# Classification by Density Estimation

The class predicted for $\vec{x}$ by the optimal Bayes classifier is

$$\underset{g \in \mathcal{G}}{\arg\max}\ \pi_g f_g(\vec{x}),$$

where $\mathcal{G}$ is the set of classes, $\pi_g$ is the prior prob. for class $g$, and $f_g$ is the density for class $g$. In most cases, the prior probabilities and the densities are unknown. One typically guesses the values of the prior probabilities or estimates them in a relatively straightforward manner (e.g., using sample proportions

from the learning sample). Assuming that good values can be obtained for the prior probabilities, one could closely approximate the performance of the Bayes classifier provided that the densities can be well estimated. The accurocy of density estimators depends on the sample size, the dimension (number of predictor variables), and the nature of the density being estimated. Often some other method of classification works better than attempting to estimate the densities directly and approximate the Bayes classifier, but in some low-dimensional

settings the simple density estimation method can work well if the sample size isn't too small.

There are many methods for density estimation. Kernel density estimation is a simple and widely-used method. For the one-dimensional version with a Gaussian kernel we have

$$\widehat{f_\lambda(x)} = \frac{1}{n} \sum_{i=1}^{n} \phi_\lambda (x - x_i),$$

where $\phi_\lambda$ denotes the pdf of Gaussian dist'n having mean 0 and standard deviation $\lambda$. If the density being estimated isn't too different from a normal

dist'n density, then setting $\lambda$ equal to

$$\frac{1.06\ s_x}{\sqrt[5]{n}}$$

is a good choice. But this value may be far from optimal for a lot of nonnormal distns, and it may be good to use a non-parametric method to determine the value to use for $\lambda$. (Note: One may also use another kernel, such as the popular Epanechnikov kernel. But typically the choice of kernel doesn't make an appreciable difference.)

In regions where the data is thinly spread, the density estimates may be

rather poor, and the decision rule can have odd characteristics. (See Fig. 6.14 in HTF.)

Kernel density estimation can be done in dimensions greater than 1, but beyond 2 or 3 dimensions, very large sample sizes may be needed to get decent performance. In many smaller sample size settings, one might be better off fitting a normal mixture model, using the data to estimate the parameters associated with Gaussian mixture distributions instead of trying to estimate the densities directly. Such a scheme for classification is in a sense a generalization of LDA.

The naive Bayes classifier also makes use of density estimation. Instead of trying to accurately estimate a multi-dimensional joint density with too little data, an assumption of independence is made. It is assumed that the components of

$$\vec{X} = (X_1, X_2, \ldots, X_p)$$

are ind. r.v's, so that the joint pdf of $\vec{X}$ is just equal to the product of the marginal pdfs. That is, one assumes

$$f_{\vec{x}}(\vec{x}) = \prod_{i=1}^{p} f_{X_i}(x_i).$$

Then the sample of size n can be used to estimate univariate densities instead of a joint density. If the assumption is true,

then the method can work very well — much better than LDA and QDA for severely nonnormal distns. Even when the assumption is not true, which is usually the case, the method can sometimes be competive with other somewhat simple methods.

A special case of the naive Bayes classifier is sometimes used, but I don't recommend it. One can assume that all of the densities are Gaussian and just use the data to estimate the means and variances. My guess is that QDA, which assumes normality, but not independence, will typically do better.