EECS 598-005: Theoretical Foundations of Machine Learning **Fall 2015**

Lecture 5: Typical Machine Learning Problem and PAC Learning

Lecturer: Jacob Abernethy *Scribes: Frank Cheng*

Editors: Luke Brandl, Nghia Nguyen, and Shuang Qiu

5.1 Review: Hoeffding's inequality (simplified)

If $x_1, \ldots, x_n \in [0,1]$ are independent random variables, then **Hoeffding's inequality** states that:

$$\mathbf{Pr}\left(\frac{1}{n}\sum_{i=1}^{n} x_i - E\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right) \geq t\right) \leq \exp(-2t^2 n)$$

The left hand side dies off exponentially quickly as t increases. Note that if we want to get a bound in terms of the absolute value of the deviation then we get a probability bound that increases by a multiple of two. We can also solve for t in terms of a given probability δ:

$$\begin{aligned} \delta &\geq 2\exp(-2t^2 n) \\ \implies \log\left(\frac{2}{\delta}\right) &\leq 2t^2 n \\ \implies t &\geq \sqrt{\frac{\log(2/\delta)}{2n}} \end{aligned}$$

Fact 5.1. *With probability $1-\delta$ we have:*

$$\left\| \frac{1}{n}\sum_i x_i - E\left(\frac{1}{n}\sum_i x_i\right)\right\| \leq \sqrt{\frac{\log(2/\delta)}{2n}}$$

5.2 One more deviation bound

We want to ensure by taking enough random samples that some event does not occur less than than ϵ. Formally, let $x_i \in \{0,1\}$ with $\mathbf{Pr}(x_i = 1) \geq \epsilon$. What is the probability that $\sum_{i=1}^n x_i = 0$? We know that

$$\prod_{i=1}^{n}(\mathbf{Pr}(x_i = 0)) \leq (1-\epsilon)^n = \exp(n\log(1-\epsilon))$$

Since log is a concave function, $\log(1+x) \leq x$ for any $x \in \mathbb{R}$. So $\exp(n\log(1-\epsilon)) \leq e^{-n\epsilon}$.

Fact 5.2. *If $n \geq \frac{\log(1/\delta)}{\epsilon}$ then with probability $1-\delta$, $\sum_{i=1}^n x_i \neq 0$.*

Note that since $x^2 < x$ for small, positive values of x, this is a tighter lower bound on n than the one given by Hoeffding's inequality for small ϵ.

5.3 Sketch of a typical machine learning problem and support vector machines

In a linear classification problem, we are given data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ independently and identically from a distribution D. Here $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$. We want to find $\mathbf{w} \in \mathbb{R}^d$, a weight coefficient vector such that $\mathbf{Pr}(\text{sgn}(\mathbf{w}^\top \mathbf{x}) \neq y)$ is small for all future $(\mathbf{x}, y) \sim D$. One way to find $\mathbf{w}$ is by solving the following maximization problem:

$$\arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i)) + \lambda \frac{\|\mathbf{w}\|^2}{2}$$

where $\lambda \in \mathbb{R}$ is a chosen parameter. The function within the arg min term is the **support vector machine's** loss function, defined as the **hinge loss**.

Definition 5.3 (Training Error)**.** *The* ***training error****, written as* $\text{err}_n(\mathbf{w})$*, is:*

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left[\text{sgn}\left(\mathbf{w}^\top \mathbf{x}_i\right) \neq y_i\right]$$

The hinge loss function is an approximation of the training error. Using the hinge loss function or otherwise, pick some $\mathbf{w} \in \mathbb{R}^d$ such that $\text{err}_n(\mathbf{w}) \leq \epsilon$. How do we measure the performance of the model? First, a definition:

Definition 5.4 (Ideal Test Error)**.** *The* ***test error****, written as* $\overline{\text{err}}(\mathbf{w})$*, is*

$$\mathbb{E}\left(\mathbb{1}\left[\text{sgn}\left(\mathbf{w}^\top \mathbf{x}\right) \neq y\right]\right) = \mathbf{Pr}\left((\mathbf{w}^\top \mathbf{x})y \leq 0\right)$$

where the expectation and probability are taken over distribution D*,* $(\mathbf{x}, y) \sim D$*.*

A good model has small ideal test error. An *erroneous* approach is as follows. Pick $\hat{\mathbf{w}}_n = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \mathbb{1}\left[(\mathbf{w}^\top \mathbf{x}_i)y_i \leq 0\right]$. Apply Hoeffding's inequality:

$$I_i = \mathbb{1}\left[(\mathbf{w}^\top \mathbf{x}_i)y \leq 0\right]$$
$$|\text{err}_n(\mathbf{w}) - \overline{\text{err}}(\mathbf{w})| = \left|\frac{1}{n}\sum_{i=1}^{n} I_i - \mathbb{E}(\mathbf{I})\right| \leq \sqrt{\frac{\log(2/\delta)}{2n}}$$

with probability $1 - \delta$. This argument is **false** because I_i are intentionally correlated to fit the data. They are no longer independent, so the bound cannot be used.

5.4 PAC-Learning: "Probably Approximately Correct"

Key pieces:

- $\mathbb{X}$ input space
- Output space $Y = \{0, 1\}$
- Concept class $\mathbb{C}$, a set of function families taking $\mathbb{X}$ to Y.

Here $\mathbb{C}$ can be viewed as part of $P(\mathbb{X})$, the power set of $\mathbb{X}$.

Definition 5.5. *A* ***learning instance*** *consists of:*

- *A distribution* $D \in \Delta(\mathbb{X})$.
- *A target concept* $c \in \mathbb{C}$.

Our goal is to have an algorithm $\mathbb{A}$ that maps a collection of learning instances to a hypothesis $h : \mathbb{X} \to Y$, hopefully with $\mathbf{Pr}_{x \sim D}(h(\mathbf{x}) \neq c(\mathbf{x})) \leq \epsilon$.

Definition 5.6 (Risk)**.** *Given* $D \in \Delta(\mathbb{X})$ *and target* $c \in \mathbb{C}$*, the* ***risk*** *of* h*, a function from* $\mathbb{X}$ *to* Y*, is:*

$$R(h) = \mathbb{E}\left[\mathbb{1}(h(\mathbf{x}) \neq c(\mathbf{x}))\right] = \mathop{\mathbf{Pr}}_{x \sim D}(h(\mathbf{x}) \neq c(\mathbf{x}))$$

where R *depends on* D *and* c*. This is also known as the* ***generalization error****.*

Definition 5.7 (Empirical Risk)**.** *The* ***empirical risk*** *on* $\mathbf{x}_1, \ldots, \mathbf{x}_n$ *is defined as:*

$$\hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left[h(\mathbf{x}_i) \neq c(\mathbf{x}_i)\right]$$