EEE 598C: Statistical Pattern Recognition
Lecture Note 6a: More Non-parametric Estimation

Darryl Morrell

October 11, 1996

1 K_n Nearest Neighbor Estimation

Our estimator of the density at point $\mathbf{x}$ using a training set of size n is

$$\hat{f}_n(\mathbf{x}) = \frac{k_n}{nV_n}$$

where k_n is the number of training vectors in a region $\mathcal{R}_n$ that contains $\mathbf{x}$ and has volume V_n. The Parzen window approach was to make V_n a given function of n, and then count the number of training vectors in $\mathcal{R}_n$. Another approach, called the k_n nearerst neighbor estimate, is to make k_n a given function of n, and let $\mathcal{R}_n$ (and consequently V_n) grow until k_n training vectors are contained in $\mathcal{R}_n$.

2 Series Expansions

Another method of estimating a density from training data is to find a series expansion of the Parzen window $\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)$:

$$\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right) = \sum_{j=1}^{m} a_j \psi_j(\mathbf{x}) \beta_j(\mathbf{x}_i)$$

a_j are the series expansion coefficients. $\psi_j(\mathbf{x})$ and $\beta_j(\mathbf{x}_i)$ can be obtained, for example, using Taylor series expansions or other polynomial approximations. With this expansion for the Parzen window, the estimator of the density becomes

$$\begin{aligned} \hat{f}_n(\mathbf{x}) &= \frac{1}{nV_n} \sum_{i=1}^{n} \varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right) \\ &= \frac{1}{nV_n} \sum_{i=1}^{n} \sum_{j=1}^{m} a_j \psi_j(\mathbf{x}) \beta_j(\mathbf{x}_i) \end{aligned}$$

$$\begin{aligned} &= \sum_{j=1}^{m} \psi_j(\mathbf{x}) \left[\frac{a_j}{nV_n} \sum_{i=1}^{n} \beta_j(\mathbf{x}_i) \right] \\ &= \sum_{j=1}^{m} b_j \psi_j(\mathbf{x}) \end{aligned}$$

where

$$b_j = \frac{a_j}{nV_n} \sum_{i=1}^{n} \beta_j(\mathbf{x}_i)$$

So the information in the training set is summarized in the coefficients b_j.

3 Estimating Posterior Probabilities

Given a set of n training data $\mathcal{X}_n$, how might we find an estimate of $P(\omega_i|\mathbf{x})$? One approach would be to first estimate the joint distribution

$$f(\mathbf{x}, \omega_i) = f(\mathbf{x}|\omega_i)P(\omega_i)$$

and then find the conditional distribution

$$P(\omega_i|\mathbf{x}) = \frac{f(\mathbf{x}, \omega_i)}{\sum_{j=1}^{c} f(\mathbf{x}, \omega_j)}$$

We could estimate this joint distribution by counting the number of training vectors that fall in a region $\mathcal{R}_n$ that includes $\mathbf{x}$ and has volume V_n as follows:

$$P(\mathbf{X} \in \mathcal{R}_n, \omega_i) \approx V_n f(\mathbf{x}, \omega_i)$$

$$f(\mathbf{x}, \omega_i) \approx \frac{P(\mathbf{X} \in \mathcal{R}_n, \omega_i)}{V_n}$$

Let k_i be the number of training vectors of class i that fall in $\mathcal{R}_n$; an estimate of $P(\mathbf{X} \in \mathcal{R}_n, \omega_i)$ is

$$\hat{P}(\mathbf{X} \in \mathcal{R}_n, \omega_i) = \frac{k_i}{n}$$

Thus, an estimate of $f(\mathbf{x}, \omega_i)$ is

$$\hat{f}_n(\mathbf{x}, \omega_i) = \frac{k_i}{nV_n}$$

and an estimate of $P(\omega_i|\mathbf{x})$ is

$$\begin{aligned} \hat{P}_n(\omega_i|\mathbf{x}) &= \frac{\hat{f}_n(\mathbf{x}, \omega_i)}{\sum_{j=1}^{c} \hat{f}_n(\mathbf{x}, \omega_j)} \\ &= \frac{\frac{k_i}{nV_n}}{\sum_{j=1}^{c} \frac{k_j}{nV_n}} \\ &= \frac{k_i}{\sum_{j=1}^{c} k_j} \end{aligned}$$

Note that $\sum_{j=1}^{c} k_j$ is the number of training vectors in $\mathcal{R}$. Thus, the estimate of $P(\omega_i|\mathbf{x})$ is the relative frequency of training vectors from state of nature ω_i in the region containing $\mathbf{x}$.

We can take either a Parzen window like approach, in which the volume V_n is a fixed function of n and we count the number of training vectors in this volume, or a k_n nearest neighbor approach in which the number of training vectors is fixed as a function of n, and the volume is increased until this number of training vectors is included.

If the estimate $\hat{P}_n(\omega_i|\mathbf{x})$ is used in a decision rule, the decision rule is to choose ω_i if

$$\hat{P}_n(\omega_i|\mathbf{x}) \geq \hat{P}_n(\omega_j|\mathbf{x})$$

Substituting in the above expression for the estimate, we get the following equivalent decision rule: choose ω_i if $k_i \geq k_j$; in other words, if there are more training vectors in $\mathcal{R}_n$ from class ω_i than any other class, choose ω_i as the true state of nature.

An approximation to this rule is the *nearest neighbor decision rule*: find the training vector that is nearest $\mathbf{x}$, and choose the class of this training vector as the class for $\mathbf{x}$. It can be shown that this nearest neighbor decision rule has the follow upper bounds on the probability of error:

$$P_{E_{NN}} \leq 2P_{E_{\text{Bayes}}}$$

$$P_{E_{NN}} \leq P_{E_{\text{Bayes}}} \left(2 - \frac{c}{c-1} P_{E_{\text{Bayes}}}\right)$$

4 Dimensionality Reduction

Non-parametric estimation methods become quite difficult to use with feature vectors of high dimensionality, particularly when a large number of training vectors is not available. One method of reducing the dimensionality of the space is to project the training vectors onto a line chosen to maximize the difference between classes.

For this development, we assume that there are two sets of training data $\mathcal{X}_1$ and $\mathcal{X}_2$, with n_1 and n_2 training vectors. We will project the vectors in these two sets onto a line represented by the vector $\mathbf{w}$; the inner product $\mathbf{w}^T\mathbf{x}$ gives the position of $\mathbf{x}$ projected onto the line. By projecting the vectors in $\mathcal{X}_1$ and $\mathcal{X}_2$ onto $\mathbf{w}$, we obtain two sets of scalars $\mathcal{Y}_1$ and $\mathcal{Y}_2$. Our goal is to choose $\mathbf{w}$ so as to maximize the distance between these two sets.

One measure of the distance between the sets $\mathcal{Y}_1$ and $\mathcal{Y}_2$ is the distance between their sample means. We denote the sample mean of $\mathcal{Y}_i$ as

$$\begin{aligned}
\tilde{m}_i &= \frac{1}{n_i} \sum_{y \in \mathcal{Y}_i} y \\
&= \frac{1}{n_i} \sum_{\mathbf{x} \in \mathcal{X}_i} \mathbf{w}^T\mathbf{x} \\
&= \mathbf{w}^T \left(\frac{1}{n_i} \sum_{\mathbf{x} \in \mathcal{X}_i} \mathbf{x} \right) \\
&= \mathbf{w}^T \mathbf{m}_i
\end{aligned}$$

where $\mathbf{m}_i$ is the sample mean of $\mathcal{X}_i$. The distance between sample means is

$$|\tilde{m}_1 - \tilde{m}_2| = \left|\mathbf{w}^T\left(\tilde{\mathbf{m}}_1 - \tilde{\mathbf{m}}_2\right)\right|$$

Note that we can make this distance arbitrarily large by scaling $\mathbf{w}$. Thus, to find a measure of distance that is invariant to the magnitude of $\mathbf{w}$, we also define the *scatter* $\tilde{s}_i^2$ of each set $\mathcal{Y}_i$:

$$\tilde{s}_i^2 = \sum_{y \in \mathcal{Y}_i} \left(y - \tilde{m}_i\right)^2$$

Note that the scatter is the unnormalized sample variance of $\mathcal{Y}_i$.

We define a criterion function $J(\mathbf{w})$ as the following:

$$J(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

This is a measure of the separation of $\mathcal{Y}_1$ and $\mathcal{Y}_2$ that is invariant to the magnitude of $\mathbf{w}$. We wish to find $\mathbf{w}$ to maximize $J(\mathbf{w})$; to do so we define the following scatter matricies:

$$S_i = \sum_{\mathbf{x} \in \mathcal{X}_i} \left(\mathbf{x} - \mathbf{m}_i\right)\left(\mathbf{x} - \mathbf{m}_i\right)^T$$

$$S_W = S_1 + S_2$$

$$S_B = \left(\mathbf{m}_1 - \mathbf{m}_2\right)\left(\mathbf{m}_1 - \mathbf{m}_2\right)^T$$

S_W is called the within class scatter, and S_B is called the between class scatter. With these definitions, we can see that

$$\tilde{s}_1^2 + \tilde{s}_2^2 = \mathbf{w}^T S_1 \mathbf{w} + \mathbf{w}^T S_2 \mathbf{w} = \mathbf{w}^T S_W \mathbf{w}$$

$$\left(\tilde{m}_1 - \tilde{m}_2\right)^2 = \mathbf{w}^T\left(\mathbf{m}_1 - \mathbf{m}_2\right)\left(\mathbf{m}_1 - \mathbf{m}_2\right)^T \mathbf{w} = \mathbf{w}^T S_B \mathbf{w}$$

The criterion function can be written as

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_W \mathbf{w}}{\mathbf{w}^T S_B \mathbf{w}}$$

This is immediately recognized as a generalized Rayleigh quotient; the $\mathbf{w}$ that maximizes $J(\mathbf{w})$ must satisfy the following generalized eigenvalue problem:

$$S_B \mathbf{w} = \lambda S_W \mathbf{w}$$

If S_W has an inverse, this problem can be converted to a conventional eigenvalue problem

$$S_W^{-1} S_B \mathbf{w} = \lambda \mathbf{w}$$

Rather than solve this problem directly, we observe that

$$S_B \mathbf{w} = \left(\mathbf{m}_1 - \mathbf{m}_2\right)\left(\mathbf{m}_1 - \mathbf{m}_2\right)^T \mathbf{w} = \alpha\left(\mathbf{m}_1 - \mathbf{m}_2\right)$$

where $\alpha = (\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{w}$ is a scalar. Thus, the minimizing $\mathbf{w}$ is

$$\mathbf{w} = \frac{\alpha}{\lambda} S_W^{-1} (\mathbf{m}_1 - \mathbf{m}_2)$$

Since the magnitude of $\mathbf{w}$ does not affect the value of $J(\mathbf{w})$, we can ignore the scalar constant $\frac{\alpha}{\lambda}$ and write

$$\mathbf{w} = S_W^{-1} (\mathbf{m}_1 - \mathbf{m}_2)$$

For the case where we have c training sets $\mathcal{X}_1$ through $\mathcal{X}_c$, we project the d dimensional feature vectors onto a $c-1$ dimensional space. This projection can be represented as a $d \times c - 1$ matrix W:

$$W = \left[\begin{array}{ccc} \mathbf{w}_1 & \ldots & \mathbf{w}_{c-1} \end{array} \right]$$

The projection can be written as

$$\mathbf{y} = W^T \mathbf{x}$$

We begin to find the matrix W by defining the following matrices:

$$S_i = \sum_{\mathbf{x} \in \mathcal{X}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T$$

where $\mathbf{m}_i$ is the sample mean of $\mathcal{X}_i$. We also define a total mean vector $\mathbf{m}$ and a total scatter matrix S_T as

$$\mathbf{m} = \frac{1}{n} \sum_{\mathbf{x}} \mathbf{x} = \frac{1}{n} \sum_{i=1}^{c} n_i \mathbf{m}_i$$

$$S_T = \sum_{\mathbf{x}} (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T$$

We define the between class scatter S_B as

$$S_B = S_T - S_W = \sum_{i=1}^{c} n_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T$$

We define the objective function $J(W)$ as

$$J(W) = \frac{\left| W^T S_B W \right|}{\left| W^T S_W W \right|}$$

To find the columns $\mathbf{w}_i$ of the matrix W, we must solve for the largest eigenvalues of the generalized eigenvalue problem

$$S_B \mathbf{w}_i = \lambda_i S_W \mathbf{w}_i$$

In order to solve for $\mathbf{w}_i$, one can solve the following equation for λ_i

$$|S_B - \lambda_i S_W| = 0$$

and then solve

$$(S_B - \lambda_i S_W) \mathbf{w}_i = 0$$