3 Nearest Neighbour Based Classifiers

Learning Objectives

This chapter is an introduction to nearest neighbour classifiers. At the end of the chapter, you will understand:

- What a nearest neighbour(NN) algorithm is
- The different variants of NN algorithms like
  - The k nearest neighbour(kNN) algorithm
  - The modified k nearest neighbour (MkNN) algorithm
  - The fuzzy kNN algorithm
  - The r near algorithm
- The use of efficient algorithms
- What is meant by data reduction
- The different methods of prototype selection used in classification like
  - The minimal distance classifier(MDC)
  - The condensed nearest neighbour(CNN) algorithm
  - The modified condensed nearest neighbour (MCNN) algorithm
  - Editing methods

One of the simplest decision procedures that can be used for classification is the nearest neighbour (NN) rule. It classifies a sample based on the category of its nearest neighbour. When large samples are involved, it can be shown that this rule has a probability of error which is less than twice the optimum error—hence there is less than twice the probability of error compared to any other decision rule. The nearest neighbour based classifiers use some or all the patterns available in the training set to classify a test pattern. These classifiers essentially involve finding the similarity between the test pattern and every pattern in the training set.

3.1 Nearest Neighbour Algorithm

The nearest neighbour algorithm assigns to a test pattern the class label of its closest neighbour. Let there be n training patterns, $(X_1, \theta_1), (X_2, \theta_2), ..., (X_n, \theta_n)$, where X_i is of dimension d and θ_i is the class label of the ith pattern. If P is the test pattern, then if

$$d(\mathrm{P}, \mathrm{X}_k) = \min\{d(\mathrm{P}, \mathrm{X}_i)\}$$

where $i = 1 \ldots n$. Pattern P is assigned to the class θ_k associated with X_k.

EXAMPLE 1

Let the training set consist of the following three dimensional patterns:

$$\begin{array}{lll}
\mathrm{X}_1 = (0.8, 0.8, 1), & \mathrm{X}_2 = (1.0, 1.0, 1), & \mathrm{X}_3 = (1.2, 0.8, 1) \\
\mathrm{X}_4 = (0.8, 1.2, 1), & \mathrm{X}_5 = (1.2, 1.2, 1), & \mathrm{X}_6 = (4.0, 3.0, 2) \\
\mathrm{X}_7 = (3.8, 2.8, 2), & \mathrm{X}_8 = (4.2, 2.8, 2), & \mathrm{X}_9 = (3.8, 3.2, 2) \\
\mathrm{X}_{10} = (4.2, 3.2, 2), & \mathrm{X}_{11} = (4.4, 2.8, 2), & \mathrm{X}_{12} = (4.4, 3.2, 2) \\
\mathrm{X}_{13} = (3.2, 0.4, 3), & \mathrm{X}_{14} = (3.2, 0.7, 3), & \mathrm{X}_{15} = (3.8, 0.5, 3) \\
\mathrm{X}_{16} = (3.5, 1.0, 3), & \mathrm{X}_{17} = (4.0, 1.0, 3), & \mathrm{X}_{18} = (4.0, 0.7, 3)
\end{array}$$

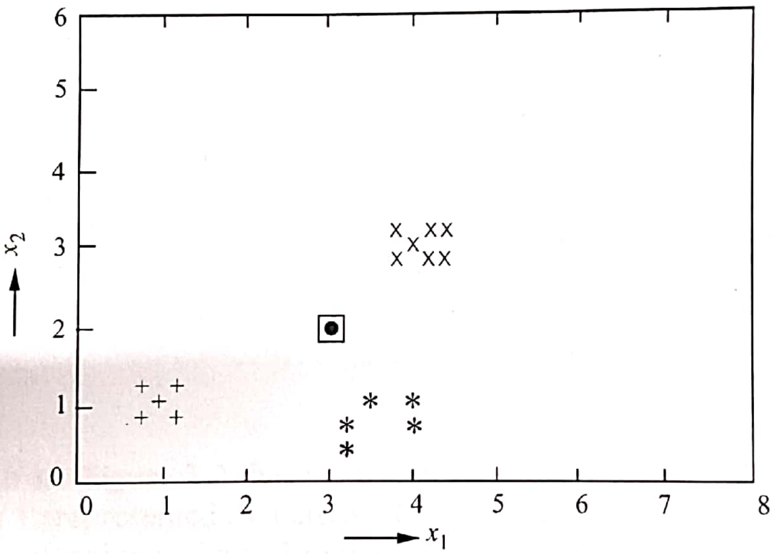

Figure 3.1 Example data set

For each pattern, the first two numbers in the triplets gives the first and second features, and the third number gives the class label of the pattern.

This can be seen plotted in Figure 3.1. Here "+" corresponds to Class 1, "X" corresponds to Class 2 and "*" corresponds to Class 3.

Now if there is a test pattern P = (3.0, 2.0), it is necessary to find the distance from P to all the training patterns.

Let the distance between X and P be the Euclidean distance

$$d(\mathrm{X}, \mathrm{P}) = \sqrt{(X[1] - P[1])^2 + (X[2] - P[2])^2}$$

The distance from a point P to every point in the set can be computed using the above formula. For P = (3.0, 2.0), the distance to X_1 is

$$d(X_1, P) = \sqrt{(0.8 - 3.0)^2 + (0.8 - 2.0)^2} = 2.51$$

We find, after calculating the distance from all the training points to P, that the closest neighbour of P is X_{16}, which has a distance of 1.12 from P and belongs to Class 3. Hence P is classified as belonging to Class 3.

3.2 Variants of the NN Algorithm

3.2.1 k-Nearest Neighbour (kNN) Algorithm

In this algorithm, instead of finding just one nearest neighbour as in the NN algorithm, k neighbours are found. The majority class of these k nearest neighbours is the class label assigned to the new pattern. The value chosen for k is crucial. With the right value of k, the classification accuracy will be better than that got by using the nearest neighbour algorithm.

EXAMPLE 2

In the example shown in Figure 3.1, if k is taken to be 5, the five nearest neighbours of P are X_{16}, X_7, X_{14}, X_6 and X_{17}. The majority class of these five patterns is class 3.

This method will reduce the error in classification when training patterns are noisy. The closest pattern of the test pattern may belong to another class, but when a number of neighbours are obtained and the majority class label is considered, the pattern is more likely to be classified correctly. Figure 3.2 illustrates this.

EXAMPLE 3

It can be seen from Figure 3.2 that the test point P is closest to point 5 which is an outlier in Class 1 (represented as a cross). If kNN algorithm is used, the point P will be classified as belonging to Class 2 represented by circles.

Choosing k is crucial to the working of this algorithm. For large data sets, k can be larger to reduce the error. The value of k can be determined by experimentation, where a number of patterns taken out from the training set(validation set) can be classified using the remaining training patterns for different values of k. It can be chosen as the value which gives the least error in classification.

EXAMPLE 4

In Figure 3.1, if P is the pattern (4.2, 1.8), its nearest neighbour is X_{17} and it would be classified as belonging to Class 3 if the nearest neighbour algorithm is used. If the 5 nearest neighbours are taken, it can be seen that they are X_{17} and X_{16}, both belonging to Class 3 and X_8, X_7 and X_{11}, belonging to Class 2. Following the majority class rule, the pattern would be classified as belonging to Class 2.

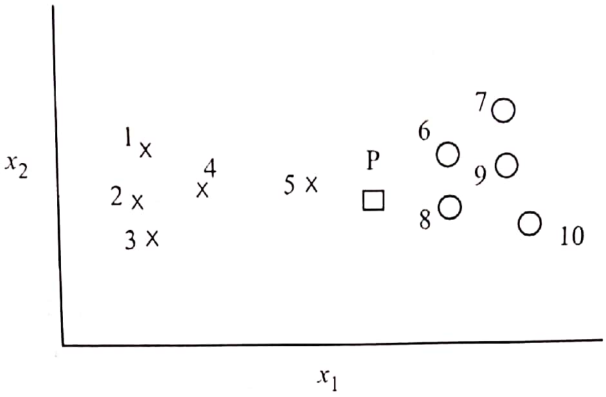

Figure 3.2 P can be correctly classified using the kNN algorithm

3.2.2 Modified k-Nearest Neighbour (MkNN) Algorithm

This algorithm is similar to the kNN algorithm, inasmuch as it takes the k nearest neighbours into consideration. The only difference is that these k nearest neighbours are weighted according to their distance from the test point. It is also called the distance-weighted k-nearest neighbour algorithm. Each of the neighbours is associated with the weight w which is defined as

$$w_j = \begin{cases} \frac{d_k - d_j}{d_k - d_1} & \text{if } d_k \neq d_1 \\ 1 & \text{if } d_k = d_1 \end{cases}$$

where $j = 1, .., k$. The value of w_j varies from a maximum of 1 for the nearest neighbour down to a minimum of zero for the most distant. Having computed the weights w_j, the MkNN algorithm assigns the test pattern P to that class for which the weights of the representatives among the k nearest neighbours sums to the greatest value.

Instead of using the simple majority rule, it can be observed that MkNN employs a weighted majority rule. This would mean that outlier patterns have lesser effect on classification.

EXAMPLE 5

Consider P = (3.0, 2.0) in Figure 3.1. For the five nearest points, the distances from P are