_{i=1}^{n} 1_{y_i w^T x_i \leq 0} \right] \leq \mathbb{E} \left[ \frac{1}{n \gamma^2} \right] . \]

Let \( S = ((x_1, y_1), ..., (x_n, y_n)) \). The LHS can be expressed as

\[ \mathbb{E}_\tau \mathbb{E}_S \left[ 1_{y_\tau w^\tau x_\tau \leq 0} \right] = \mathbb{E}_S \mathbb{E}_\tau \left[ 1_{y_\tau w^\tau x_\tau \leq 0} \right] . \]

Observe now that \( w_\tau \) depends only on \((x_1, y_1), ..., (x_{\tau-1}, y_{\tau-1})\), hence

\[ \mathbb{E}_S \mathbb{E}_\tau \left[ 1_{y_\tau w^\tau x_\tau \leq 0} \right] = \mathbb{E}_S \mathbb{E}_\tau \mathbb{E}_{(x,y) \sim P} \left[ 1_{y w^\tau x \leq 0} \right] = \mathbb{E}_S \mathbb{E}_\tau [L_{0-1}(w_\tau)] \]

Remark: If we keep iterating perceptron algorithm we finally get \( L_{0-1}(w_T) = 0 \) (how many steps?) where

\[ L_{0-1}(w) = \frac{1}{n} \sum_i 1_{y_i w^T x_i \leq 0} \]
PAC Learning

Assume we are given:

- Domain set $\mathcal{X}$. Typically $\mathbb{R}^d$ or $\{0, 1\}^d$. Think of 32x32 pixel images.
- Label set $\mathcal{Y}$, typically binary like $\{0, 1\}$ or $\{-1, +1\}$.
- A concept class $\mathcal{C} = \{h : h : \mathcal{X} \to \mathcal{Y}\}$.

Given a learning problem, we analyse the performance of a learning algorithm:

- Training data $S = (x_1, y_1), ..., (x_m, y_m)$, where sample $S$ was generated by drawing $m$ IID samples from the distribution $D$.
- Output a hypothesis from a hypothesis class $\mathcal{H} = \{h : h : \mathcal{X} \to \mathcal{Y}\}$ of target functions.
PAC Learning

Assume we are given:

- Domain set \( \mathcal{X} \). Typically \( \mathbb{R}^d \) or \( \{0, 1\}^d \). Think of 32x32 pixel images.
- Label set \( \mathcal{Y} \), typically binary like \( \{0, 1\} \) or \( \{-1, +1\} \).
- A concept class \( \mathcal{C} = \{h : h : \mathcal{X} \to \mathcal{Y}\} \).

Given a learning problem, we analyse the performance of a learning algorithm:

- Training data \( S = (x_1, y_1), ..., (x_m, y_m) \), where sample \( S \) was generated by drawing \( m \) IID samples from the distribution \( D \).
- Output a hypothesis from a hypothesis class \( \mathcal{H} = \{h : h : \mathcal{X} \to \mathcal{Y}\} \) of target functions.

We measure the performance through \textbf{generalization error} that is

\[
\text{err}(h) = \mathbb{E}_{(x, y) \sim D}[\ell_{0-1}(h(x), y)].
\]
PAC Learning

**Definition (PAC learnable).** A concept class $C$ of target functions is PAC learnable (w.r.t to $H$) if there exists an algorithm $A$ and function $m^A_C : (0, 1)^2 \rightarrow \mathbb{N}$ with the following property:

Assume $S = ((x_1, y_1), ..., (x_m, y_m))$ is a sample of IID examples generated by some arbitrary distribution $D$ such that $y_i = h(x_i)$ for some $h \in C$ almost surely. If $S$ is the input of $A$ and $m > m^A_C$ then the algorithm returns a hypothesis $h_S \in H$ such that, with probability $1 - \delta$ (over the choice of the $m$ training examples):\[
\text{err}(h_S) < \epsilon
\]

The function $m^A_C$ is referred to as the **sample complexity** of algorithm $A$. 

Optimization for Machine Learning
Example 2.2 (Half-spaces). A second example that is of some importance is defined by hyperplane. Here we let the domain be $\chi = \mathbb{R}^d$ for some integer $d$. For every $w \in \mathbb{R}^d$, induces a half space by consider all elements $x$ such that $w \cdot x \geq 0$. Thus, we may consider the class of target functions described as follows

$$C = \{f_w : w \in \mathbb{R}^d, f_w(x) = \text{sign}(w \cdot x)\}$$
Examples

Example 2.1 (Axis Aligned Rectangles). The first example of a hypothesis class will be of rectangles aligned to the axis. Here we take the domain $\chi = \mathbb{R}^2$ and we let $\mathcal{C}$ include be defined by all rectangles that are aligned to the axis. Namely for every $(z_1, z_2, z_3, z_4)$ consider the following function over the plane

$$f_{z_1, z_2, z_3, z_4}(x_1, x_2) = \begin{cases} 
1 & z_1 \leq x_1 \leq z_2, \ z_3 \leq x_2 \leq z_4 \\
0 & \text{else}
\end{cases}$$

Then $\mathcal{C} = \{f_{z_1, z_2, z_3, z_4} : (z_1, z_2, z_3, z_4) \in \mathbb{R}^4\}$. 
ERM algorithm

Definition (ERM). Empirical Risk Minimization algorithm is defined as follows:

Return

\[
\arg \min_{h \in \mathcal{H}} \text{err}_S(h),
\]

where \( \text{err}_S(h) = \frac{1}{m} \sum \ell_{0-1}(h(x_i), y_i) \)

Theorem (Finite classes are PAC learnable). Consider a finite class of target functions \( \mathcal{H} = h_1, ..., h_t \) over a domain. Then if size of sample \( S \) is \( m > \frac{2}{\epsilon^2} \log \frac{2|\mathcal{H}|}{\delta} \) then with probability \( 1 - \delta \) we have that

\[
\max_{h \in \mathcal{H}} |\text{err}_S(h) - \text{err}(h)| < \epsilon.
\]
ERM algorithm analysis

Proof. Applying Hoeffding’s inequality we obtain that for every $S$ and fixed $h$ since $\text{err}_S(h)$ is sum of IID bernoulli with expectation $\text{err}(h)$:

$$\mathbb{P}_S[|\text{err}_S(h) - \text{err}(h)| > \epsilon] \leq 2e^{-2m\epsilon^2}.$$
Proof. Applying Hoeffding’s inequality we obtain that for every $S$ and fixed $h$ since $\text{err}_S(h)$ is sum of IID bernoulli with expectation $\text{err}(h)$:

$$\Pr_S[|\text{err}_S(h) - \text{err}(h)| > \varepsilon] \leq 2e^{-2\varepsilon^2}.$$  

Applying union bound we obtain that

$$\Pr_S[\exists h : |\text{err}_S(h) - \text{err}(h)| > \varepsilon] \leq 2|\mathcal{H}|e^{-2\varepsilon^2}.$$
ERM algorithm analysis

Proof. Applying Hoeffding’s inequality we obtain that for every $S$ and fixed $h$ since $\text{err}_S(h)$ is sum of IID bernoulli with expectation $\text{err}(h)$:

$$\Pr_S[|\text{err}_S(h) - \text{err}(h)| > \varepsilon] \leq 2e^{-2\varepsilon^2m^2}.$$

Applying union bound we obtain that

$$\Pr_S[\exists h : |\text{err}_S(h) - \text{err}(h)| > \varepsilon] \leq 2|\mathcal{H}|e^{-2\varepsilon^2m^2}.$$

We want the RHS to be less than $\delta$. Choose $m$ appropriately!
ERM algorithm analysis

Proof. Applying Hoeffding’s inequality we obtain that for every $S$ and fixed $h$ since $\text{err}_S(h)$ is sum of IID bernoulli with expectation $\text{err}(h)$:

$$\Pr[S : |\text{err}_S(h) - \text{err}(h)| > \epsilon] \leq 2e^{-2me^2}.$$ 

Applying union bound we obtain that

$$\Pr[S : \exists h : |\text{err}_S(h) - \text{err}(h)| > \epsilon] \leq 2|\mathcal{H}|e^{-2me^2}.$$ 

We want the RHS to be less than $\delta$. Choose $m$ appropriately!

What if the hypothesis class has infinite cardinality?
Conclusion

• Introduction to Statistical Learning.
  – Perceptron Algorithm.
  – Loss functions and PAC learning
  – ERM algorithm

• Next lecture we will talk about VC dimension.