CS 683: Advanced Algorithms	Classification via Nearest Neighbors
April 24, 2001	Scribes: David Kempe and Chaitanya Swamy

1 Introduction

In the last lecture, we saw how to efficiently calculate approximate nearest neighbors in high dimensions. In this lecture, we show a natural application of nearest neighbor computations to the classification of samples.

2 The Classification Problem (Cover-Hart, 1967)

We consider the following classification problem: we are given a training set consisting of a sequence of examples. Each example is a labeled point (x, θ) where $x \in \mathbb{R}^D$ and the label θ comes from one of m classes. The goal is to "learn" the rule underlying the labeling, i.e. to be able to predict the labels θ of new examples x.

Without further restrictions on the actual "rule" underlying the labeling, we cannot expect to be able to solve the problem exactly or even approximately. In the absence of constraints on the labels, an adversary with knowledge of the algorithm may always choose labels so as to fool the algorithm on every sample. However, most practical applications exhibit much more structure in their labeling, and it is therefore natural to place restrictions on the labeling.

We consider a *probabilistic generative* model of data — more specifically, a *Mixture Model* consisting of a superposition of distributions for each class :

1. Each class $i \in \{1, \ldots, m\}$ is generated from a density function $f_i : \mathbb{R}^D \to \mathbb{R}$ which is continuous and positive at every point. Therefore, the probability that a sample x of type i lies in U is $\Pr[x \in U] = \int_U f_i(y)dy$, for any measurable set U.

2. We also have a *mixture vector* $q = (q_1, \ldots, q_m)$ of probabilities that a sample belongs to a class i. That is, $q_i \geq 0$ for all i, and $\sum_i q_i = 1$.

3. To produce a labeled example, we first pick a class θ according to the distribution q, and then pick x according to the density function f_θ, and report (x, θ).

The training set consists of n training samples $(x_1, \theta_1), \ldots, (x_n, \theta_n)$, which are generated as described above. In addition, a test sample (x, θ) is generated, but the algorithm is only given x. The goal is to guess θ.

The "Nearest Neighbor" heuristic guesses θ by finding the nearest neighbor from among all training samples, and guesses that x has the same label θ as this nearest neighbor. More formally, let $x'_n \in \{x_1, \ldots, x_n\}$ be the point nearest to x. Then, the algorithm outputs $\theta = \theta'_n$, where θ'_n is the class of x'_n.

We want to evaluate the performance of "Nearest Neighbor" in terms of the number of wrong guesses. In order to do so, we need a notion of an optimal algorithm. Suppose the algorithm knew the distribution q and the density functions $f_1, \ldots, f_m$. Then, the best way to classify x would be to classify it with the maximum likelihood label, i.e. the label k such that $\Pr[\theta = k \,|\, x]$ is maximized. This is called the Maximum Likelihood rule or Bayes' heuristic. Let $\hat{q}_k = \Pr[\theta = k \,|\, x]$ be the likelihood that a given sample x has label $\theta = k$. By Bayes' Theorem, $\hat{q}_k = \frac{q_k f_k(x)}{\sum_i q_i f_i(x)}$.

Note that the choice of x gives a mapping from the a priori vector q to the a posteriori vector $\hat{q}$. Since we are in a probabilistic domain, it is natural to expect that any algorithm will end up misclassifying some examples. For a classification algorithm $\mathcal{A}$ that only looks at the test sample x, but not at the training data, we define the "risk" of $\mathcal{A}$ on a given sample x to be $r_{\mathcal{A}}(x) = \Pr[\mathcal{A}$ misclassifies $x]$. The performance of $\mathcal{A}$ is then the expected risk of $\mathcal{A}$, over all x, i.e. $R_{\mathcal{A}} = \mathrm{E}_x[r_A(x)]$, where x is chosen according to the *compound density* $f(x) = \sum_i q_i f_i(x)$. Since the f_i are all continuous and positive, so is f.

On any point x, the risk of the Bayes rule is $r_B(x) = 1 - \max_k \hat{q}_k(x)$ and $R_B = \mathrm{E}_x[r_B(x)]$. Let $R(n)$ denote the expected risk of "Nearest Neighbor" after n random samples. That is, $R(n)$ is the expectation of $\Pr[\theta'_n \neq \theta]$, over all the random $\{(x_j, \theta_j)\}, (x, \theta)$. As $n \to \infty$, this risk $R(n)$ converges to some value R with probability 1. We call R the *large-sample "Nearest Neighbor"* risk.

3 The Main Theorem

In the following theorem, we relate the performance of the simple "Nearest Neighbor" heuristic which does not have knowledge of the underlying distribution of points, to the performance of Bayes' heuristic, which has knowledge of the density functions. The performance of the "Nearest Neighbor" heuristic will be considered in the limit as the training set size tends to infinity.

Theorem 3.1 *For continuous and positive density functions $f_1, \ldots, f_m : \mathbb{R}^D \to \mathbb{R}$, and any mixture vector $(q_1, \ldots, q_m)$, the "Nearest Neighbor" heuristic guarantees a risk $R \leq 2R_B$ in the limit of large samples.*

The intuition behind the proof (and the factor 2 approximation) is that classification errors by "Nearest Neighbor" can result from one of three cases:

1. "Nearest Neighbor" classified the test sample with the true maximum likelihood label, but it actually belonged to a different class, i.e. its generation was an event of smaller probability. The Bayes' heuristic will always make the same misclassification, since it classifies according to maximum likelihood.

2. "Nearest Neighbor" classified the test sample with a label that was not the maximum likelihood label. This must have resulted from "wrong" training data, i.e. events of smaller probability must have occurred in the training set. We want to bound the

probability of this in terms of the probability of the first case, because the second kind of error is not incurred by the Bayes' heuristic (which knows the distribution).

3. The test sample was from a class that does not coincide with the maximum likelihood estimate, and "Nearest Neighbor" misclassifies it with another label that is also different from the maximum likelihood estimate. Since the Bayes' heuristic will also misclassify this sample, this case does not cause additional difficulties in the analysis. Also, its probability is smaller than that of the previous two cases.

In order to bound the probability of the second case, we first want to prove a lemma showing that any test sample is, with high probability, eventually approximated arbitrarily closely by training samples.

Lemma 3.2 *For all points x, the sequence $(x'_n)_{n=1}^{\infty}$ converges to x almost surely.*

Proof. Fix any point x, and consider any $\varepsilon > 0$. Let $B_\varepsilon(x)$ be the ball of radius ε around x, and $\alpha = \int_{B_\varepsilon(x)} f(y)dy$. Since f was assumed to be positive, we know that $\alpha > 0$. The probability that none of $x_1, \ldots, x_n$ lie in $B_\varepsilon(x)$ is $(1-\alpha)^n$ which converges to 0 as $n \to \infty$. ∎

Proof of Theorem 3.1. Let us first fix the sample point x, and its nearest neighbor x'_n from among the training samples. Conditioned upon x and x'_n, the probability that x is misclassified by the "Nearest Neighbor" heuristic is

$$\begin{aligned}
r(x, x'_n) &= \Pr[\theta \neq \theta'_n \mid x, x'_n] \\
&= \sum_{j \neq k} \Pr[\theta = j \text{ and } \theta'_n = k \mid x, x'_n] \\
&= \sum_{j \neq k} \Pr[\theta = j \mid x, x'_n] \cdot \Pr[\theta'_n = k \mid x, x'_n] \\
&= \sum_{j \neq k} \Pr[\theta = j \mid x] \cdot \Pr[\theta'_n = k \mid x'_n] \\
&= \sum_{j \neq k} \hat{q}_j(x) \cdot \hat{q}_k(x'_n).
\end{aligned}$$

In the third step, we used that the samples are chosen independently from the underlying distribution, and in the fourth step that the probability of one particular sample belonging to a given class does not depend on other samples.

Knowing the probability of misclassifying any one sample, we can now look at the risk $R(n)$ of misclassifying a test sample after n training samples, and consider its limit R as the number n of samples grows large. Since the risk $R(n)$ is the expectation of $r(x, x'_n)$ when taken over all pairs (x, x'_n), we obtain that almost surely,

$$\begin{aligned} R &= \lim_{n\to\infty} R(n) \\ &= \lim_{n\to\infty} \mathrm{E}_{(x,x'_n)}[r(x, x'_n)] \\ &= \lim_{n\to\infty} \mathrm{E}_{(x,x'_n)}[\sum_{j\neq k} \hat{q}_j(x) \cdot \hat{q}_k(x'_n)] \\ &= \mathrm{E}_x[\sum_{j\neq k} \hat{q}_j(x) \cdot \hat{q}_k(x)]. \end{aligned}$$

In the last step, we interchanged the limit and the expectation (which is valid since $\sum_{j\neq k} \hat{q}_j(x) \cdot \hat{q}_k(x'_n)$ is bounded) and used that $\lim_{n\to\infty} x'_n = x$ almost surely, which we proved in Lemma 3.2.

In order to bound the error probability of the "Nearest Neighbor" heuristic, we want to relate $R = \mathrm{E}_x[\sum_{j\neq k} \hat{q}_j(x)\hat{q}_k(x)]$ to $R_B = \mathrm{E}_x[1 - \max_k \hat{q}_k(x)]$. We will actually be able to prove a bound term by term, i.e. relating $r(x) := \sum_{j\neq k} \hat{q}_j(x)\hat{q}_k(x)$ to $r_B(x)$ for each x. So fix some x, and let $\hat{k}$ be an index attaining $\max_k \hat{q}_k(x)$.

By rearranging terms, and observing that $\sum_{j=1}^{m} \hat{q}_j(x) = 1$, we can rewrite

$$r(x) = \left(\sum_{j=1}^{m} \hat{q}_j(x)\right)^2 - \sum_{j=1}^{m} \hat{q}_j(x)^2 = 1 - \sum_{j=1}^{m} \hat{q}_j(x)^2$$

Applying the Cauchy-Schwartz inequality to $\sum_{j\neq k} \hat{q}_j(x)$, we obtain that

$$(m-1) \sum_{j\neq k} \hat{q}_j(x)^2 \geq \left(\sum_{j\neq k} \hat{q}_j(x)\right)^2 = r_B(x)^2.$$

Now, we add $(m-1)\hat{q}_{\hat{k}}(x)^2$ to both sides, and use that $\hat{q}_{\hat{k}}(x)^2 = (1 - r_B(x))^2$, obtaining

$$\begin{aligned} (m-1) \sum_{j} \hat{q}_j(x)^2 &\geq r_B(x)^2 + (m-1)\left(1 - r_B(x)\right)^2 \\ &= (m-1) - 2(m-1)r_B(x) + m r_B(x)^2. \end{aligned}$$

Dividing both sides by $(m-1)$ and rearranging terms now yields

$$2r_B(x) - \frac{m}{m-1} r_B(x)^2 \geq 1 - \sum_{j} \hat{q}_j(x)^2 = r(x).$$

Therefore, $r(x) \leq 2r_B(x)$ for all x, completing the proof. ∎

The approximation guarantee of $2 - o(1)$ proved in Theorem 3.1 is actually tight, as can be seen by the following example:

Example 3.1 Let $f_1 = f_2 = \ldots = f_m$ be an arbitrary positive density function, and define $q_1 = 1 - R_B, q_2 = q_3 = \ldots = q_m = \frac{R_B}{m-1}$. Then, $r_B(x) = R_B$ for any x, and

$$\begin{aligned}
r(x) &= \sum_{j \neq k} \hat{q}_j(x)\hat{q}_k(x) \\
&= 2(m-1)(1-R_B)\frac{R_B}{m-1} + (m-1)(m-2)\frac{R_B^2}{(m-1)^2} \\
&= 2R_B - 2R_B^2 + \frac{m-2}{m-1}R_B^2 \\
&= 2R_B - o(R_B).
\end{aligned}$$