CMSC35000-1 Introduction to Artificial Intelligence **Winter 2007**

Lecture 4: 1/23/2007

Lecturer: Partha Niyogi *Scribe: Xueyuan Zhou*

Decision Tree and Generalization

4.1 Decision Tree

Notation:

Data point (x_i, y_i), where $0 \leq i \leq n$, $x_i \in R^k$ and $y \in \{+1, -1\}$. Let $X = \{x_1, \cdots, x_n\}$ and $Y = \{+1, -1\}$. $x_i(j)$ is the value of the j^{th} dimension of vector x_i. Hypothesis space $\mathcal{H}$. P is distribution on $X \times Y$.

Bayesian deterministic function is defined as

$$f_p(x) = sign(\mathbb{E}[y|x]) = sign(P(y = +1|x) - P(y = -1|x)) \tag{4.1}$$

If $f_p(x)$ is linear, we can use Perceptron classifier to learn the function. But $f_p(x)$ could also be nonlinear, in which case we have to find some nonlinear classifiers.

Decision tree is such a method that can learn nonlinear function $f_p(x)$. In decision tree T, each node is one question, which is a mapping $q : X(j) \to Y$. And each leaf represent label $y_i \in Y$. The hypothesis space here is $\mathcal{H} = \{q\}$. Therefore, decision tree is a mapping $T : X \to Y$.

How can we build a decision tree based on a given training dataset? One choice for tree building is sets of hyperplanes

$$\mathcal{H} = \{w \cdot x\} \tag{4.2}$$

Another choice is

$$\mathcal{H} = \{(j, t) | j \in 1 \cdots k, t \in \mathbb{R}\} \tag{4.3}$$

where t is a threshold value for the j^{th} dimension of x. For any $x_i \in X$,

$$y_i(j) = \begin{cases} +1 & \text{if } x_i(j) > t \\ -1 & \text{if } x_i(j) \leq t \end{cases} \tag{4.4}$$

How to choose node ? We can use impurity function to select mapping for each node. Let's introduce impurity function first.

Let D denote pair (x_i, y_i), where $i = 1 \cdots n$. Then we have

$$P(D) = \frac{\# y_i | y_i = +1}{n} \tag{4.5}$$

When $P(D) = 0$ or $P(D) = 1$, we call D homogeneous set.

One choice for impurity function I(P(D)) is entropy.

$$I(P) = Entropy(P) = P \log(\frac{1}{P}) + (1 - P) \log(\frac{1}{1 - P}) \tag{4.6}$$

Another choice might be $I(P) = P(1 - P)$.

Assume for some node, our question is q, by which we divide D into D_{+1} and D_{-1}, where $D_{+1} = \{(x_i, y_i) \in D | q(x_i) = +1\}$, and $D_{-1} = \{(x_i, y_i) \in D | q(x_i) = -1\}$. Then the impurity function of D and q is

$$I(q, D) = \frac{| D_{+1} | I(P(D_{+1}))}{| D |} + \frac{| D_{-1} | I(P(D_{-1}))}{| D |} \tag{4.7}$$

By minimizing $I(q, D)$ we can find q for current node.

$$q = \arg\min_{q \in \mathcal{H}} \{I(q, D)\} \tag{4.8}$$

The error rate for decision tree will be

$$e(D) = \frac{\#errors}{| D |} = \frac{| D_{+1} | (1 - P(D_{+1}))}{| D |} + \frac{| D_{-1} | (1 - P(D_{-1}))}{| D |} \tag{4.9}$$

When each training data point corresponds to one leaf, error rate of decision trees is zero. If n, the size of training data, is quite large, size of decision will be large too. Note: min error at each node does not guarantee to get min error for the whole decision tree. This is similar to the local minima and global minima problem.

4.2 Generalization

Many techniques for classification can achieve zero error rate.

1. *Linear Classifier*: if data is linear separable, we can achieve 0 error rate. If data is nonlinear distributed, then we can also get 0 error rate by increasing data dimension.
2. *Multiplier Perceptron Classifier*: by increasing hidden units and hidden layers, we can get 0 error rate.
3. *Nearest Neighbor Classifier*: for training data points, in fact there is no error at all.

One common feature of them is that they reduce errors by increasing the complexity of classifiers. For linear classifier, increasing dimension means we have more parameters for the extra dimensions. More hidden units and hidden layers in MPL introduce more parameters. Nearest Neighbor Classifier partition vector space into n cells. With more labeled data, the number of cells will increase.

Definition 4.1 *Empirical error is defined as*

$$I_{emp}(h) = \frac{1}{n} \sum_{i=1}^{n} | y_i - h(x_i) | \, , \, h \in \mathcal{H} \tag{4.10}$$

This error is based on training data only. By minimizing $I_{emp}(h)$, we can have

$$\hat{h}_n = \arg\min_{h \in \mathcal{H}} \{I_{emp}(h)\} \tag{4.11}$$

Definition 4.2 *Best error is defined by expectation*

$$I(h) = \mathbb{E}[\mid y - h(x) \mid] \tag{4.12}$$

By minimizing $I(h)$ we have

$$h^* = \arg\min_{h \in \mathcal{H}} \{I(h)\} \tag{4.13}$$

We are interested in that when $n \to \infty$, what is the difference between $\hat{h}_n$ and h^*. As data is provided more and more, we hope $\hat{h}_n$ gets closer and closer to h^*. If $\hat{h}_n \to h^*$ when $n \to \infty$, then

$$\mid I(\hat{h}_h) - I_{emp}(\hat{h}_n) \mid \to 0 \tag{4.14}$$

So in this case, empirical error will converge to best error. This means $\hat{h}_n$ is consistent with h^*.

For these 0 error classifier, e.g. Nearest Neighbor, we can achieve $I_{emp}(\hat{h}) = 0$. But as we know, $I(\hat{h}) \geq$ Bayesian error rate, so

$$\mid I(\hat{h}_n) - I_{emp}(\hat{h}_n) \mid \neq 0 \tag{4.15}$$

This shows that 0 error rate classifiers can not be generalized. These classifiers will not perform better in the future.

Assume $\exists e_p, g_p$, $e_p = \mathbb{E}[\mid y - g_p(x) \mid] = I(g_p)$, s.t. $(\forall f : x \to y)$ $(I(g_p) \leq I(f))$. So $g_p(x)$ is the optimal classifier based on the true distribution. Now we have three kinds of classifier. One is $\hat{h}_n$, which is obtained from equation (4.11). One is h^*, obtained from equation (4.13). And the optimal classifier g_p. From the definition, it is clear that

$$(\forall h \in \mathcal{H})(I_{emp}(\hat{h}_n) \leq I_{emp}(h^*)) \tag{4.16}$$

$$(\forall h \in \mathcal{H})(I(h^*) \leq I(\hat{h}_n)) \tag{4.17}$$

And we also know

$$(\forall h \in \mathcal{H})(I(g_p) \leq I(h)) \tag{4.18}$$

Recall that $I(h) = \mathbb{E}[\mid y - h(x) \mid]$ and $I_{emp}(h) = \frac{1}{n}\sum_{i=1}^{n} \mid y_i - h(x_i) \mid$. Let $z = \mid y - h(x) \mid$. Then from law of large number

$$(\forall \epsilon > 0)(P(\mid \frac{1}{n}\sum_{i=1}^{n} z_i - \mathbb{E}[z] \mid > \epsilon) \leq 2e^{-\epsilon^2 n}) \tag{4.19}$$

This is equivalent that with probability $\geq 1 - 2e^{-\epsilon^2 n}$, $I_{emp}(h) \longrightarrow I(h)$ as $n \to \infty$. (more generalization in next lecture)

Note: equation (4.15) gives a nice explanation about 0 error classifiers can not be generalized. Assuming in some ideal case, Bayesian error is 0, then $I_{emp}(\hat{h}_n) = 0$ might be optimal, which means it may perform the same best in the future. But in fact Bayesian error is not 0(or we are only interested in that case), which means error can not be avoided in the future (this is true even for human beings), then current classifier is not consistent in the future.