Chapter 4. Classification methods
Part 4

March 31, 2007

Suppose each sample $X = (\mathbf{x}_1, ..., \mathbf{x}_p)$ belongs to one of J classes, J is 2 or more. Denote the classes by $C_j, j = 1, ..., J$. How can we do the classification?

1 k-Nearest-Neighbor Techniques (kNN)

The nearest neighbor method (Fix and Hodges (1951), see also Cover and Hart (1967)) represents one of the simplest and most intuitive techniques in the field of statistical discrimination. It is a nonparametric method, where a new observation is placed into the class of the observation from the learning set that is closest to the new observation, with respect to the covariates used. The determination of this similarity is based on distance measures.

Formally this simple fact can be described as follows: Let

$$L = \{(X_i, y_i), i = 1, ..., N_L\}$$

be a training or learning set of observed data, where $y_i \in C_j, j = 1, ..., J$ denotes class membership and the vector $X_i = (x_{i1}, ..., x_{ip})^\top$ represents the predictor values. The determination of the nearest neighbors is based on a distance function $d(.,.)$, also called metric. Then for a new observation x the nearest neighbor $(y_{(1)}, X_{(1)})$ within the learning set is determined by

$$d(x, X_{(1)}) = \min(d(x, X_i))$$

and $\hat{y} = y_{(1)}$, the class of the nearest neighbor, is selected as prediction for y. The notation $X_{(k)}$ and $y_{(k)}$ here describes the kth nearest neighbor of x and its class membership, respec-

tively. For example, such typical distance functions are the Euclidean distance (metric)

$$d(X_i, X_j) = ||X_i - X_j|| = \{\sum_{s=1}^{p}(x_{is} - x_{js})^2\}^{1/2}$$

or its absolute distance

$$d(X_i, X_j) = \sum_{s=1}^{p} |x_{is} - x_{js}|.$$

In general, both measures can be seen as special cases of the so-called Minkowski distance

$$d(X_i, X_j) = \{\sum_{s=1}^{p} |x_{is} - x_{js}|^q\}^{1/q}.$$

The method has been explained by the random occurrence of the learning set, as described in Fahrmeir et al. (1996). The class label $y_{(1)}$ of the nearest neighbor $x_{(1)}$ of a new case x is a random variable. So the classification probability of x into class $y_{(1)}$ is $P(y_{(1)}|X_{(1)})$. For large learning sets x and $X_{(1)}$ coincide very closely with each other, so $P(y_{(1)}|X_{(1)}) \approx P(y|x)$ results approximately. For a new observation x is predicted as belonging to the true class y with the probability approximately $P(y|x)$.

The above idea can also be called 1-nearest neighbor classification. (what is the problem with this method? over-fit)

A first extension of this idea, that is widely and commonly used in practice, is the so-called k-nearest neighbor method. Here not only the closest observation within the learning set is referred for classification, but also the k most similar cases. The parameter k has to be selected in practice (e.g. 5, 6, 7,..). We shall discuss this later. Then the decision is in favour of the class label, most of these neighbors belong to.

Let k_r denote the number of observations from the group of the nearest neighbors, that belong to class C_j:

$$\sum_{j=1}^{J} k_j = k$$

Then a new observation is predicted into the class ℓ with

$$\ell \text{is the class corresponding to the largest numbers in } k_j.$$

This is equivalent to calculate the frequencies

$$\hat{p}_j = \frac{k_j}{k}$$

and

$$\ell \text{is the class corresponding to the largest value in } \hat{p}_j.$$

This prevents one single observation from the learning set deciding the predicted class.

1.1 Selection of k

Suppose we have only training set $\{(X_i, Y_i) : i = 1, ..., n\}$. The degree of locality of this technique is determined by the parameter k: For $k = 1$ one gets the simple nearest neighbor method as maximal local technique; For k being the total number of observation in the training set, a global majority vote of the whole learning set results. This implies a constant prediction for all new observations, that have to be classified: Always the most frequent class within the learning set is predicted.

For a working k, for each observation ℓ in the training set, consider data $\{(X_i, Y_i) : i \neq \ell\}$ and find the k nearest neighbor of X_ℓ, and counted the number of observations in each class C_j,

$$k_{\ell,1}, ..., k_{\ell,J}$$

Suppose k_{ℓ,j_ℓ} is the largest. then observation ℓ is classified as in class j_ℓ. The classification error is then

$$err(\ell) = \begin{cases} 0, & \text{if observation } \ell \text{ correctly classified} \\ 1, & \text{if observation } \ell \text{ correctly classified} \end{cases}$$

The cross-validation error is then defined as

$$CV(k) = n^{-1} \sum_{\ell=1}^{n} err(\ell).$$

We select k such that it minimizes

$$CV(k)$$

We can also define the CV based on the predicted probability. Following the above notation, define

$$p_{\ell,1} = \frac{k_{\ell,1}}{k}, ..., p_{\ell,J} = \frac{k_{\ell,J}}{k}$$

The classification error is then

$$err(\ell) = \sum_{j=1}^{J} \{Y_{\ell,j} - p_{\ell,j}\}^2$$

and

$$CV = n^{-1} \sum_{\ell=1}^{n} err(\ell)$$

2 Weighted k-Nearest-Neighbors (wkNN)

This extension is based on the idea, that such observations within the learning set, which are particularly close to the new observation (y, X), should get a higher weight in the decision than such neighbors that are far away from (y, X). This is not the case with kNN: Indeed only the k nearest neighbors influence the prediction; however, this influence is the same for each of these neighbors, although the individual similarity to (y, X) might be widely different. To reach this aim, the distances, on which the search for the nearest neighbors is based in the first step, have to be transformed into similarity measures, which can be used as weights.

we introduce a distance within the k nearest neighbor

$$D(x, X_{(i)}) = \frac{d(x, X_{(i)})}{d(x, X_{(k)})}$$

and calculate the frequencies

$$\hat{p}_j = \frac{\sum_{i=1}^{k} w_i I(y_{(i)} \in C_j)}{\sum_{i=1}^{k} w_i}$$

with

$$w_i = K(D(x, x_{(i)}))$$

Here, K is a kernel function. It can be Gaussian, Epanechnikov and others.

[The connection with KN kernel method?]

3 Adaptive k-Nearest-Neighbors (ADkNN)

Nearest neighbor techniques are based on the assumption that locally the class probabilities $P(C_j|x)$ are approximately constant.

In high-dimensions, nearest neighbors are far away causing bias and degrading performance. Adapt metric used in k-NN, so that resulting neighborhoods stretch out in directions in which the class probabilities change the least.

As an example, Figures 1 and 2 show two classes in two dimensions, Class 1 almost completely surrounds Class 2. The modified neighborhood extends further parallel to the decision boundaries and shrinks the neighborhood in the direction orthogonal to the decision boundary.

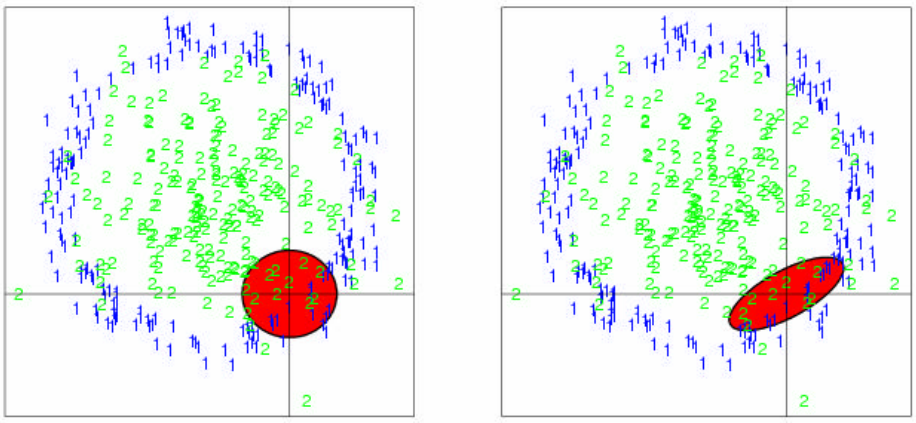

Figure 1: Illustration example

3.1 Discriminant-Adaptive Nearest Neighbour(DANN)

Instead of using

$$d(x, x') = ||x - x'||$$

we use

$$d(x, x') = \{(x - x')^\top \Sigma (x - x')\}^{1/2}$$

where

$$\Sigma = W^{-1/2}[W^{-1/2} B W^{-1/2} + \epsilon I]W^{-1/2}$$

where $W = \sum_{j=1}^{J} \pi_j W_j$, π_j is the frequencies of observations in class C_j,

$$W_j = k_j^{-1} \sum_{X_i \in C_j} (X_i - \bar{X}_j)(X_i - \bar{X}_j)^\top \qquad \text{(within-class covariance matrix)},$$

$$B = \sum_{j=1}^{J} \pi_j (\bar{X}_j - \bar{X})(\bar{X}_j - \bar{X})^\top \qquad \text{between class class covariance matrix}$$

and $\bar{X}_j = k_j^{-1} \sum_{X_i \in C_j} X_i$ and $\bar{X} = \sum_{i=1}^{k} X_i$.

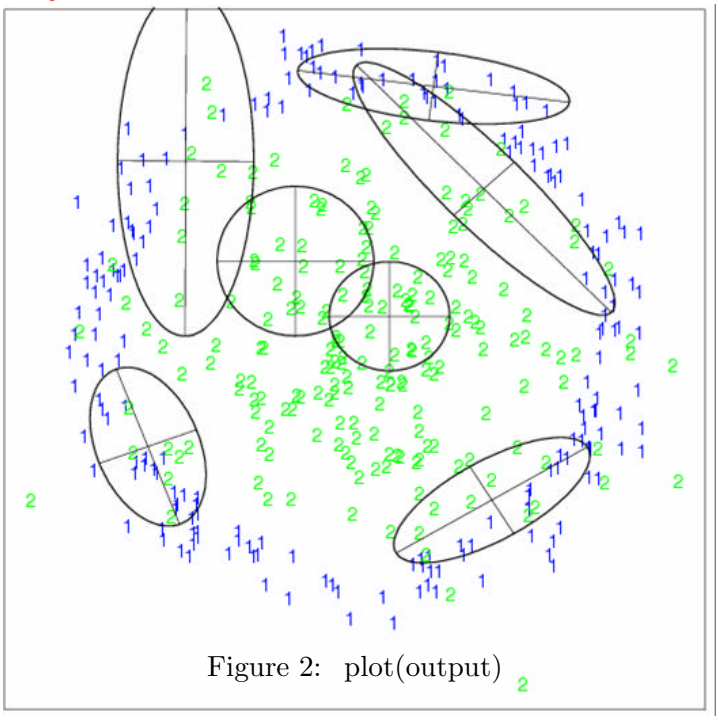

Figure 2: plot(output)

The classification procedure

1. Initialize the metric $\Sigma = I$
2. Spread out the nearest neighborhood of k_m points around the test point x_0, in the metric Σ (k_m should be large)
3. Calculate the weighted between and within sum-of-square matrices W and B using the points in the neighborhood.
4. Define a new metric

$$\Sigma = W^{-1/2}[W^{-1/2}BW^{-1/2} + \epsilon I]W^{-1/2}$$

5. Iterate steps 2,3 and 4
6. After completion using Σ for K-NN classification at the test point x_0.

Example 3.1 *Speaker independent recognition of the eleven steady state vowels of British English using a specified training set of lpc derived log area ratios. (***(training set)** , **(validation set)***), we use SVM and fda to classify the data. The response variable has 11*

categories. There are 10 covariates $\mathbf{x}_1, ..., \mathbf{x}_{10}$*. we use the training data to estimate the separating plane and validation set to check the methods.*

*SVM method: The error rate for the testing set is 0.3831169 (using kernel='radial', gamma = 0.3) (***(code)***)*

*FDA method: The error rate for the testing set is 0.4935065 (using method = mars, degree = 2); 0.5692641 (using method = ppr, nterms = 2); (***(code)***)*

*CART method: The error rate for the testing set is 0.6082251 (***(code)***)*

*kNN method: The error rate for the testing set is 0.4891775 (using kernel = "gaussian", distance = 2) (***(code)***)*

Example 3.2 *The Waveform data With 300 (***(training points)***, and 500* **(validation points)***),*

*SVM method: The error rate for the testing set is 0.164 (using kernel='radial') (***(code)***)*

FDA method: The error rate for the testing set is 0.192 (using method = mars, degree = 2)

*kNN method: The error rate for the testing set is 0.22 (using kernel = "gaussian", distance = 2) (***(code)***)*

Example 3.3 *Classification in genetics For the leukemia gene expression data (***(training points)***. There are 38 cells with 250 genes (selected from about 7000 genes). they are from two types of cells.*

We arbitrarily choose cell 21-33 for test and the others for learning

*kNN method: All are correctly clasified (using kernel = "triangular", distance =2) (***(code)***)*