CS281B/Stat241B (Spring 2006) Statistical Learning Theory **Lecture: 1**

Pattern Classification and Perceptron Algorithm

Lecturer: Peter Bartlett *Scribe: Xia Jiang*

1 Outline

We will focus on three theoretical aspects in CS 281B/STAT 241B: prediction methods, such as pattern classification; algorithms, such as kernel methods and ensemble methods; and performance estimates, which will be based on concentration of measure, empirical process theory, regularize and model selection.

In lecture 1, we will discuss the probabilistic formulation of prediction problems and the perceptron algorithm.

2 Prediction Problems

Let $\mathcal{X}$ be the *observation space* (input space), and be the $\mathcal{Y}$ *outcome space* (output space). For observation space, $x \in \mathcal{X}$. For outcome space $y \in \mathcal{Y}$. Our goal is to access to a data set $(X_1, Y_1), (X_2, Y_2)...(X_n, Y_n)$. We aim to find a function $f : \mathcal{X} \to \mathcal{Y}$, thus that $f(X)$ accurately predicts Y.

Define *loss function* by $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathcal{R}$. For pattern classification problem, we will classify a finite number of distinct classes, $|\mathcal{Y}| \leq \infty$, and we define loss function as the indicator function below:

$$\ell(\hat{y}, y) = \mathbf{1}[\hat{y} \neq y] = \begin{cases} 1 & \text{if } \hat{y} \neq y \\ 0 & \text{otherwise} \end{cases}$$

We incur a loss of 1 if we assign the class label incorrectly. For regression problems, we use the loss function $\ell(\hat{y}, y) = (\hat{y} - y)^2$.

We assume there is a probability distribution P on $\mathcal{X} \times \mathcal{Y}$, which $(X_1, Y_1), (X_2, Y_2)...(X_n, Y_n), (X, Y)$ are iid draws from P. We need to choose a function $f : \mathcal{X} \to \mathcal{Y}$, such that, the risk of f is small, that is,

$$L(f) = \mathbb{E}\ell(f(X), Y)$$

is small, or $L(f_n)$ is small with high probability. Because $x \mapsto f_n(X, X_1, Y_1, ..., X_n, Y_n)$ is random (depends the data we use) f_n, as well as $L(f_n) = \mathbb{E}[\ell(f_n(X; X_1, Y_1, ...X_n, Y_n), Y)|(X_1, Y_1, ..., X_n, Y_n)]$, are random variables.

Let L^* denote the minimal risk of function f:

$$L^* = \inf_f L(f)$$

where the infimum is over all measurable f.
For binary classification with $\mathcal{Y} = \{\pm 1\}$, $L^* = L(f^*)$, where $f^*(x) = \arg\max_{y \in \mathcal{Y}} P(Y = y|X = x)$. We are interested in the following questions: Can we ensure that $L(f_n)$ if close to L^*? Does $\mathbb{E}L(f_n) \xrightarrow{P} L^*$ hold for all data generating distribution P? Can we estimate $L(f_n)$? We will explore these questions in the next few lectures.

3 Pattern Classification

We just consider the binary classification case (where $\mathcal{Y} = \{\pm 1\}$) for simplicity. The distribution P on $\mathcal{X}$ x $\mathcal{Y}$ be represented by (μ, η), where μ = marginal distribution on $\mathcal{X}$, $\eta(x) = P(Y = 1|X = x)$.

Recall that

$$f^*(x) = \arg\max_{y \in \mathcal{Y}} P(Y = y|x) = \begin{cases} 1 & \text{if } \eta(x) \geq 1/2 \\ -1 & \text{otherwise} \end{cases}$$

Theorem 3.1. For any $f : \mathcal{X} \to \mathcal{Y}$

$$\begin{aligned} L(f) - L^* &= L(f) - L(f^*) \\ &= \mathbb{E}[\mathbf{1}(f(x) \neq f^*(x))|2\eta(x) - 1|] \geq 0 \end{aligned}$$

Proof.

$$\begin{aligned} L(f) - L^* &= \mathbb{E}_{X,Y}(\mathbf{1}[f(X) \neq Y] - \mathbf{1}[f^*(X) \neq Y]) \\ &= \mathbb{E}_{X,Y}(\mathbf{1}[f(X) \neq f^*(X)](\mathbf{1}[f(X) \neq Y] - \mathbf{1}[f^*(X) \neq Y])) \\ &= \mathbb{E}_{X,Y}(\mathbf{1}[f(X) \neq f^*(X)](2 \cdot \mathbf{1}[f^*(X) = Y] - 1)) \\ &= \mathbb{E}_X(\mathbf{1}[f(X) \neq f^*(X)](2\mathbb{E}[f^*(X) = Y|X] - 1)) \\ &= \mathbb{E}_X(\mathbf{1}[f(X) \neq f^*(X)] \cdot [2(1/2 + |\eta(X) - 1/2|) - 1]) \\ &= \mathbb{E}_X[\mathbf{1}(f(x) \neq f^*(x))|2\eta(x) - 1|] \end{aligned}$$

□

Notice that the increase in risk depends on how far the conditional probability, η, is from 1/2 on the set where f and f^* disagree.

One approach to minimizing risk is to estimate the regression function η with some $\hat{\eta}$, such that $\hat{\eta}$: $\mathcal{X} \to [0, 1]$ and use the following classification function :

$$f_{\hat{\eta}}(x) = \begin{cases} 1 & \text{if } \hat{\eta}(x) \geq 1/2 \\ -1 & \text{otherwise} \end{cases}$$

This is the *plug-in decision function.* The following theorem gives an upper bound of $L(f_{\hat{\eta}}) - L(f^*)$.

Corollary 3.2. $L(f_{\hat{\eta}}) - L^* \leq 2\mathbb{E}|\eta(x) - \hat{\eta}(x)|$.

Proof.

$$\begin{aligned} L(f_{\hat{\eta}}) - L^* &= \mathbb{E}[\mathbf{1}(f(x) \neq f^*(x))|2\eta(x) - 1|] \\ &= 2\,\mathbb{E}_X[\mathbf{1}(f(x) \neq f^*(x))|\eta(x) - 1/2|] \\ &\leq 2\,\mathbb{E}_X[\mathbf{1}(f(x) \neq f^*(x))|\hat{\eta}(x) - \eta(x)|] \end{aligned}$$

since $f_{\hat{\eta}}(x) \neq f^*(x)$ implies that

$$\begin{aligned} |\eta(x) - \hat{\eta}(x)| &= |\hat{\eta}(x) - 1/2| + |\eta(x) - 1/2| \\ &\geq |\eta(x) - 1/2| \end{aligned}$$

□

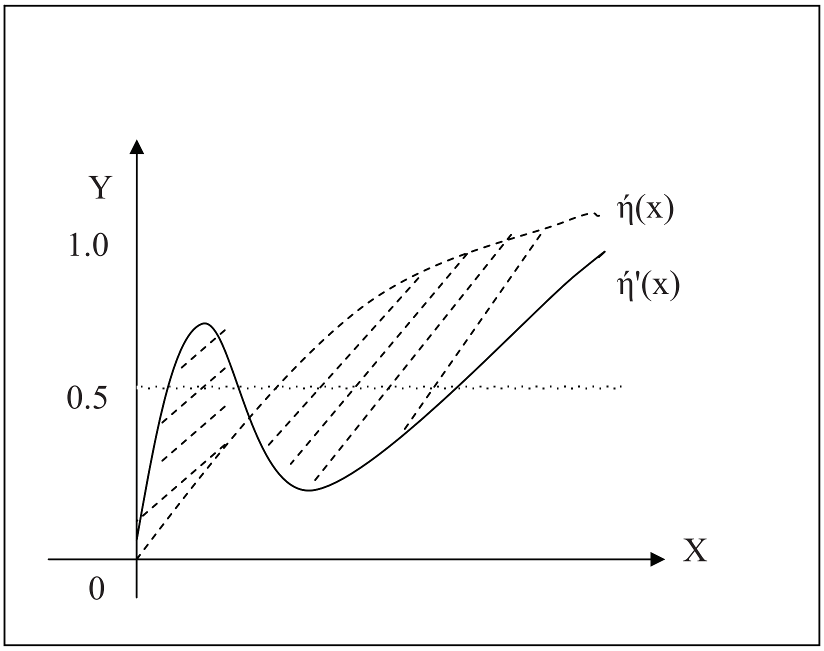

Figure 1: The difference between optimal loss function and plug in loss function is less than twice of the shaded area.

Thus the difference between the risk of the plug-in decision function and the optimal risk is less than twice of the difference between η and $\hat{\eta}$ (see Figure 1).

Note. Estimating η accurately gives a low risk. However, estimating η is not necessary and it might be more difficult than the problem of finding an f with small risk.

An alternative approach is to fix $\mathcal{F}$, a set of functions $f : \mathcal{X} \to \{\pm 1\}$, then choose $f_n(X; X_1, Y_1, ..., X_n, Y_n) = f(X)$. For example in perceptron algorithm, for $\mathcal{X} = R^d$, choose a set of function $\mathcal{F} = \{x \mapsto \text{sign}(w^T x) : w \in R^d\}$ which are linear threshold functions with decision regions as the half spaces.

4 Perceptron Algorithm

This algorithm tries to classify the data using the class of linear threshold functions. Given linearly separable training data $(x_1, y_1), ..., (x_n, y_n)$, the algorithm will adjust w such that a hyperplane will split the data into two classes correctly.

Algorithm 1 (Perceptron Algorithm).

1. Given$(x_1, y_1), ..., (x_n, y_n) \in \mathcal{X} \times \{\pm 1\}$
2. $w_0 = 0, t = 0$
3. **While** $\exists i \in \{1, ..n\}$ s.t. $y_i \neq \text{sign}(w_t^T x_i)$ **do**
   Pick $i \in 1, ..., n$
   such that $y_i \neq \text{sign}(w_t^T x_i)$
   $$w_{t+1} = w_t + y_i x_i$$

$t = t + 1$

4. **End**

The intuition behind this algorithm is that whenever there is a wrong classification, it rotates w_t such that x_i is closer to being classified correctly.