

Training and Testing Error in Classification

Hossein Falaki
Computer Science Department,
University of California, Los Angeles
falaki@cs.ucla.edu

1 Introduction

In this note we will study the training and testing error of classification. We will use *Perceptron* as a sample classifier and look into the concepts of *VC dimension*, and the relation between training and testing error with respect to the VC dimension.

2 Perceptron

Consider the assumptions presented in Table 1. Suppose y_i values are either +1 or -1. The Perceptron classifier is defined as follows:

$$f(X) = \text{sgn}(\beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p) \quad (1)$$

Where:

$$\text{sgn}(r) = \begin{cases} +1 & \text{if } r \geq 0 \\ -1 & \text{if } r < 0 \end{cases}$$

In essence the perceptron is dividing positive and negative values with hyper planes in the p-dimensional space of X_1, \dots, X_p .

Definition 2.1. VC-dimension: *Vapnik-Chervonenkis dimension is the number of totally random (coin flips) points that a classifier can correctly explain.*

For example in a two dimensional space, the maximum number of random points that a line can separate is three. Therefore the VC dimension of this setting is three.

3 Training vs. Testing Error

We can define the training error of a classifier in the following way:

$$\text{Training Error} = \frac{1}{n} \sum_{i=1}^n 1_{y_i \neq \text{sgn}(x_i \beta)} \quad (2)$$

observations	response	features			
1	y_1	x_{11}	x_{12}	\dots	x_{1p}
2	y_2	x_{21}	x_{22}	\dots	x_{2p}
\vdots	\vdots			\vdots	
n	y_n	x_{n1}	x_{n2}	\dots	x_{np}

Table 1: Assumptions in a perceptron

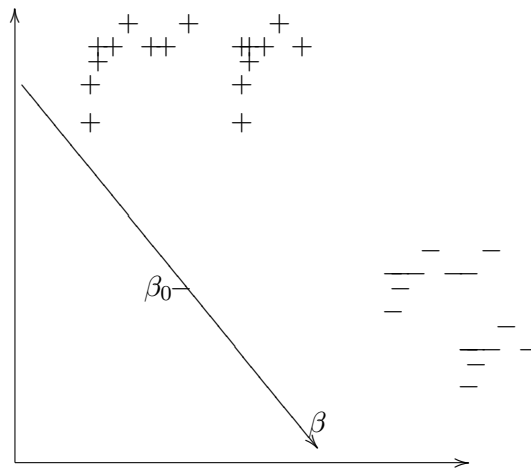


Figure 1: Visualization of a perceptron

This is also referred to as the *training risk* ($R(\beta)$).

The training error can be re-written as the expectation of error over an empirical uniform distribution.

$$\hat{p} = \frac{1}{n} \sum_{i=1}^n \delta_{x_i, y_i}$$

$$\begin{aligned} E_{\hat{p}}(1_{Y \neq \text{sgn}(X\beta)}) &= \frac{1}{n} \left(\sum_{i=1}^n 1_{y_i \neq \text{sgn}(x_i\beta)} \right) \\ &= R_{emp}(\beta) \end{aligned}$$

Similarly testing error can be defined as:

$$\text{Testing Error} = E_p(1_{Y \neq \text{sgn}(X\beta)}) = R(\beta) \quad (3)$$

We estimate $\hat{\beta}$ by minimizing $R_{emp}(\beta)$, therefore:

$$R_{emp}(\hat{\beta}) < R(\hat{\beta})$$

It is interesting to note, that in the worst case, where the testing data is just a number of random (coin flipping) points, the testing error would be:

$$\text{Testing Error} = E_p(1_{Y \neq \text{sgn}(X\beta)}) = \frac{1}{2}$$

An interesting question is the relation between the training error and the testing error. In doing so we will use a similar metric called *score* which is $y \cdot \text{sgn}(X\beta)$.

$$y \cdot \text{sgn}(X\beta) = \begin{cases} +1 & \text{if } y = \text{sgn}(X\beta) \\ -1 & \text{if } y \neq \text{sgn}(X\beta) \end{cases}$$

The training score is:

$$\text{Training Score} = \frac{1}{n} \sum_{i=1}^n y_i \cdot \text{sgn}(X_i\beta)$$

According to the *law of large numbers* for fixed β , and with totally random training data, this value $\rightarrow 0$, as $n \rightarrow \infty$. Therefore for fixed β , and random data, training and testing scores are equal.

But in real situations, β is not fixed. We choose β to maximize the training score. In this case:

$$\begin{aligned} P\left(\max_{\substack{\hat{y}_1 \\ \vdots \\ \hat{y}_p}} \frac{1}{n} \sum y_i \cdot \text{sgn}(x_i\beta) > \epsilon\right) &= P\left(\bigcup_{\substack{\hat{y}_1 \\ \vdots \\ \hat{y}_p}} \frac{1}{n} \sum y_i \cdot \text{sgn}(x_i\beta)\right) \\ &= \sum_{\substack{\hat{y}_1 \\ \vdots \\ \hat{y}_p}} P\left(\frac{1}{n} \sum y_i \cdot \text{sgn}(x_i\beta)\right) \\ &= N \times cte \times e^{-n \cdot \text{rate} \cdot \epsilon^2} \end{aligned} \quad (4)$$

Where N is the number of possible combinations of $(\hat{y}_1, \dots, \hat{y}_p)$. This value has been proved to be:

$$N = \text{number of } \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_p \end{pmatrix} = \left(\frac{en}{VC - dim}\right)^{vc-dim} < 2^n \quad (5)$$

Using this result we can bound the difference between the testing and training scores when the input is purely random.

Consider the assumptions in Table 2. For each observation we flip a coin and we compute the score as follows:

observations	response	features				random
1	y_1	x_{11}	x_{12}	\dots	x_{1p}	z_1
2	y_2	x_{21}	x_{22}	\dots	x_{2p}	z_2
\vdots	\vdots			\vdots		
n	y_n	x_{n1}	x_{n2}	\dots	x_{np}	z_n

Table 2: Assumptions in a perceptron with a random coin flip

$$\begin{aligned}
\text{Score} &= \frac{1}{2} \sum_i \underbrace{z_i y_i}_{\text{coin flipping}} \cdot \text{sgn}(x_i \beta) \\
&= \frac{1}{2} \left[\underbrace{\sum_{i, z_i+} y_i \cdot \text{sgn}(x_i \beta)}_{\text{testing score}} - \underbrace{\sum_{i, z_i-} y_i \cdot \text{sgn}(x_i \beta)}_{\text{training score}} \right] \\
&= \left(\frac{en}{VC - \dim} \right)^{vc - \dim} \times cte \times e^{-nrates \cdot \epsilon^2} \quad (\text{using 4 and 6})
\end{aligned}$$

As we know $\left(\frac{en}{VC - \dim} \right)^{vc - \dim} < 2^n$, therefore the difference between the testing and the training score goes to zero as $n \rightarrow \infty$.

4 Acknowledgment

This note is based on Prof. Ying Nian Wu's lectures on Theoretical Statistics at UCLA.