CMSC35000-1 Introduction to Artificial Intelligence	**Winter 2005**
Lecture 6: 1/20/2005	
Lecturer: Partha Niyogi	*Scribe: George Kuan*

Nearest Neighbors Learning Algorithm and Generalization

6.1 Nearest Neighbors Learning Algorithm

We looked at perceptrons, MLPs, Kernel Based Methods, and decision trees. In decision trees, we have questions and trees that we have to build. If we work with threshold functions, we can get zero error. Similarly, we can get zero training error using a linear combo of bases using kernel based methods. Using MLPs we get local minimum but not global minimum. For perceptrons, we saw a proof of convergence.

The simplest classifier is **nearest neighbors**. We are given a set of (X_i, Y_i). Let f be the classifier that when given X, we look at the nearest X_i to this X and use that X_i's associated Y_i. That is to say, $f(X) = Y_j$ where $j = \arg\min \|X_i - X\|$. This classifier function f has zero training error.

There are a number of variations we can make on the nearest neighbors algorithm. We may consider many different distance functions on different metric spaces. This is a flexibility. Another point of flexibility, is that we can find the k nearest neighbors of X instead of just a single nearest neighbor X_i and do a majority vote to get a Y_j. This is a variation on nearest neighbors.

Most of these pattern classification techniques we have covered are actually frameworks on which we can make quite a few variations by refining some parameters or making slight adjustments to the techniques. In decision trees, we have to choose impurity function, class of questions, and pruning criteria. In Kernel based methods, we had to choose the kernel. For MLPs, we had to choose the architecture of network and the number of nodes in each layers. For nearest neighbors, most crucially, we need to know what is the distance metric.

A disadvantage of nearest neighbors is that we need to store the k nearest neighbors. But remember, in the kernel based classifier we have $f(x) = \text{sign}(\sum \alpha e^{-||x-x_i||^2})$. By keeping/summing over all the α's and $e^{-||x-x_i||^2}$, we are essentially keeping all the training data around anyway. This sum (for kernel-based method) is like a weighted distance. For a particular x, if x is close to a particular x_i and far from all the x_j's, then the effect of x_i dominates. In a sense, this is almost a nearest neighbor classifier. So, kernel based machine is doing a kind of smoothed nearest neighbor. With the α and for x not near a particular x_i, we have something similar to k nearest neighbors. So the kernel based method is a kind of a soft nearest neighbor algorithm. Nearest neighbor essentially memorizes the training data. Other classifier methods are genetic algorithms and generative models (Bayes Nets).

6.2 Generalization

Now we will turn our attention to the problem of **generalization**. This subject is called machine learning, pattern classification. The goal is to take data D and give a function f_D (an empiricially chosen function/classifier). When we discussed Bayes optimal classifier, we saw that there was a distribution P on

$X \times Y$ from which we can draw (X_i, Y_i) pairs, we have an **error function**

$$e(f, z) = \begin{cases} 1 & \text{if } f(x) \neq y \\ 0 & \text{otherwise} \end{cases}$$

. Furthermore, the **average or expected error rate** is $\mathbb{E}[e(f,z)] = \int e(f,z)P(z)dz = \sum_y \in |y - f(x)|P(x,y)dx$ where $\mathbb{E}(e(f,z))$ is the expected value function on the error function $e(f,z)$. What minimizes this expected value, $\min_f \mathbb{E}[e(f,z)]$ is the Bayes discriminant (function).

We can say that $\text{Err}(f) = \mathbb{E}_Z[e(f,z)]$ (error rate of a classifier). We can further define an **empirical error**:

$$\hat{\text{Err}}(f) = \frac{1}{n}\sum_{i=1}^{n} e(f, z_i)$$

.

Given a collection of functions $\mathcal{H}$, the generic "induction principle" is to pick a class of functions $\mathcal{H}$ and do **empirical risk minimization** (choose the one function from $\mathcal{H}$ that minimizes the risk of error). If there are multiple functions that get error 0, we haven't described how to choose between the multiple functions. We'll ignore that for now. Let's define the empirical error:

$$f_D = \text{avg } \min_{f \in \mathcal{H}} \hat{\text{Err}}(f)$$

$$f_{\mathcal{H}} = \text{avg } \min_{f \in \mathcal{H}} \text{Err}(f)$$

. Note that the minimum exists but maybe not the minimizer, but that is a technical fact which we will ignore for now.

What was problematic is that we saw the following inequalities: $\hat{\text{Err}}(f_{\mathcal{H}}) \geq \hat{\text{Err}}(f_D)$ by definition but $\text{Err}(f_D) \geq \text{Err}(f_{\mathcal{H}})$. This shows how making training error zero doesn't guarantee a good function. We would like to ask the question, how can we guarantee that the empiricial risk minimization is a good principle: when is it a good and when is it bad? As we see from the inequalities, it may or may not be good. If we are lucky $f_{\mathcal{H}}$ and f_D might be close, but they might be very far from each other.

How do we think about this fact? First, let us see what is the relation between $\hat{\text{Err}}$ and Err. If they are the same, then minimizing one is minimizing the other. First we fix f and see that:

$$\hat{\text{Err}}(f) = \frac{1}{n}\sum_{i=1}^{n} e(f, z_i)$$

where $z_i \sim P$ where z_i are independent draws from P.

Define $e(f,z)$ to be a random variable 0-1 valued (randomness depends on z). probability $\mathbb{P}[e(f,z) = 1] = \text{Err}(f) = \int e(f,z)dP(z)$.

Also, we know that the **empirical average of this random variable** $e(f,z)$ is:

$$\hat{\text{Err}}(f) = \frac{1}{n}\sum_{i=1}^{n} e(f, z_i)$$

.

In probability, you saw the law of large numbers. For example $\frac{\#\text{ heads}}{n} \overset{n\to\infty}{\to} p$. Notice that $\hat{\text{Err}}$ follows this. Let us state the weak law of large numbers formally: Given random variables $X_1, X_2, \ldots$ iid random variables 0-1 valued drawn from probability distribution P $(\sim P)$

$$\frac{1}{n}\sum_{i=1}^{n} X_i \to \text{expected value } \mathbb{E}[X] \text{ because } \mathbb{E}[X_i] = p\ \forall i$$

$$\mathbb{P}\left[\left\|\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbb{E}[X]\right\| > \epsilon\right] \leq 2e^{-\epsilon^2 n}$$

. [Chernoff Bound, Hoeffding Bound] Applying the Chernoff Bound here on $\hat{\mathrm{Err}}$, we can say (**statement 1**) for fixed f $\mathbb{P}[|\hat{\mathrm{Err}}(f) - \mathrm{Err}(f)|] \leq 2e^{-\epsilon^2 n}$. This means that with large n, the empirical and true error are within an ϵ of each other. To make this statement, we fixed f and randomized z's.

Because we want to minimize over a whole class of functions, we need a **uniform law of large numbers**. There are consequences here. (**Statement 2**) for fixed f $\mathbb{P}[f \in \mathcal{H}|\hat{\mathrm{Err}}(f) - \mathrm{Err}(f)| > \epsilon] \leq \delta$. This statement is the same as saying with probability $> 1 - \delta, \forall f \in \mathcal{H}|\hat{\mathrm{Err}}(f) - \mathrm{Err}(f)| \leq \epsilon$. This is just saying that the complement of the event in statement 2 must happen with probability $> 1 - \delta$. Recall from probability that $\mathbb{P}[A] \leq \delta \Rightarrow \mathbb{P}[\bar{A}] > 1 - \delta$.

So with probability $> 1 - \delta \forall f \in \mathcal{H}|\hat{\mathrm{Err}}(f) - \mathrm{Err}(f)| \leq \epsilon(*)$. Thus you would think that minimizing $\hat{\mathrm{Err}}(f)$ over $\mathcal{H}$ is similar to minimizing $\mathrm{Err}(f)$ over $\mathcal{H}$.

We would actually like to say that $\mathrm{Err}(f_{\mathcal{H}}) \leq \mathrm{Err}(f_D) \leq \mathrm{Err}(f_{\mathcal{H}}) + 2\epsilon$ [because f_D minimizes error but not $f_{\mathcal{H}}$] This is easy to show. $\mathrm{Err}(f_D) \leq \hat{\mathrm{Err}}(f_D) + \epsilon \leq \hat{\mathrm{Err}}(f_{\mathcal{H}}) + \epsilon$ (by the definition of empirical risk minimization) and all the preceding is $\leq \mathrm{Err}(f_{\mathcal{H}}) + \epsilon + \epsilon = \mathrm{Err}(f_{\mathcal{H}} + 2\epsilon$. The essential idea: if statement 2 is true, then (*) is true and it implies this result. This tells us that empirical risk minimization is good. $\mathrm{Err}(f_{\mathcal{H}})$ is the best error for function in $\mathcal{H}$. This makes learning possible. In order to compute $f_{\mathcal{H}}$, we need to know nature. To compute f_D, we don't need to know nature. Now we are saying that what we learn from experience in f_D is no worse than $2\epsilon+$ what we would get if we had known nature. This requires statement 2. Statement 1 is insufficient.

Now let's ask, under what conditions would statement 2 be true? We will consider $\mathcal{H}$ to be a finite set of functions $\mathcal{H} = f_1, f_2, \ldots, f_N$. With this, we can prove a version of statement 2. We'll define an event E_1 s.t. E_1 is the event where $|\hat{\mathrm{Err}}(f_1) - \mathrm{Err}(f_1)| > \epsilon$. This is an event, we can draw z_1 to z_n and check if the event is true. Let E_i be the event such that $|\hat{\mathrm{Err}}(f_i) - \mathrm{Err}(f_i)| > \epsilon$.

We get the following:

$$\mathbb{P}[\cup_{i=1}^{N} E_i] \leq \sum_{i=1}^{N} \mathbb{P}[E_i] \leq \sum_{i=1}^{N} 2e^{-\epsilon^2 n}$$

using (Union Bound i.e. $\mathbb{P}(A \cup B) \leq \mathbb{P}(a) + \mathbb{P}(B)$ and $E = \cup_{i=1}^{N} E_i$).

Let $\mathcal{H} =$ finite set. $\mathbb{P}[\exists f \in \mathcal{H} s.t. |\hat{\mathrm{Err}}(f) - \mathrm{Err}(f)| > \epsilon] \leq 2Ne^{-\epsilon^2 n} < \delta$ (because N is fixed and we can vary n to make $2Ne^{-\epsilon^2 n}$ as small as we want). $2Ne^{-\epsilon^2 n} = \delta \Rightarrow 2N/\delta = e^{\epsilon^2 n} \Rightarrow n = \frac{1}{\epsilon^2} \log \frac{2N}{\delta}$.

We know that with probability $> 1 - \delta \mathrm{Err}(f_D) < \mathrm{Err}(f_{\mathcal{H}}) + 2\epsilon$.

Thus we make the following statement: fix $\epsilon, \delta > 0$, if $n > \frac{1}{\epsilon^2} \log \frac{2N}{\delta}$ then with probability $> 1 - \delta$, $\mathrm{Err}(f_D) \leq \mathrm{Err}(f_{\mathcal{H}} + 2\epsilon$. How is this statement derived? Recall that event $\cup_{i=1}^{N} E_i$ is true exactly when one E_i is true which means $|\hat{\mathrm{Err}}(f_i) - \mathrm{Err}(f_i)| > \epsilon$. So we get $\mathbb{P}[\exists f \in \mathcal{H}$ s.t. $|\hat{\mathrm{Err}}(f) - \mathrm{Err}(f)| > \epsilon$. We can show this using the finiteness of $\mathcal{H}$.

In words, our result says that for fixed ϵ and δ, if we draw enough samples (for sufficiently large n) we have high probability that our error is good and by extension f_D is good.

Fix $\delta > 0$ and fix n samples. We are constrained in the number of samples we can draw. From $2Ne^{-\epsilon^2 n} = \delta$, we get that $\epsilon = \sqrt{\frac{1}{n} \log \frac{2N}{\delta}}$. Thus with probability $> 1 - \delta$, $\mathrm{Err}(f_D) \leq \mathrm{Err}(f_{\mathcal{H}} + 2\sqrt{\frac{1}{n} \log(\frac{2N}{\delta})}$. This is another form of the result. Observe that if we want to do well, we want N to be small because otherwise our error will blow up. Earlier (in previous lectures) we mentioned that we want $\mathcal{H}$ to be small (and therefore N), now we see precisely why.

So why don't we want the cardinality of $\mathcal{H}$, i.e. N, to be 1? If $N = 1$, then $\mathcal{H}$ is unlikely to represent the Bayes discriminant function.

This is the tradeoff. We want $\mathcal{H}$ to be small to reduce error, but we want it to be expressive especially of the Bayes discriminant function.

In retrospect, it seems that nearest neighbors may be a bad idea. In nearest neighbors, $\mathcal{H}$ is set of all $f : X \to 0, 1$. For kernel based methods, we have $\sum_{i=1}^{n} \alpha_i K_{*i}$ which makes $\mathcal{H}$ an infinite set, but our results for error only applies to a finite set, hence it is not strong enough to make any claims about many of our methods.

It turns out that we can replace $\mathcal{H}$ with Vapnik and Chervonenkis's VC($\mathcal{H}$) (VC dimension of infinite $\mathcal{H}$) and then replace N with $VC(\mathcal{H})$ in our results.

This should give us an intuition why $\mathcal{H}$ should be small. We have already seen that driving training error to zero is trivial. This will provide us with a good theoretical infrastructure to examine future questions. We will see that in learning languages, the issue of how to learn from a finite set of experiences will come up again (we have to restrict ourselves to a set of languages).