Today’s Lecture: Generalization

- I have a training set \( \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \)

- I choose some classifier (e.g., a linear classifier \( \text{sign}(\theta \cdot x) \)) on the basis of this training set

- How well will the classifier perform on test examples?
Preliminaries: Chernoff Bounds

Let $X_1, X_2, \ldots X_n$ be $n$ independent, identically distributed Bernoulli random variables with

$$P(X_i = 1) = p, \quad P(X_i = 0) = 1 - p$$

Define

$$\hat{p} = \frac{\sum_{i=1}^{n} X_i}{n}$$

($\hat{p}$ is a natural estimate of $p$)

Let $\gamma > 0$ be any constant. Then

$$P(|p - \hat{p}| > \gamma) \leq 2e^{-2n\gamma^2}$$
Preliminaries: The Union Bound

Let $A_1, A_2, \ldots, A_k$ be $k$ events (that may not be independent). Then

$$
P(A_1 \cup A_2 \cup \ldots \cup A_k) \leq P(A_1) + P(A_2) + \ldots + P(A_k)
$$
Hypothesis Classes/Empirical Risk Minimization

A hypothesis class $\mathcal{H}$ is a set of classifiers that we are considering. One example is the set of all hyperplane classifiers:

$$\mathcal{H} = \{\text{sign}(\theta \cdot x) : \theta \in \mathbb{R}^d\}$$

Given a training set $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, define the empirical risk (or training error) for any $h \in \mathcal{H}$ to be

$$\hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} [[h(x_i) \neq y_i]]$$

where $[[\pi]] = 1$ if $\pi$ is true, 0 otherwise.

In Empirical Risk Minimization, we choose

$$h_{\text{ERM}} = \arg\min_{h \in \mathcal{H}} \hat{R}(h)$$

Question: how well does this method generalize to test examples?
The IID Assumption

- It is critical to make some assumption that training and test examples are related.

- To this end, we assume that training examples \((x_i, y_i)\) are independent, identically distributed random variables, drawn from some underlying distribution \(D\). We don’t know \(D\), our only evidence about \(D\) is the training set.

- Test examples are also drawn from the distribution \(D\).

- The generalization error (probability of error on a new test example) is

\[
R(h) = P(h(x) \neq y)
\]

where the pair \((x, y)\) has distribution \(D\).

- Clearly, we’d like to choose some \(h \in \mathcal{H}\) such that \(R(h)\) is as small as possible.
Finite Hypothesis Classes

- For now we’ll assume that $\mathcal{H}$ is finite, and consists of $k$ hypotheses,
  \[ \mathcal{H} = \{h_1, h_2, \ldots, h_k\} \]

- Later, we’ll consider
  1. Infinite hypothesis classes (VC dimension)
  2. Large margin classifiers
Applying Chernoff Bounds

Consider any classifier $h \in \mathcal{H}$. We have $R(h) = P(h(x) \neq y)$, and

$$\hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} [[h(x_i) \neq y_i]] = \frac{1}{n} \sum_{i=1}^{n} Z_i$$

where $Z_i = [[h(x_i) \neq y_i]]$

Each $Z_i$ is a Bernoulli random variable with $P(Z_i = 1) = R(h)$, and $P(Z_i = 0) = 1 - R(h)$

By Chernoff bounds, for any $\gamma > 0$,

$$P(|R(h) - \hat{R}(h)| > \gamma) \leq 2e^{-2n\gamma^2}$$
Uniform Convergence Bounds

- Fix some constant $\gamma$. For any $j$, let $A_j$ be the event that $|R(h_j) - \hat{R}(h_j)| > \gamma$. By Chernoff bounds we have $P(A_j) \leq 2e^{-2n\gamma^2}$ for all $j$.

- It follows that

$$P(\exists h \in \mathcal{H}. |R(h) - \hat{R}(h)| > \gamma) = P(A_1 \cup A_2 \cup \ldots \cup A_k) \leq \sum_{j=1}^{k} P(A_j) \quad \text{(Union bound)}$$

$$\leq \sum_{j=1}^{k} 2e^{-2n\gamma^2} \quad \text{(Chernoff)}$$

$$= 2ke^{-2n\gamma^2}$$

- Conversely, for any $\gamma > 0$,

$$P(\forall h \in \mathcal{H} \ |R(h) - \hat{R}(h)| \leq \gamma) \geq 1 - 2ke^{-2n\gamma^2}$$
An Example

Given some value $\gamma > 0$, and some value $\delta > 0$, how large must $n$ be before we can guarantee that with probability at least $1 - \delta$, we have

$$\mathbb{P}(\forall h \in \mathcal{H}. |R(h) - \hat{R}(h)| \leq \gamma) \geq 1 - \delta$$

Answer: choose $n$ such that

$$n \geq \frac{1}{2\gamma^2} \log \frac{2k}{\delta}$$

This function of $k, \gamma, \delta$ is often called the sample complexity of learning.
A Second Example

- We have $P(\exists h \in \mathcal{H} . \ |R(h) - \hat{R}(h)| > \gamma) \leq 2ke^{-2n\gamma^2}$

- If we set $\delta = 2ke^{-2n\gamma^2}$ then

$$
\gamma = \sqrt{\frac{1}{2n} \log \frac{2k}{\delta}}
$$

hence for any $\delta > 0$,

$$
P \left( \exists h \in \mathcal{H} . \ |R(h) - \hat{R}(h)| > \sqrt{\frac{1}{2n} \log \frac{2k}{\delta}} \right) \leq \delta
$$

- **Theorem.** Let $|\mathcal{H}| = k$, and let any $n, \delta > 0$ be fixed. Then with probability at least $1 - \delta$, we have that for all $h \in \mathcal{H}$,

$$
|R(h) - \hat{R}(h)| \leq \sqrt{\frac{1}{2n} \log \frac{2k}{\delta}}
$$
What can we prove about \( h_{\text{ERM}} \)?

- **Recap:** \( h_{\text{ERM}} = \arg\min_{h \in \mathcal{H}} \hat{R}(h) \)

- **Define** \( h^* = \arg\min_{h \in \mathcal{H}} R(h) \) (the best hypothesis in \( h \))

- **If for all** \( h \in \mathcal{H}, |R(h) - \hat{R}(h)| \leq \gamma \), it follows that

\[
R(h_{\text{ERM}}) \leq \hat{R}(h_{\text{ERM}}) + \gamma \leq \hat{R}(h^*) + \gamma \leq R(h^*) + 2\gamma
\]

- **Theorem.** Let \( |\mathcal{H}| = k \), and let any \( n, \delta > 0 \) be fixed. Then with probability at least \( 1 - \delta \), we have that

\[
R(h_{\text{ERM}}) \leq \left( \min_{h \in \mathcal{H}} R(h) \right) + 2\sqrt{\frac{1}{2n} \log \frac{2k}{\delta}}
\]
An Example: Disjunctions

- Assume inputs are binary vectors with $d$ dimensions, i.e., $\mathbf{x} \in \{0, 1\}^d$

- Consider a simple hypothesis class containing hypotheses of the form

\[
\text{if } (x_3 = 1) \text{ or } (x_{100} = 1) \text{ or } (x_{500} = 1) \quad \text{then } y = +1 \\
\text{else } y = -1
\]

- Each hypothesis is a disjunction of $r$ variables (in this case $r = 3$)

- In this case we have

\[
|\mathcal{H}| = k = \binom{d}{r} \leq d^r \quad \Rightarrow \quad \log k \leq r \log d
\]

hence we need $n = O(r \log d)$ training samples for learning
An Example: All Possible Boolean Functions

- Assume inputs are binary vectors with $d$ dimensions, i.e., $x \in \{0, 1\}^d$

- Take $\mathcal{H}$ to be the hypothesis class of all possible boolean functions over $d$ variables

- In this case

$$|\mathcal{H}| = k = 2^{(2^d)} \Rightarrow \log k = 2^d \log 2$$

and we need $n = O(2^d \log 2)$ training examples for learning
Decision Trees
Decision Trees of Bounded Depth

- Assume inputs are binary vectors with $d$ dimensions, i.e., $x \in \{0, 1\}^d$
- Let $N_r$ be the number of decision trees with depth $r$
- We have $N_0 = 2$
- To calculate $N_{r+1}$, note that for a tree of depth $r + 1$, we have:
  1. $d$ choices of the variable at the root
  2. $N_r$ choices of the left sub-tree
  3. $N_r$ choices of the right sub-tree

Hence $N_{r+1} = d \times N_r \times N_r$

- Defining $L_r = \log_2 N_r$, we get $L_0 = 1$, and $L_{r+1} = \log_2 d + 2L_r$
  $\Rightarrow L_r = (2^r - 1)(1 + \log_2 d) + 1$