COS 511: Foundations of Machine Learning

Rob Schapire
Scribe: Mert R Sabuncu

Lecture #16
April 3, 2003

1 Linear Regression (cont'd)

We are given the training set $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m)$, where $\mathbf{x}_i \in \mathbb{R}^n,\ y_i \in \mathbb{R}$. We also assume that each $(\mathbf{x}_i, y_i)$ is independent and identically distributed (i.i.d.) with respect to the distribution D, i.e. $(\mathbf{x}, y) \sim D$, $\forall \mathbf{x} \in \mathbb{R}^n,\ y \in \mathbb{R}$. In the linear model, the goal is to find the vector $\mathbf{v}$ such that $R_{\mathbf{v}} = E_{(x,y)\sim D}[(\mathbf{v} \cdot \mathbf{x} - \mathbf{y^2})]$ is minimized. $R_{\mathbf{v}}$ is called the *risk* and can be considered as the analog of the generalization error in classification. Note that $R_{\mathbf{v}}$ is a random variable that depends on the training set through the algorithm that generates $\mathbf{v}$.

Let's consider the following algorithm:

- Run the Widrow-Hoff (WH) algorithm (with parameter η) on the training data: $S =< (\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m) >$.
- This generates the outputs: $\mathbf{w}_1, ..., \mathbf{w}_m$.
- Output $\mathbf{v} = \frac{\mathbf{1}}{\mathbf{m}} \sum_{\mathbf{t=1}}^{\mathbf{m}} \mathbf{w_t}$.

Now, we will show that the expected risk for this output, $\mathbf{v}$ is small.

Theorem 1 *Let $E_S[R_{\mathbf{v}}]$ be the expectation of the risk over the different possible training sets (samples) and $R_{\mathbf{u}}$ be the risk of an arbitrary vector $\mathbf{u}$, then:*

$$E_S[R_{\mathbf{v}}] \leq \min_{\mathbf{u}} \left[\frac{R_{\mathbf{u}}}{1-\eta} + \frac{\|\mathbf{u}\|^{\mathbf{2}}}{\eta \cdot m} \right].$$

Proof.

- **Claim 1:** $(\mathbf{v}\ \cdot \mathbf{x} - y)^2 \leq \frac{1}{m} \sum_{t=1}^{m} (\mathbf{w}_t \cdot \mathbf{x} - y)^2$
  Pf.

$$\begin{aligned} (\mathbf{v} \cdot \mathbf{x} - y)^2 &= \left(\left(\frac{1}{m} \sum_t \mathbf{w}_t \right) \cdot \mathbf{x} - y \right)^2 \\ &= \left(\frac{1}{m} \left(\sum_t (\mathbf{w}_t \cdot \mathbf{x} - y) \right) \right)^2 \leq \frac{1}{m} \sum_t (\mathbf{w}_t \cdot \mathbf{x} - y)^2 \end{aligned}$$

  The inequality can easily be shown using the convexity of $f(x) = x^2$.

- **Claim 2:** Let $E[\cdot] := E_{S,x,y}[\cdot]$, then:

$$E[(\mathbf{u} \cdot \mathbf{x}_t - y_t)^2] = E[(\mathbf{u} \cdot \mathbf{x} - y)^2] = R_{\mathbf{u}}$$

  Pf. This is a result of the i.i.d. assumption.

- **Claim 3:** $E[(\mathbf{w}_t \cdot \mathbf{x}_t - y)^2] = E[(\mathbf{w}_t \cdot \mathbf{x} - y)^2]$
  Pf. This is a result of the i.i.d assumption and the fact that $\mathbf{w}_t$ is independent of $(\mathbf{x}_t, y_t)$.

Armed with these claims, we can prove the theorem:

$$\begin{aligned}
E_S[R_{\mathbf{v}}] &= E[(\mathbf{v} \cdot \mathbf{x} - y)^2] \\
&\leq^1 E\left[\frac{1}{m}\sum_t (\mathbf{w}_t \cdot \mathbf{x} - y)^2\right] = \frac{1}{m}\sum_t E[(\mathbf{w}_t \cdot \mathbf{x} - y)^2] \\
&=^3 \frac{1}{m}\sum_t E[(\mathbf{w}_t \cdot \mathbf{x}_t - y_t)^2] = \frac{1}{m}E\left[\sum_t (\mathbf{w}_t \cdot \mathbf{x}_t - y_t)^2\right] \\
&\leq^4 \frac{1}{m}E\left[\frac{\sum_t (\mathbf{u} \cdot \mathbf{x}_t - y_t)^2}{1-\eta} + \frac{\|u\|^2}{\eta}\right] \\
&= \frac{1}{m}\left[\frac{\sum_t E[(\mathbf{u} \cdot \mathbf{x}_t - y_t)^2]}{1-\eta} + \frac{\|u\|^2}{\eta}\right] \\
&=^2 \frac{1}{m}\left[\frac{\sum_{t=1}^m R_{\mathbf{u}}}{1-\eta} + \frac{\|u\|^2}{\eta}\right] = \frac{R_{\mathbf{u}}}{1-\eta} + \frac{\|u\|^2}{\eta m}
\end{aligned}$$

The first inequality (1) and the equalities (2) and (3) can easily be justified using the corresponding claims, stated above. The inequality (4) is a result of the theorem proven in the previous lecture. □

2 Artificial Neural Networks

Like support vector machines (SVM) and other learning algorithms, such as boosting methods, *(artificial) neural networks* are powerful algorithms that are commonly used for different learning problems. They offer ideal solutions to a variety of classification problems such as speech, character and signal recognition, as well as functional prediction and system modeling where the physical processes are not understood or are highly complex. This approach is inspired by the way the densely interconnected, parallel structure of the mammalian brain processes information. However, it should be noted that artificial neural networks are a very simplified model of the brain. Artificial neural networks are built out of units called *neurons* or *perceptrons*. Figure 1 is a illustration of a typical neuron.

Figure 1: A typical neuron

The main idea in artificial neural networks is to build a feed-forward network of neurons (perceptrons). Figure 2 shows a generic multi-layer artificial network composed of 9

perceptrons. In the simplest case, each perceptron produces an output according to the following linear threshold function:

$$h(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_i w_i x_i > t \\ 0 & \text{else} \end{cases}$$

where x_i's are the inputs to the preceptron and w_i's are the corresponding connection weights. Note that the final output is a function of all the connection weights in the network and the inputs to the network, i.e. $\hat{y} = f_{\mathbf{w}}(\mathbf{x})$ where $\mathbf{x}$ and $\mathbf{w}$ are the input and weight vectors of the network.

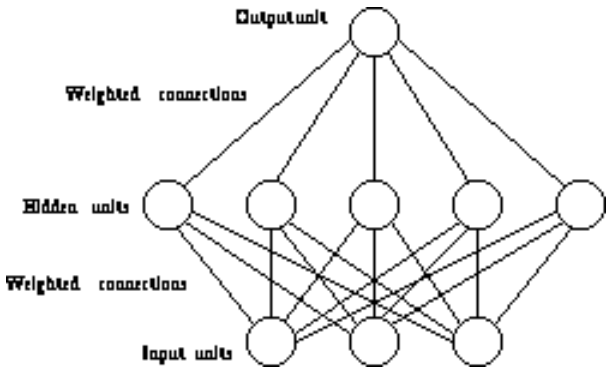

Figure 2: A typical Artificial Neural Network

Note that the goal for an artificial neural network (as any other learning algorithm) is to achieve $y = \hat{y}$ on the test data. Hence, the problem boils down to finding the weight vector $\mathbf{w}$. However, due to the discontinuous nature of the preceptron function, it is hard to use hill-climbing methods to find a solution. Therefore, instead of the hard-thresholding function, other sigmoid-type continuous functions are used in practice. Figure 3 is a plot of an example for a sigmoid-type function, $\sigma(x) = \frac{1}{1+e^{-x}}$. Another common sigmoid is the $\tanh(x)$ function.

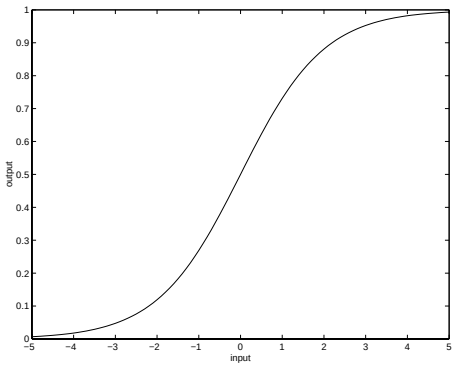

Figure 3: A sigmoid function

Since we have a smooth output function, gradient descent type algorithms can be employed for optimization. An elegant way to do the optimization is to work out the gradient in a backward fashion. This algorithm is called the *Backpropagation Algorithm*. However, it should be noted that gradient descent based methods suffer from local minima problems and "flat" plateaus, where gradients don't contain much information.

3 Modeling Probability Distributions

Modeling and estimating probability distributions is a general problem in computational learning theory. For example, we would like to infer the distribution of eye-colors in a society, using a limited number of samples. Another application is modeling a probability distribution over English word strings to use in speech recognition. Here is a couple of examples for real-life applications of probability distribution modeling:

Example 1 *A speech recognition algorithm tries to decide between the following two sentences: "he sat on a chair" or "he fat on a chair". If we have a good model of the probability distribution, the algorithm could conclude that the first sentence is more likely.*

Example 2 *Let's consider the case where we are trying to estimate the gender of a person based on his/her height. Figure 4 shows a plot for the estimates of the conditional probability densities. Based on these estimates, one can choose the likelier case for each sample, i.e. the gender for which the probability density takes a higher value for a specified height.*

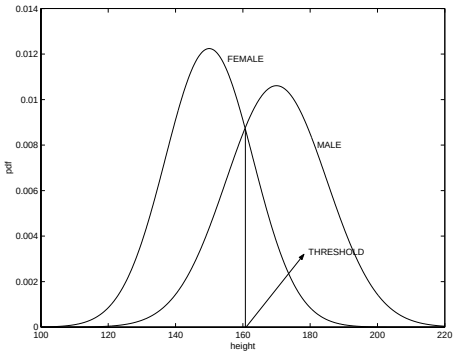

Figure 4: Estimated conditional probability densities

This kind of an approach, where classification is done based on the estimated probability distributions, is called a generative approach. This approach is different from the discriminative approach, where the focus is on modeling the class boundaries or the class membership probabilities directly. For instance, in the previous example the discriminative approach would be to try to estimate the threshold. The intuition is these approaches use the resources more efficiently. However, the generative approach is more natural in some cases. Finally, note that generative approaches deal with probabilities in the form: $Pr(x|y)$, where y is the class and x is the test data. On the other hand in discriminative techniques, one directly estimates $Pr(y|x)$. These two conditional probabilities are related through the Bayes' Rule.

Bayes' Rule:

$$Pr(y|x) = \frac{Pr(x|y)Pr(y)}{Pr(x)}.$$