CHAPTER 13

Object Recognition

Science has been arranging, classifying, methodizing, simplifying, everything except itself.

–Elihu Root

LEARNING OBJECTIVES

One of the principal aims of image processing is the recognition of the objects present in images. After the image features are extracted from the segmented objects, pattern recognition tasks must be carried out for identification of the object. This chapter deals with pattern recognition tasks such as classification and clustering. After studying this chapter, the reader will become familiar with the following:

- Concept of classifiers
- Evaluation of classifiers
- Structural and syntactic recognition methods
- Clustering algorithms

13.1 PATTERNS AND PATTERN CLASSES

The human brain has a unique capability called categorization. *Categorization* is the ability to assign a meaningful label (or pattern class) to an object. The process of assigning a meaningful label to an unknown object is known as *recognition*. For example, when we encounter an animal in a picture or in reality, we register its identity using its features. When an unknown animal is encountered, we try to recognize it by comparing its features (called patterns) with known stored patterns that we already have. This process of comparing an unknown object with stored patterns to recognize the unknown object is called *classification*. Thus, classification is the process of applying a label or pattern class to an unknown instance. In the absence of any prior knowledge of the object or stored pattern, we use a trial-and-error process to recognize the object. This trial-and-error process of grouping of objects is called *clustering*. Figure 13.1 illustrates the steps in a typical object recognition process.

As discussed in Chapters 1 and 2, the image is acquired first. Then the region of interest (ROI) is identified. As discussed in Chapter 9, the ROI varies with applications. If the application is a diagnostic system, say, for cancer identification in a mammogram, the ROI would be a lesion. Segmentation algorithms are used to extract the lesion from the background.

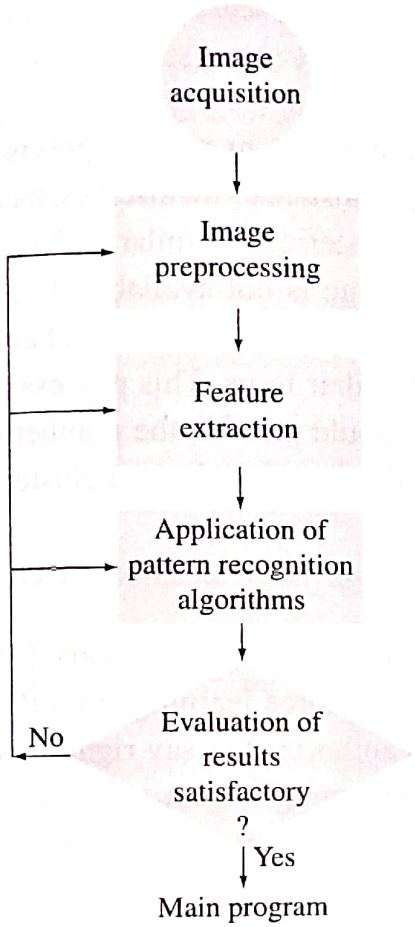


Fig. 13.1 Typical object recognition process

As discussed earlier, feature extraction is a step for extracting the features relevant to the application. This includes the size, shape, location, and other visual features necessary for the application. The feature vector or pattern vector is a vector that contains the features that are extracted.

A feature vector is typically of the form

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} \quad \text{or} \quad x = [x_1, x_2, \ldots, x_{n-1}, x_n]^T$$

Feature extraction algorithms, as discussed in Chapter 12, are used to extract object features. Feature selection algorithms such as PCA, are then used prior to the formulation of the feature vector. Principal component analysis (PCA) is useful in identifying the most relevant features that are necessary for identifying the objects.

The next stage as shown in Fig. 13.1 is pattern recognition, where the feature vectors are learned and recognized.

An important component in pattern recognition is the ability of the system to learn from the data. Learning means the development of algorithms by acquiring knowledge from the given empirical data. Some of the important types of learning are supervised and unsupervised learning.

13.1.1 Supervised Learning

A simple model of supervised learning is provided in Fig. 13.2.

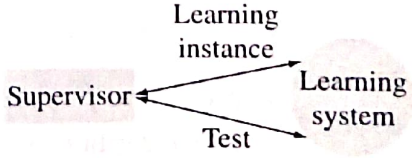


Fig. 13.2 Supervisory system

The teacher/supervisor provides a label/cost for each pattern in a training set. Based on the training set, the system learns and generates a concept to classify the pattern. Once the system becomes a learnt system, the supervisor supplies the test data, using which it tests the system. This kind of learning, where there is an interaction between the supervisor and the learning system, is called supervised learning.

13.1.2 Unsupervised Learning

There is no explicit teacher or supervisor component in an unsupervised system. The learning system itself learns by trial and error. The instances themselves, based on similarity measures, form groups or clusters. The goal of clustering is similar to that of classification. However, it is performed when domain knowledge is not available. For example, when we encounter an animal that we have not seen earlier, we often use trial and error to group the unknown animal with the closest animal familiar to us. This process is called natural grouping. In unsupervised learning, the users should provide the number of clusters they desire. Thus, natural grouping varies based on the number of initial clusters desired.

13.1.3 Reinforced Learning

In this method, the output of the learning system is binary. The binary feedback of right/wrong is sent back to the input and is used to reinforce learning from the data. The role of reinforced learning is that of a critic who is authorized to say right or wrong. Thus the learning continues till the critic agrees with the learning system.

13.2 TEMPLATE MATCHING

Template matching is one of the simplest techniques in object recognition, where the target object to be identified is defined as a template. The template is then superimposed on and correlated with the image. At every pixel, the degree of similarity is evaluated. This technique is also often referred to as matched filtering. The correlation is high when there is a perfect match between the template and the image. Based on the highest correlation value, the degree of match can be determined.

Let us assume that $f(x, y)$ is the given image and $w(x, y)$ is the template. The correlation of the template and the image is given as

$$c(x,y) = \sum_{\alpha}\sum_{\beta} w(\alpha,\beta) f(x+\alpha, y+\beta)$$

where α and β represent the shifts of the correlation template.

The template is superimposed on the image and image correlation is performed. The correlation between the template and the image replaces the centre pixel of the mask in the

resultant image. It is then moved to the adjacent position. This process is repeated till the centre of w visits all the pixels of the image. At the end of this process, the maximum value indicates the best match. The biggest disadvantage of this scheme is that no variation in scale or orientation is permitted.

Example 13.1 Consider the template and image array shown in Figs 13.5(a) and 13.5(b). Perform template matching and show the result.

Solution The resultant correlation array is as shown in Fig. 13.3(c).

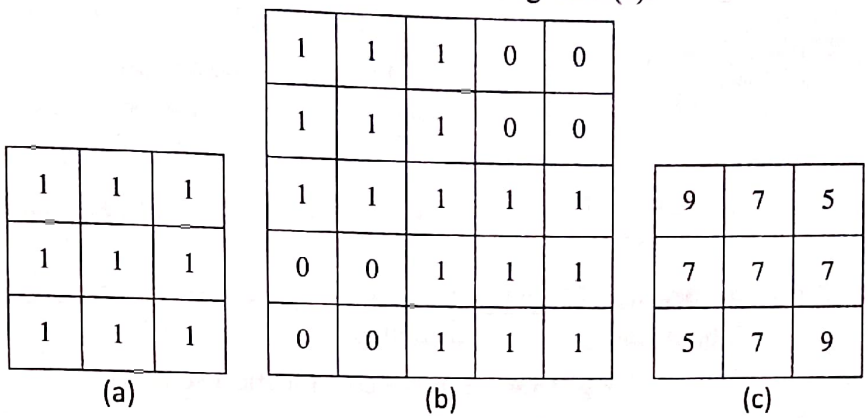


Fig. 13.3 Template matching (a) Template (b) Image array (c) Resultant correlation array

It can be observed that each element of the result is obtained by counting the number of similar values between the template and the image array. The highest value (9 in this case) indicates the position where there is a perfect match between the template and the image.

The biggest disadvantage of template matching is that no variation in scale or orientation is permitted. In addition, this scheme, when used for higher dimensions, involves large calculations and hence feature-based schemes are preferred over template matching schemes.

13.3 INTRODUCTION TO CLASSIFICATION

Classification is a supervised learning method. How will you separate an apple from a sweet lemon? To classify fruits, first an idea of the fruits involved—apple and sweet lemon—is required, with some parameters such as colour, size, and shape. Thus a dataset needs to be constructed to train a classifier on how an apple looks. These attributes are called input features, attributes, or independent variables. The input also includes labels of known instances. So a classifier takes a set of features as input and learns what features characterize an apple or a sweet lemon. Then it takes an unknown instance (which needs to be classified) and assigns the label 'apple' or 'sweet lemon'. This label is a dependent variable and this process is called classification, prediction, or recognition. A class is supposed to be predefined and non-overlapping, and should completely partition the entire dataset evenly.

How do classifiers achieve this? A classification model is supposed to 'learn' the complex relationships that exist between the input image features using the training data. This resultant learning process is known as a *concept* or *model*. After the learning stage is over, the classifier is called a 'learnt system' and produces a classification model. So when an unknown instance is given as input, and if the classifier assigns the label correctly, it is known as a correct classification; otherwise it is termed as an incorrect classification and implies that more inputs are required to train the classifier. A sample classification scheme is shown in Fig. 13.4.

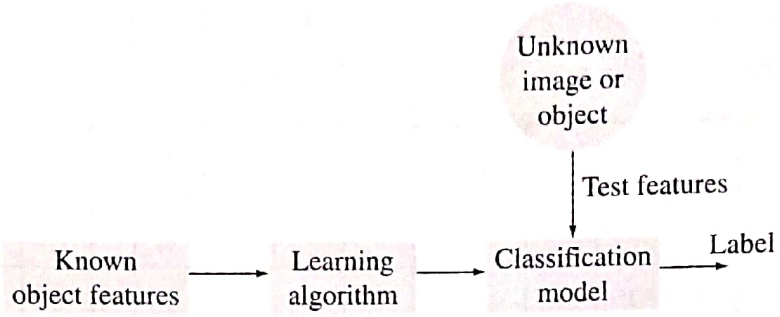


Fig. 13.4 Sample classification scheme

Hence the classification process involves the following two phases:

1. Training phase
2. Testing phase

In the training phase, the classifier algorithm is fed with a large set of known data. This dataset is called training data or labelled data. It is called so because the labels of known instances are given along with their attributes. The training data should generally be large and representative in nature. For example, if the application is about diagnosis of cancer, then all types of cancer images should be given as input for the classifier to be effective. Once the training phase is over, a data-driven classification model is created. The second phase is called the testing phase where the constructed model is tested and evaluated with unknown test data. The classification task is both descriptive as well as predictive, that is, if the model can explain its classification decisions, it is called descriptive. Decision tree-based classifiers are descriptive in nature, whereas neural networks-based classification schemes generally cannot explain their decisions.

Based on the results obtained, the performance of the classifier is evaluated. Real-world classifiers are known to be unstable and ill-defined, as a small change in the test data may cause a large variation in the result. In addition, there is no single classifier available that works well with all sorts of data. Misclassification also involves risks. For example, a cancer diagnosis classifier system producing a wrong prediction result may cause legal and social problems. Thus, the performance evaluation of a classifier is a tricky issue and assumes larger significance.

13.3.1 Factors Affecting Classifier Performance

Generally the performance of the classifier depends on factors such as the nature of data and the nature of learning.

Nature of data A classification model depends on the availability of good quality training data. Most of the poor classification results are related to the non-availability of good quality data. For example, a classifier for a rare disease may suffer from non-availability of good quality data, as many samples cannot be obtained. Another problem is that of missing data; the missing values may cause classification problems and affect the performance of the classifier. Missing data may be unintentional (as data may not be available at the time of data entry) or deliberate (as, often, the user may not be willing to provide personal data).

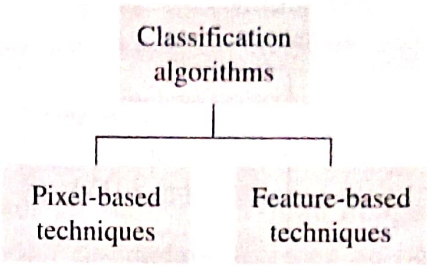


Fig. 13.5 General classification schemes

Nature of learning The learning process should not take more data than necessary (known as over-fitting of the model). For example, if testing is carried out with the training data itself, the classifier may yield high predictive results which do not convey much about the real ability of the classifier. Over-fitting leads to generalization error.

There are many applications of classifiers in image processing. Good examples of classification are face recognition, iris recognition, signature verification, and speaker identification. As shown in Fig. 13.5, the classifier may be either pixel-based or feature-based.

In pixel-based techniques, the input to the classifier is raw pixel data. For example, in remote sensing, the application may be the classification of sand and sea regions present in an image. To accomplish this, the classifier can be fed with several training images that have pixel data of sea and sand regions. The classifier is expected to learn from the raw training images and generate the concept of sand or sea. In the testing stage, the classifier can be tested with test images. This kind of classification using raw pixel data is called a pixel-based technique. Feature-based techniques, on the other hand, extract the features of the image such as size, shape, location, and texture, which can then be used for classification. Both techniques are essentially similar as a single classifier may do both pixel-based and feature-based classification. For example, the Bayesian classifier can be used for both pixel and feature classifications. The difference between the methods lies only in the nature of data.

13.3.2 Classifier Design

There are many ways to design a classifier. Some of the popular design techniques are shown in Fig. 13.6.

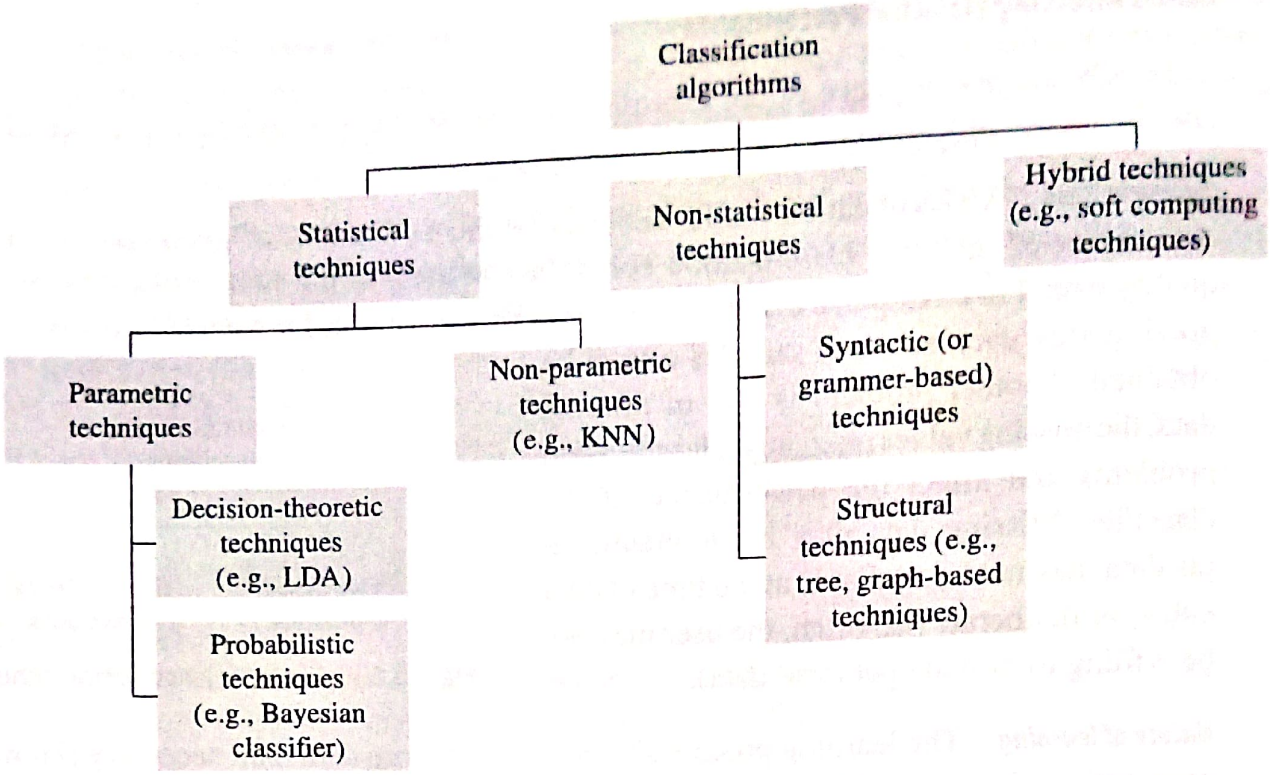


Fig. 13.6 Types of classification algorithms

Statistical classifiers use statistical principles for deriving models from the given training dataset using statistical learning techniques. Statistical classifiers are of two types:

Parametric classifier Parametric classifiers take a set of training data and construct a classification model. It is assumed that the probability distribution function or density function of the data is known for each class, but not the specific statistical values such as mean and variance. So the parameters are estimated from the data itself by assuming a distribution for a given data. Bayesian classifier is a good example of parametric classifiers where the input data is assumed to be a Gaussian distribution and the statistical parameters such as mean and variance are estimated. Parametric classifiers are classified into the following two types based on the methods used for classification:

Decision-theoretic techniques Decision-theoretic techniques are often called discriminant function analysis. The idea is to design a decision function or a discriminant function that responds differently for each class, that is, its response for each class is unique. This decision function can be integrated with the decision rule for classification. Examples of decision-theoretic techniques are linear discriminant analysis, template matching, and Bayesian decision function-based techniques.

Probabilistic techniques Bayesian classifier is a popular probabilistic technique. Suppose, for example, a forest image has more pixels having green colour, it makes sense to predict an unknown pixel of that image as a green pixel. Thus probability plays an important role in

prediction. Bayesian classifier calculates the prior probability and conditional probability and uses these values to assign a label to the unknown instance.

Non-parametric classifier When nothing is known about the densities of the data, no assumptions can be made. In such a case non-parametric classifiers are useful. *K*-nearest neighbour is a popular non-parametric classifier.

Another approach is to use the structural methods classification technique. Many real-world applications are complex. Sometimes it may be difficult to identify the correct features for recognizing the object. The formulation of feature vector using statistical techniques is difficult. Hence structural techniques extract some basic patterns of the object and use those basic primitives for recognition.

Structural methods are of the following two types:

Syntactic or grammar-based techniques As discussed in Chapter 12, the objects of an image can be described using structural relations. Picture description language describes an object as a set of graphic primitives. So an object can be encoded using strings. Once strings are designed for an object, formal grammar can be designed for checking whether the structure is valid or not. Classification techniques such as string matching and shape matching are examples of this technique.

Structure-based techniques In this technique, graphs are used instead of strings. An object can be modeled as a graph. The object is then matched with another object using graph matching techniques.

Many instances in technical literature use the term syntactic recognition to refer to both types of structural methods. The subtle difference is that syntactic methods are string-based, whereas structure-based methods can involve complex data structures such as graphs and trees. Hybrid techniques are a combination of these two techniques. One popular technique is using soft computing paradigms such as fuzzy logic, neural networks, and genetic algorithms for classification.

Sections 13.4 and 13.5 discuss statistical and structural methods in detail. Soft computing techniques are covered in Chapter 14.

13.4 DECISION-THEORETIC METHODS

The decision-theoretic approach is often called discriminant function analysis. Let us assume that the classification is done for objects having only two features. To classify the 2-feature object, the two features (x and y) are plotted as a point in a 2D graph called the feature space. For objects having multiple features the graph is multidimensional. The idea now is to design decision boundaries or discriminating functions to separate the feature vector clusters in the feature space. For the 2D example, a discriminant function is a line

that can separate feature clusters. The decision function or the discriminant function is designed to give different responses for different classes. The algorithms that use this approach of the decision function are called discriminant analysis.

One of the oldest techniques used in the decision-theoretic approach in pattern recognition is the linear discriminant analysis (LDA), where the decision function was designed to classify an Iris flower dataset. The problem was one of classifying the input data of sepal length, sepal width, petal length, and petal width to three classes of *Iris setosa*, *Iris versicolor*, and *Iris virginica*. The idea of LDA is to use the decision functions to discriminate the input features. As discussed for the two features X_1 and X_2, the decision boundary would be $d(X) = X_1 - mX_2 - c$. The key idea is to design a decision surface such that $d(X) > 0$ would identify a particular feature and $d(X) < 0$ would identify another feature. All the points at the decision boundary would satisfy the condition $d(X) = 0$. This idea can also be extended for multiple features. Let $x = (x_1, x_2, \ldots, x_n)^T$ represent the n-dimensional vectors. Let the number of classifiers be k. The problem is to assign an unknown instance x to any one of the classes. This is done by designing k decision functions $d_1(x), d_2(x), \ldots, d_k(x)$. The instance is classified as class i and not j if

$$d_i(x) > d_j(x);\ i \neq j \text{ for } i, j = 1, 2, \ldots, k$$

Then the decision boundary separating two classes i and j is given as follows:

$$d_i(x) - d_j(x) = 0$$

A decision rule can be designed as follows: Assign the instance to the class w_i if $d_{ij} > 0$ and assign the instance to w_j if $d_{ij} < 0$. There are many ways in which the decision functions can be designed. The simplest method is to design a decision boundary perpendicular to the line that connects the mean of the class.

In general, the procedure of using decision functions is given as follows:

1. Compute the numerical values of all the discriminating functions for all the classes for the target vector x.
2. Choose a predicted class for x that is associated with a discriminant function whose value is the largest.

Another simple classifier that can be conceived is minimum-distance classifier. In this classifier, the idea is to assign the unknown instance x to a class if the distance between the unknown sample x and the prototype vector of the classes is minimum.

The procedure is as follows:

1. Find the mean vector of all the classes. The mean vector of a pattern class is given as

$$m_j = \frac{1}{N_j} \sum_{x \in w_j} x_j;\ j = 1, 2, \ldots, k$$

Here, N is the number of vectors from class j.

2. Use the Euclidean distance and compute the distance between the unknown instance and the mean vector. Hence this is equivalent to

$$d_j = \|x - m_j\|$$

The norm can be defined as $\|a\| = (a^{\mathrm{T}} a)^{\frac{1}{2}}$.

This is equivalent to

$$d_j = x^{\mathrm{T}} m_j - \frac{1}{2} m_j^{\mathrm{T}} m_j; \text{ for } j = 1, 2, ..., k$$

The problem now becomes assigning an unknown instance x to either class i or class j if this is a two-class problem. Similarly, the distance function for class i can be calculated as follows:

$$d_i = x^{\mathrm{T}} m_i - \frac{1}{2} m_i^{\mathrm{T}} m_i; \text{ for } i = 1, 2, ..., k$$

Hence the decision boundary d_{ij} can be calculated as $d_i(x) - d_j(x)$

For $n = 2$, the dividing decision function is a line and for $n = 3$, it is a plane and if $n > 3$, it is called hyper plane.

Example 13.2 Let us assume the mean vectors of two classes 1 and 2 respectively are $m_1 = (3.2, 1.2)$ and $m_2 = (3, 1.0)$. What is the decision boundary?

Solution The decision function d_1 is

$$d_1(x) = x^{\mathrm{T}} m_1 - \frac{1}{2} m_1 m_1^{\mathrm{T}}$$

$$= (x_1\ x_2)^T\ (3.2\ 1.2) - \frac{1}{2}(3.2\ 1.2)(3.2\ 1.2)^T$$

$$= 3.2x_1 + 1.2x_2 - 5.84$$

Similarly, the decision function d_2 can be designed as

$$d_2(x) = x^T m_2 - \frac{1}{2} m_2 m_2^T$$

$$= (x_1\ x_2)^T\ (3\ 1) - \frac{1}{2}(3\ 1)(3\ 1)^T$$

$$= 3x_1 + x_2 - 5$$

Therefore, the decision boundary is given as

$$d_{12}(x) = d_1(x) - d_2(x)$$
$$= 0.2x_1 + 0.2x_2 - 0.84$$

If the decision function $d_{12}(x) > 0$, the instance is assigned to class 1. Otherwise, the instance is assigned to the class 2.

Example 13.3 Consider two classes of features with mean and variance as {1, 3} and {5, 5}. Design the decision function.

Solution Decision function $d_1(x)$

$$d_1(x) = x^T m_1 - \frac{1}{2} m_1 m_1^T$$

$$= (x_1 x_2)^T (1,3) - \frac{1}{2}(1,3)(1,3)^T = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}(1,3) - \frac{1}{2}(1,3)\begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

$$= x_1 + 3x_2 - \frac{1}{2}(1+9) = x_1 + 3x_2 - \frac{10}{2} = x_1 + 3x_2 - 5$$

Decision function $d_2(x)$

$$d_2(x) = (x_1 \ x_2)^T (5 \ 5) - \frac{1}{2}(5 \ 5)(5 \ 5)^T$$

$$= 5x_1 + 5x_2 - 25$$

$\therefore$ Decision boundary

$$d_{12} = d_1(x) - d_2(x)$$
$$= x_1 + 3x_2 - 5 - 5x_1 - 5x_2 + 25 = -4x_1 - 2x_2 + 20$$

$\therefore$ Decision boundary $= 2x_1 + x_2 - 10$

13.5 BAYESIAN CLASSIFIERS

The Bayesian classifier is one of the popular classifier models widely used in image processing. There are three different types of Bayesian classifiers available, as shown in Fig. 13.7.

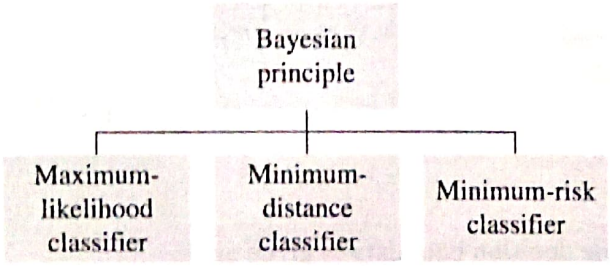


Fig. 13.7 Types of Bayesian classifiers

The most popular version of the Bayesian classifier is maximum-likelihood classifier. The Bayesian classifier requires the following three pieces of information:

1. $P(i)$—Prior probability of the class
2. $P(x/i)$—Conditional probability that class i has x. This can be calculated from the training data table
3. $P(x)$—Sum of $P(x/i)$ over the entire dataset. This information is not probability information, but serves as a normalization factor.

What is Bayesian principle?

As per the Bayesian principle, one can find the inverse probability $P(i/x)$ from $P(x/i)$ and $P(i)$. The Bayes theorem can be given as

$$P\left(\frac{i}{x}\right)=\frac{P\left(\frac{x}{i}\right)P(i)}{P(x)}$$

The prior probability of the class can be obtained from the training set. In images, the prior probability can be estimated by plotting a histogram of the image. For example, in an image that has both sand and sea regions, one can estimate the presence of sand pixels approximately by observing the histogram. This is called prior probability. Similarly, the information $P(x/i)$ can be estimated from the training dataset. $P(x)$ is same for all classes i and hence can be ignored.

Maximum likelihood classifier Now, the Bayesian rule can be applied as the only term that is unknown is P(i/x). The Bayesian optimality rule states that any instance assigned to a wrong class is worse and all types of misclassifications are equally worse. Therefore, according to the Bayesian maximum likelihood classifier, the instance is assigned to class i for which P(i/x) is maximum.

The advantages of the Bayesian classifiers are that they are easy to use, require only one scan of the training set, are not affected much by missing values, and produce good results for datasets with simple relationships.

13.5.1 Naïve Bayesian Classifier

If the features or attributes are assumed to be independent, the resulting classifier is called naïve Bayesian classifier.

The algorithm for the Bayesian classifier is stated as follows:

1. Train the classifier with training images or labelled featured data.
2. Compute the probability $P(i)$ using intuition, based on experts' opinion, or using histogram-based estimation.
3. Compute $P(i/x)$.
4. Find the maximum $P(i/x)$ and assign the unknown instance to that class.

Naïve Bayesian classifier does not work for real-time datasets as it is naïve to assume that the features are independent of each other and also naïve Bayesian classification does not work for continuous data. Let us discuss these issues now.

Example 13.4 Let us assume a simple dataset, as shown in Table 13.1. Let us apply the Bayesian classifier to predict (2, 2).

Table 13.1 Sample dataset

a_1	a_2	Class (i)
2	0	c_1
0	2	c_1
2	4	c_2
0	2	c_2
3	2	c_2

Solution Here $c_1 = 2$ and $c_2 = 3$ from the training set. Therefore the prior probabilities are $P(c_1) = 2/5$ and $P(c_2) = 3/5$. The conditional probability is estimated.

$$P(a_1 = 2/c_1) = 1/2;\ P(a_1 = 2/c_2) = 1/3$$

$$P(a_2 = 2/c_1) = 1/2;\ P(a_2 = 2/c_2) = 2/3$$

Therefore,

$$P(x/c_1) = P(a_1 = 2/c_1) \times P(a_2 = 2/c_1)$$

$$= 1/2 \times 1/2 = 1/4$$

$$P(x/c_2) = P(a_1 = 2/c_2) \times P(a_2 = 2/c_2)$$

$$= 1/3 \times 2/3 = 2/9$$

This is used to evaluate

$$P(c_1/x) = P(c_1) \times P(x/c_1)$$

$$= 2/5 \times 1/4 = 2/20 = 0.1$$

$$P(c_2/x) = P(c_2) \times P(x/c_2)$$

$$= 3/5 \times 2/9 = 6/45 = 0.13$$

Since $P(c_2/x) > P(c_1/x)$, the sample is predicted to be in class c_2.

Example 13.5 Let us consider a classification problem that involves classification of an image pixel using a single feature colour into two classes—forest and non-forest. Let the prior probability of the forest class be 0.6, the feature i of colour green belonging to the forest image in the training set be 0.2, and the probability of the green pixel feature belonging to the forest in the overall population be 0.4. What is the probability that an image is a forest image given that the image contains the green colour feature?

Solution For the two given classes, the only feature commonly available is colour. Let the feature be x. So the available information is

(a) Prior probability of the class $P(i)$ is 0.6
(b) Conditional probability that the class i has $x = P(x/i) = 0.2$
(c) $P(x) = 0.4$

So as per Bayesian theorem, $P\left(\frac{i}{x}\right) = \frac{P\left(\frac{x}{i}\right) \cdot P(i)}{P(x)} = \frac{0.2 \times 0.6}{0.4} = \frac{0.12}{0.4} = 0.3$

Example 13.6 Use Naïve Bayes classifier and classify the unknown pixel X. There are two classes of pixels # and * present in the image as shown:

$$\begin{pmatrix} \# & \# & \# & * \\ \# & X & \# & * \\ \# & \# & * & * \\ \# & \# & * & * \end{pmatrix}$$

Consider the 8-neighbourhood of X and determine the class of X.

Solution The prior probabilities are $P(\text{Pixel} = \text{'\#'}) = \frac{\text{Number of \# pixels}}{\text{Total number of pixels}} = \frac{9}{16} \cong 0.56$

$$P(\text{Pixel} = \text{'*'}) = \frac{\text{Number of * pixels}}{\text{Total number of pixels}} = \frac{6}{16} \cong 0.38$$

Given the 8-neighbourhood of X, it is possible to calculate the likelihood of X.
Likelihood of X given '#' in the 8-neighbourhood

$$= \frac{\text{Number of \# pixels in neighbourhood of } X}{\text{Total number of '\#' pixels}} = \frac{7}{9} \cong 0.78$$

Likelihood of X given '*' in the 8-neighbourhood

$$= \frac{\text{Number of * pixels in neighbourhood of } X}{\text{Total number of '*' pixels}} = \frac{1}{6} \cong 0.17$$

Now posterior probability can be calculated as

[Prior probability P(pixel = '#')] × [Likelihood of X given '#' in the 8-neighbourhood]

$$= 0.56 \times 0.78 = 0.4368$$

[Prior probability P(pixel = '*')] × [Likelihood of X given '*' in the 8-neighbourhood]

$$= 0.38 \times 0.17 = 0.0646$$

Since 0.4368 is greater than 0.0646, the pixel X must be #.

13.5.2 Bayesian Classifier for Continuous Attributes

For a training set of many images, finding $P(x/i)$ is a difficult exercise. Therefore, it is easier to approximate this as a function with fewer parameters. The approximation of the input data is in the form of a Gaussian distribution. This kind of approximation is called parametric approximation, and can be written as

$$P\left(\frac{x}{i}\right) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x-m_i)^2}{2\sigma_i^2}}$$

Here m_i and σ_i are the mean and standard deviation of class i.

An alternate way is to find a decision function to separate the classes as follows:

$$P\left(\frac{i}{x}\right) = P\left(\frac{j}{x}\right)$$

By applying the Bayes theorem and assuming that x is normally distributed, the decision boundary for this problem can be designed as

$$P(i) \times \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-m_i}{\sigma_i}\right)^2} = P(j) \times \frac{1}{\sigma_j\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-m_j}{\sigma_j}\right)^2}.$$

By simplifying this, this decision boundary would be as follows:

$$D = -2\ln\left(\frac{P(i)}{\sigma_i}\right) + \left(\frac{x-m_i}{\sigma_i}\right)^2 + 2\ln\left(\frac{P(j)}{\sigma_j}\right) + \left(\frac{x-m_j}{\sigma_j}\right)^2$$

Example 13.7 Assume that there are two classes i and j. The mean and variance for class i is given as (24, 2) and for class j as (20, 3). Assume that the data is distributed in Gaussian distribution. How will you classify the instance $x = 22$. Assume that the prior probability of these two classes is 1/2.

Solution The calculations are as follows:

$$P\left(\frac{x=22}{i}\right) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{22-m_i}{\sigma_i}\right)^2} = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{22-24}{2}\right)^2}$$
$$= 0.1207$$

Similarly

$$P\left(\frac{x=22}{j}\right) = \frac{1}{\sigma_j\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{22-m_j}{\sigma_j}\right)^2} = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{22-20}{3}\right)^2}$$
$$= 0.1065$$

Now by substituting this in the Bayesian formula, we get

$$P\left(\frac{i}{x}\right)=\frac{P(i)P\left(x=\frac{22}{i}\right)}{P(i)\times P\left(x=\frac{22}{i}\right)+P(j)\times P\left(x=\frac{22}{j}\right)}$$

$$=\frac{0.0604}{0.0604+0.0532}=0.5317$$

Therefore, the instance is likely to be classified as class i as its probability $P\left(\frac{x}{i}\right)$ is more than 0.5.

The decision function can be obtained by substitution and simplification to get

$$D=5x^2-272x+3354.8076$$

Solving this yields two roots, 21.8261 and 32.5739. Since $x = 22$ lies between these two roots, the instance is classified as class i.

Generally real-word problems involve objects having multiple attributes. So a set of features is used in the form of a feature vector. If we assume that there are k classes, the formula becomes

$$P\left(\frac{i}{x}\right)=\frac{P(i)P\left(\frac{x}{j}\right)}{\sum_{j=1}^{k}P(j)P\left(\frac{x}{j}\right)}$$

If this data $P(x)=\sum_{j=1}^{k}P(j)P\left(\frac{x}{j}\right)$ is assumed to be a Gaussian distribution, then the multivariate normal density is given as follows:

$$P(x)=\frac{1}{\sqrt{(2\pi)^d \det\Sigma}}e^{-\frac{1}{2}(x-m)^T\Sigma(x-m)}$$

Here, d is the dimension. If there are more features involved, the mean becomes a mean vector and Σ, a covariance matrix.

13.5.3 Bayesian Minimum Distance Classifier

Bayesian minimum distance classifier, which is based on the Mahalanobis distance, is assigned to the class in which $P(i/x)$ is minimum.

In the formula,

$$P\left(\frac{x}{i}\right)=P(i)\frac{1}{\sqrt{(2\pi)^d \det\Sigma_i}}e^{-\frac{1}{2}(x-m_i)^T\Sigma_i^{-1}(x-m_i)}$$

If prior probability is same for all classes, then factors like logarithm and the term $\frac{1}{\sqrt{(2\pi)^d \det \Sigma_i}}$ can be ignored as it is a scaling factor. The factor $(x - m_i)^T \Sigma_i^{-1} (x - m_i)$ is called Mahalanobis distance.

The minimum distance classifier based on Mahalanobis distance can be given as follows:

1. Compute the mean vector for all the classes for a given x.
2. Compute the Mahalanobis distance

   $(x - m_i)^T \Sigma_i^{-1} (x - m_i)$ for all classes $i = 1, 2, \ldots, k$, assuming k class problem

   for unknown instance x and mean vector of all classes.
3. Assign x to a class for which the associated Mahalanobis distance is minimum.

The calculation of the Mahalanobis distance is computationally intensive. Therefore, it can be replaced by simple distance measures such as Euclidean distance or city block distance. However, Mahalanobis distance is a proven measure with regard to reliability.

13.5.4 Bayesian Minimum Risk Classifier

Another useful extension of the Bayesian classifier is the minimum risk classifier. A cost function can be assigned to the classification. In case of any misclassification, such as if the instance is assigned to a wrong class, a penalty can be assigned so that the misclassification can be avoided in future. Such a cost function is called loss function. The cost of the decision is estimated based on the nature of the application in which the classifiers are used. For example, in a fingerprint application for criminal identification or in a medical application, a misclassification can be fatal. The estimated cost or loss function is multiplied with the posterior probabilities for taking the final decision of assigning a label for the unknown instance. Then a decision rule can be designed as follows:
Assign an instance x to class i if

$$\text{Loss}\left(\frac{\alpha_2}{i}\right) \times P\left(\frac{i}{x}\right) > \text{Loss}\left(\frac{\alpha_1}{j}\right) \times P\left(\frac{j}{x}\right);$$

otherwise, assign the instance x to j.

Here α_1 and α_2 are the costs of the decisions.

13.6 K-NN CLASSIFIER

Similarity measures can be used to determine 'alikeness' of different tuples in the databases. In this, a representative of every class is selected. The classification is performed by

assigning each tuple to the class to which it is more similar. Let us assume that the classes are $\{C_1, C_2, \ldots, C_n\}$ and the training dataset D has $\{t_1, t_2, \ldots, t_n\}$ tuples. The idea is to assign unknown instance t_i to class C_j such that the similarity measure of (t, C_j) is greater than or equal to the similarity measure of (t, C_j), where C_i is not equal to C_j. The similarity can be obtained using distance measures. The algorithm can be stated as follows:

1. Choose the representative of the class. Normally, the centre or centroid of the class is chosen as the representative of the class.
2. Compare the test tuple and the centre of the class.
3. Classify the tuple to the appropriate class.

Example 13.8 Find the nearest neighbour of the following:

(a) 25 in the list {1 3 15 20 40 50} in 1D

(b) (1, 3) in the list {(1,1),(3,3),(5,5)} in 2D

Solution The neighbour can be determined by the distance between the elements.

(a) The element 25 is closer to 20 as its difference is less than the other elements. So 20 is the closest 1-neighbour of 25. The three closest neighbours of 25 are 20, 15, and 40.

(b) Any distance measure can be used. If Euclidean distance is used, then the Euclidean distance between (1, 3) and {(1,1), (3,3), (5,5)} are 2, 2, and 3 respectively. So the nearest neighbours of (1, 3) are {(1,1), (3,3)}

This procedure can be generalized as KNN algorithm. Here K is an integer. KNN algorithm is as follows:

1. Pick a suitable value for K.
2. Identify the K neighbours for the unknown instance that needs to be classified.
3. Take the majority class of the K neighbours as the class of the target unknown instance.

So the logic is that the class of the unknown instance is the majority class of the closest neighbours as chosen by K.

Example 13.9 The following points, co-ordinates, and classes are given:

Point	Coordinates	Class
x_1	(2, 0)	Class 1
x_2	(4, 2)	Class 1
x_3	(2, 3)	Class 1
x_4	(-1, 2)	Class 2
x_5	(-2, 3)	Class 2

Classify the point (1, 1) using nearest neighbour technique with K=1.

Solution Find Euclidean distance between the given features and (1, 1) $D_e = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ where (x_1, y_1) and (x_2, y_2) are the two points between which we have to calculate the distance.

$$(2,0)(1,1) \Rightarrow D_e = \sqrt{(2-1)^2+(0-1)^2} = \sqrt{2} = 1.414$$

$$(4,2)(1,1) \Rightarrow D_e = \sqrt{(4-1)^2+(2-1)^2} = \sqrt{(9+1)} = \sqrt{10} = 3.1623$$

$$(2,3)(1,1) \Rightarrow D_e = \sqrt{(2-1)^2+(3-1)^2} = \sqrt{(1+4)} = \sqrt{5} = 2.236$$

$$(-1,2)(1,1) \Rightarrow D_e = \sqrt{(-1-1)^2+(2-1)^2} = \sqrt{(4+1)} = \sqrt{5} = 2.236$$

$$(-2,3)(1,1) \Rightarrow D_e = \sqrt{(-2-1)^2+(3-1)^2} = \sqrt{(9+4)} = \sqrt{13} = 3.605$$

The minimum distance is between (1, 1) and (2, 0) whose class is 1. Therefore, class of (1, 1) is 1. If $k = 3$, the there nearest points are computed and its majority class is assigned to the point. This is left as an exeicise to the reader.

13.7 REGRESSION METHODS

As discussed earlier, a classifier is about assigning a label to an instance. The label should be a categorical variable. However, if the output of a prediction is numerical, regression should be used. Regression is one of the methods used for numerical prediction. Regression analysis models the relationship between one or more independent variables (results) and a dependent variable (input attributes).

The simplest regression is fitting a line to a set of points. It can be described as

$$Y = W_0 + W_1 x,$$

where W_0 and W_1 are the weights of the regression coefficients. The coefficients can be calculated by the method of least squares to fit a line that minimizes the error between the actual data and the estimate. If D is the training set,

$$W_i = \frac{\sum_{i=1}^{|D|}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|}(x_i - \bar{x})^2}$$

$$W_0 = \bar{Y} - W_1 \bar{X}$$

where $\bar{x}$ and $\bar{y}$ are the mean values of data x and y, respectively.

Example 13.10 Consider the sample dataset in Table 13.2. What is its regression analysis?

Solution A linear regression model can be obtained as follows. A line can be fit to the given data as $y = W_0 + W_1 x$.

Table 13.2 Sample data for regression

Size (x)	Number of pixels (y)
1	1
2	4
3	9
4	16
5	25

$$\bar{x} = \frac{15}{5} = 3$$

$$\bar{y} = \frac{55}{5} = 11$$

$$W_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\begin{array}{r}(1-3)(1-11)+(2-3)(4-11)+(3-3)(9-11)\\ +(4-3)(16-11)+(5-3)(25-11)\end{array}}{[(1-3)^2+(2-3)^2+(3-3)^2+(4-3)^2+(5-3)^2]}$$

$$= \frac{60}{10} = 6$$

$$W_0 = \bar{y} - W_1 \times \bar{x} = 11 - (6 \times 3) = -7$$

The scatter plot and the regression line are shown in Fig. 13.8.

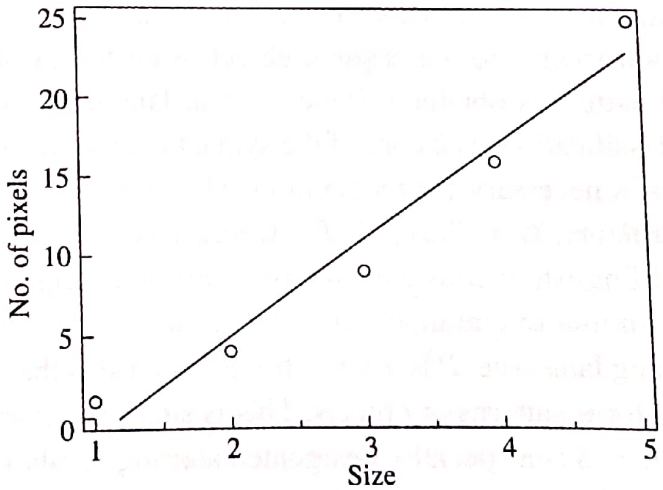


Fig. 13.8 Scatter plot and fitted line

By substituting these parameters, we get the regression equation for the dataset as

$$y = W_0 + W_1 x$$

that is,

$$y = -7 + 6x.$$

For example, when $x = 3$, this yields $y = -7 + 6(3) = 11$.

However, the actual value is 9. The difference of 2 is called the prediction error. The model is accurate when more data is present along the direction of the regression line. If three attributes are involved (say A_1, A_2, and A_3), the linear regression model would be

$$Y = w_0 + w_1x_1 + w_2x_2 + w_3x_3$$

Similarly, in image processing, nonlinear regressions are encountered. In such cases, one can apply transformations to convert nonlinear regression into linear regression models.

STRUCTURAL AND SYNTACTIC CLASSIFIER ALGORITHMS

Unlike statistical methods, structural methods exploit the relationships that exist among the basic elements of the objects. Any object can be decomposed into a set of basic primitives. In Chapter 10, we have discussed picture description languages that use graphic primitives to construct the object. Syntactic methods use strings, and structural algorithms use techniques such as graphs, to encode the objects and the problem of recognition becomes a matching problem. Some of the popular approaches are shape matching algorithms and string matching algorithms.

1 Syntactic Classifiers

The syntactic technique (also known as grammar-based or linguistic approach) uses small sets of pattern primitives and grammatical rules for recognizing the object. For example, in picture description language, the basic primitive is the graphic element. The idea is to decompose the object in terms of the basic primitives. This process of decomposing an object into a set of primitives is called *parsing*. The basic primitives can then be reconstructed to the original object using formal languages to check whether the recognized pattern is obtained. Hence formal language theory plays an important role in syntactic classification. The core of the syntactic approach is to design and develop a set of grammar rules necessary for recognition. The grammar in formal languages is defined as a 4-tuple structure, $G = \{T, N_T, S, P\}$, where T is a set of terminal symbols similar to the alphabets of English or constants of a programming language. N_T is a set of non-terminals or symbols, similar to grammar units such as nouns or verbs of English or the variables of a programming language. P is a set of production rules that define how the symbols can be combined to form patterns or objects. This is similar to phrases or clauses as per grammar rules in English. S is a specially designated starting symbol (or root), similar to a sentence in English and in a programming language. The set of sentences that are recognized as valid by the grammar G is called a language, $L(G)$.

The implementation of a syntactic classifier is the same as that of the statistical classifier. In the first phase, the syntactic classifier is given the training dataset of valid strings of known objects. The patterns are decomposed into basic patterns and the grammar necessary for combining the primitives to reconstruct the original object is identified in the training phase.

The second stage is the testing stage where unknown patterns are given to the grammar of the syntactic classification system. Each unknown pattern is decomposed into the basic primitives and checked using a parser. The parser recognizes the primitives as per the grammar rules similar to the checking of the English language statements using English grammar rules. For example, let us assume that there are two classes C_0 and C_1, and the languages that correspond to these two classes are $L(C_0)$ and $L(C_1)$, respectively. If the target sentence is recognized by $L(C_0)$, the pattern is assigned to class C_0. Similarly, if the sentence is recognized by $L(C_1)$, the pattern is assigned to class C_1. The target test sentence is declared invalid if both the grammars do not recognize it. However, if the sentence is parsed correctly by both the grammars, some sort of conflict recognition should be done to resolve the conflict. This two-class problem can also be extended to multiple classes.

13.8.2 Shape-matching Algorithms

Let us assume that the shapes A and B have shape numbers in the form of a string of chain codes. Let the string represent the shape characteristics of the boundary of an object. By this assumption, the shapes have a similarity of α if

$$S_j(A) = S_j(B) \text{ for } j = 4, 6, 8, \ldots, \alpha$$
$$S_j(A) \neq S_j(B) \text{ for } j = k+2, k+4, \ldots$$

Here, j is the order. The similarities are recorded in a matrix called similarity matrix. Another way is to use the distance measure for shape matching. The distance measure is given as the reciprocal of the similarity measure, which is given as $D(A, B) = \frac{1}{k}$, where $D(A, B)$ is the distance between two shapes A and B and k is the degree of similarity.

13.8.3 String-matching Algorithms

Let there be two regions, a and b. Assume that they are coded into two strings as follows:

$$a = \{a_1, a_2, \ldots, a_n\}$$
$$b = \{b_1, b_2, \ldots, b_n\}$$

Let us assume that $a_1 = b_1$, $a_2 = b_2$, etc. Let the position where there is no match, that is, $a_k \neq b_k$, be α. Then the following two measures can be defined:

1. The number of symbols that do not match

$$\beta = \max(|a|, |b|) - \alpha$$

where $|a|, |b|$ are the lengths of the strings a and b. α is the number of matches between these strings. If none of the symbols match, β is zero.

2. Degree of similarity $R = \frac{\alpha}{\beta} = \frac{\alpha}{\max(|a|, |b|) - \alpha}$

When the strings are the same, $R = \infty$. The value of R is high when there is a good match between the strings.

13.8.4 Rule-based Algorithms

Tree search is a popular approach that uses rules for classification. The simplest way of performing classification is to generate rules that are of the form IF (condition) THEN (conclusion). The IF part is called rule antecedent or precondition and the THEN part is called rule consequent. Decision rules are generated using a technique called 'covering algorithm' where the best attribute is chosen to perform the classification based on the training data. The algorithm chooses the best attribute that minimizes the error and uses that attribute in generating a rule.

A tree-based classifier is shown in Fig. 13.9. In this decision tree, every node can have only two children. The root is a specially designated node and all the other intermediate nodes of the tree represent the rule conditions; the leaves (children) of the tree are classes that are assigned to the instances. The unknown object or instance features are taken and their values are compared and validated with the conditions represented sequentially in the internal nodes of the tree. Tracing the path from the root to the assigned class gives the conditions that led to the classification of that instance. This knowledge can be encoded as a decision rule in the form of an *if-then-else* rule.

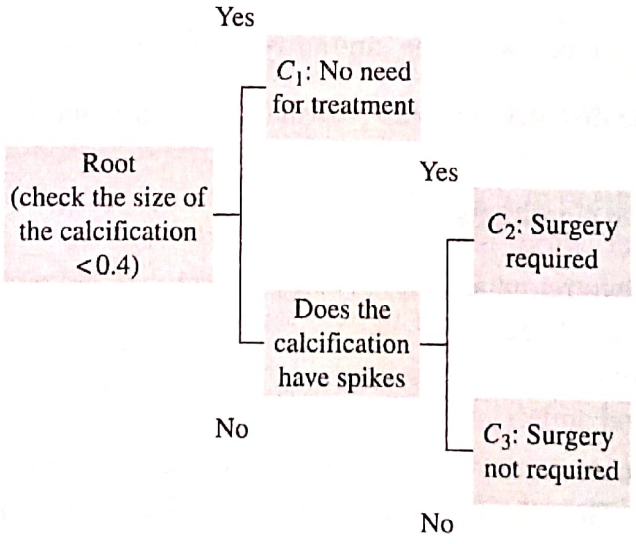


Fig. 13.9 Sample tree classifier

For example, let us assume two unknown cases related to a small application of classification of calcifications present in kidney. As per the doctor's suggestion, if calcification is less than 0.3, no medical treatment is generally required, medication is sufficient. However, if it is bigger and if the calcification has spikes, then surgery is required.

Let us assume two test cases.

1. Calcification < 0.3 mm
2. Calcification = 2 cm with spikes.

Then the first case belongs to class C_1 and second case belongs to class C_2.
Figure 13.11 shows that there are three classes, namely
(a) class C_1—no need for treatment,
(b) class C_2—surgery required, and
(c) class C_3—surgery not required.

The conditions of this tree classifier are the size of the calcification and the presence of spikes. The unknown features are checked at the internal nodes sequentially and based on the results of the condition, the successor nodes are chosen. The search is continued till the instance is assigned to a class. This approach is called top-down search. The problem can be visualized as a tree search problem. Some of the algorithms that are useful in tree search are DFS, BFS, and A* algorithms.

13.8.5 Graph-based Approach

The graph-based approach is an extension of the tree-based approach. Initially an object is modelled as a graph. Graph matching is a procedure of comparing two graphs and identifying whether they are similar or dissimilar. A match, either perfect or partial, gives an idea about the similarity between the objects. A graph G is defined as an ordered set of vertices and edges. Figure 13.10 shows a sample graph.

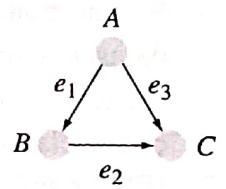
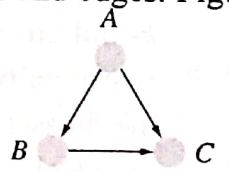


Fig. 13.10 Sample graph

This graph has three vertices $\{A, B, C\}$. The link between two vertices is called edge; the edges in the graph are $\{e_1, e_2, e_3\}$. A set of edges is called incidence function. The incidence function for the graph in Fig. 13.10 is $\{(A, B), (B, C), (A, C)\}$. Two graphs are said to be same or identical if



1. the vertices are the same,
2. the edges are the same,
3. the incidence functions are the same, and
4. the indegree and outdegree of each vertex in the two graphs are same.

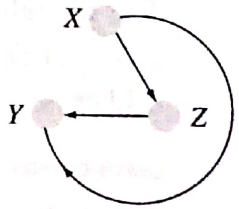


Fig. 13.11 Two similar but structurally different graphs

Figure 13.11 shows the fact that two graphs can be similar even though they are structurally different. Two graphs are said to be isomorphic if there is a bijective function (one-to-one mapping) between them.

A bijective function is shown in Fig. 13.12.

Graph matching is a difficult process. It involves trial-and-error. Initially, a subgraph $G_1(0)$ is considered trivially isomorphic to graph G_2. (A trivial graph is a single vertex with no edges.) Then the vertices are added one by one and the condition of isomorphism is checked. In case there is a mismatch, the process backtracks and replaces the node with some other node and the procedure is repeated. If there is a complete match, the graph is declared isomorphic. Otherwise, the graphs are considered as dissimilar graphs.

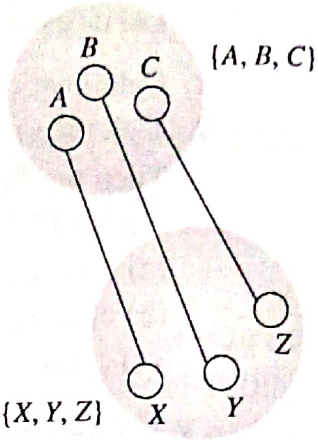


Fig. 13.12 Bijective function

13.9 EVALUATION OF CLASSIFIER ALGORITHMS

Accuracy is the ability of the classification models to correctly determine the class of an unknown test image. Classifiers are affected by noise and outliers present in the dataset. Some of the popular techniques for evaluation of the classification models are as follows:

Separate training/test sets This is one of the simplest methods for testing the classifier. The dataset is separated into two sets. One of them is called training set. The other set, called the test dataset, is used for testing the performance of the classifier. For better evaluation of the classifier, it is better to ensure that the test cases are different from the training set. If the number of instances in the testing set is N, and if the classifier correctly classifies C instances, the predictive accuracy of the classifier is C/N.

k-fold cross validation Another popular method that is used frequently is cross validation. k-fold cross validation is an improvement over the previous method. k is an integer and is generally taken as 10. The dataset is divided equally into k subsets. For example, if the dataset has 100 instances, then 10 datasets are created with 10 instances each. Each time a classifier is tested, $k - 1$ subsets are together considered as the training set and the remaining one set is treated as the test dataset. The process is then repeated for k trials. The overall performance of the classifier is the average error of misclassification of the classifier across all k trials. Since the value of k is normally 10, this process is called 10-fold cross validation.

Leave-one-out cross validation This is also called N-folding or jack-knifing technique. This is an extreme form where every instance is considered as a dataset. Then the N classifiers are generated and each of them is used to classify the single instance. The amount of computation involved is very large and therefore, this method is considered unsuitable for many real-world applications. The predictive accuracy of the classifier is defined as the ratio of correctly classified samples to total number of instances. Many objective metrics are available for quantifying the quality of the classifiers. Some of the useful parameters are as follows:

Accuracy This indicates the ability of the classifier to predict unknown instances.

Classification time This is the time the classifier takes for constructing the model (learning time) and the actual time taken for classification of an unknown instance (testing time). Lesser the learning time, more preferable the model.

Robustness The classifiers are known to be unstable. Noise, outliers, and errors/missing data often result in misclassification. Therefore, robustness or immunity of the classifier towards noise or missing data is an important criterion for evaluating the classifier.

Scalable The classifier should be able to handle large datasets.

Goodness of fit This is the measure that indicates the quality of the model generated from a given training data.

Generally, the results of the classifier are shown in the form of a table called confusion matrix. A simple two-class confusion matrix is shown in Table 13.5.

Table 13.3 Confusion matrix

Expert vs classifier		Predicted class as per classifier		Total number of instances
		+	−	
Actual class as per the expert	+	TP	FN	P
	−	FP	TN	N

Confusion matrix indicates the performance of the classifier in classifying the instances. Many objective metrics can be defined based on the confusion matrix. Some of the most important metrics are described as follows:

True positive rate or TP rate This metric is also referred to by various terms such as sensitivity, recall, and hit rate. It indicates the sensitivity of the classifier and is described as the probability that it will produce a true positive result. It can be calculated as $\frac{TP}{P}$, where $P = TP + FN$.

False positive rate or FP rate This term is also referred to as false alarm rate or specificity. It is the probability that a classifier produces erroneous results as positive results for negative instances. It can be calculated as $\frac{FP}{N}$, where $N = FP + TN$.

False negative rate or FN rate This is the probability that a classifier produces erroneous results as negative results for positive instances. It can be calculated as $\frac{FN}{P}$, where $P = TP + FN$.

True negative rate or TN rate This is the probability that a classifier produces results as negative results for negative instances. It can be calculated as $\frac{TN}{N}$, where $N = FP + TN$.

TP rate and TN rate indicate correct classifications and FP rate and FN rate indicate erroneous classifications.

Positive predictive value or precision This is the probability that an object is classified correctly as per the actual value. It is defined as $\frac{TP}{TP+FP}$.

Negative predictive value This is the probability that an object is not classified properly as per the actual value. It is defined as $\frac{TN}{TN+FN}$.

Accuracy This is also referred to as recognition rate. The accuracy of the classifier can be shown as $\frac{TP+TN}{TP+TN+FP+FN}$. It indicates the ability of a classifier to classify instances correctly.

Error rate This is also referred to as misclassification rate. The error rate of the classifier indicates the proportion of instances that have been wrongly classified, and is given as

$$\frac{FP+FN}{TP+TN+FP+FN}$$

The metrics can also be combined. For example, precision and recall (i.e., TP rate) can be combined to give a metric called F1 score, which is defined as

$$\text{F1 score} = (2 \times \text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall})$$

Unlike classifiers where the predicted variable is categorical, the regression predicted variable is numerical. So the metrics used to quantify the predictions are in terms of errors. Error is defined as the difference between the actual value y_i and the predicted value y_i'. This is also called loss. For example, if the actual value is 7.3 and regression predicts this as 7, then the loss is 0.3. This loss is called the error. The subtraction between the actual and predicted values may result in a negative value. To avoid such cases, the following errors are defined:

$$\text{Absolute error} = |y_i - y_i'| \quad \text{Squared error} = (y_i - y_i')^2$$

Absolute error and squared error indicate the loss functions. The loss for every instance of a dataset can be calculated. The sum of all the losses results in a metric called test error rate or generalization error. It indicates the average loss for the entire test dataset. The total number of instances in the dataset is denoted by d.

$$\text{Mean absolute error} = \frac{\sum_{i}^{d} |y_i - y_i'|}{d} \quad \text{Mean squared error} = \frac{\sum_{i=1}^{d} |(y_i - y_i')^2|}{d}$$

The performance of the classifier can also be visualized as a graph. The receiver operating characteristic (ROC) graph is an effective tool for visualization of classifier performance as well as comparing the performance of many classifiers. It is a 2D graph plot whose x-axis is the FP rate and y-axis is the TP rate. For a classifier, FP rate and TP rate can be plotted as an (x, y) value in a graph. Hence, for a classifier, the performance is a point in a graph. To compare two classifiers, the points need to be compared. If a classifier point is located, say, in the top north-west of another point, it is considered as a better classifier because the best classifier is the point given as (0, 1). This is shown in Fig. 13.13(a).

Some classifiers allow some of the parameters to be tuned. By tuning the parameters the classifier may result in multiple (x, y) values that can be plotted. This results in a curve called the ROC curve. Fig. 13.13(b) shows an ROC curve of a classifier. The ROC curve is helpful in understanding the tuning process that results in the best way of classification. The area under the curve indicates the accuracy of the model. A model is considered perfect if the area under the ROC curve is one. It is indicated by the point (0, 1). A classifier performance can be crudely compared with the best classifier represented as (0, 1) using a Euclidean distance formula given as

$$\text{Euclidean distance} = \sqrt{\text{FP rate}^2 + (1 - \text{TP rate})^2}$$

The Euclidean distance ranges from 0 (best classifier) to 1 (worst classifier).

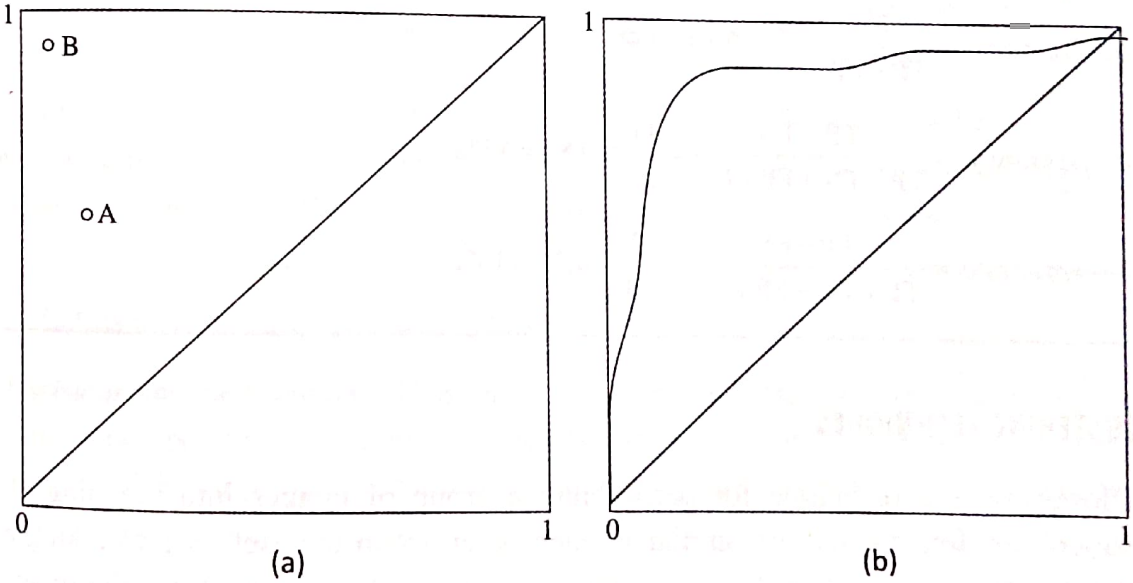


Fig. 13.13 ROC curve (a) Sample graph (diagonal line indicates the performance of the average classifier) (b) Sample ROC curve

Classifiers are now used in many applications. One of the major application domains where classifiers are used widely in image processing is biometrics. Section 13.7 illustrates the application of classifiers in biometrics.

Example 13.11 Let us assume that the classifier performance with respect to a dataset is as shown in Table 13.6. Evaluate the performance of the classifier.

Table 13.4 Classifier performance

Expert vs classifier		Predicted class as per classifier		Total number of instances
		+	−	
Actual class as per the expert	+	8 (TP)	1 (FN)	P = TP + FN = 8 + 1 = 9
	−	1 (FP)	3 (TN)	N = FP + TN = 1 + 3 = 4

Solution The performance of the classifier can be evaluated using the following metrics:

1. True positive rate (TP rate) $= \frac{TP}{P} = \frac{8}{9} \cong 0.89 = 89\%$
2. False positive rate (FP rate) $= \frac{FP}{N} = \frac{1}{4} = 0.25 = 25\%$
3. False negative rate (FN rate) $= \frac{FN}{P} = \frac{1}{9} = 0.11 = 11\%$
4. True negative rate (TN rate) $= \frac{TN}{N} = \frac{3}{4} = 0.75 = 75\%$
5. Precision $= \frac{TP}{TP+FP} = \frac{8}{9} \cong 0.89 = 89\%;$
6. Accuracy $= \frac{TP+TN}{TP+TN+FP+FN} = \frac{11}{13} \cong 0.85 = 85\%$
7. Error rate $= \frac{FP+FN}{TP+TN+FP+FN} = \frac{2}{13} \cong 0.15 = 15\%$

13.10 CLUSTERING TECHNIQUES

Clustering is a technique for partitioning a group of images into meaningful disjoint subgroups. Images that are similar to each other group themselves into a single cluster. All the images in a subgroup are similar to each other. At the same time, the images across the clusters are different. Cluster analysis is different from classification. Clustering is an example of unsupervised learning where there is no idea about the classes or clusters prior to clustering. Some of the important topics to be discussed with respect to clustering algorithms are as follows:

1. Method for finding the similarities and dissimilarities of the images
2. Categorization of clustering algorithms
3. Evaluation of clustering algorithms

13.10.1 Similarity Measures

Image clustering algorithms are based on the notion of similarity or dissimilarity between images. The term *proximity* can be used to denote similarity and dissimilarity together. Similarity measures are indicated by distance functions. Distance functions characterize how close one image is to another. A distance function is called a metric if it fulfils the following criteria:

1. $D(i,j) \geq 0$ for all i and j
2. $D(i,j) = 0$ if $i = j$
3. $D(i,j) = d(j,i)$ for all i and j
4. $D(i,j) \leq d(i,k) + d(k,j)$ for all i, j, and k

This property is called triangle inequality.

The distance measures depend on the data type of the objects involved in the clustering process. For convenience, some of the data types are mentioned in Table 13.5.

Table 13.5 Object data types

Data types	Example	Distance measures
Nominal (Categorical) variables	Identification number, label number	Distance measures involving matching
Binary variables	Variables that indicate the presence or absence of a feature, such as occurrence or non-occurrence of an event	Jaccard coefficient
Qualitative data	Shape number	Number of matches
Quantitative variables	Size, centroid, and area	Euclid, Manhattan, and Minkowski
Ordinal or ranked variables	If Grades = {S, A, B}, inherent ordering is present, as S > A > B	Rank matching

Let us now discuss these data types:

Nominal or categorical variables These are strings or labels that characterize the variables. The arithmetic operations of categorical variables do not convey any meaning. For example, the average of the employees' identification number does not convey any useful information. The distance between the categorical variables x and y can be measured as follows:

$$D(x, y) = (n - m)/m$$

Here, m is the number of matches between the attributes of x and y.

Binary variables The distance between two binary objects can be expressed in terms of the Jaccard coefficient. The distance between x and y binary objects is given as

$$D(x, y) = (n - m)/(n - s)$$

Here m is the number of features present in both the images x and y, and s is the number of features absent in both the images.

Quantitative variables The Euclidean distance is one of the most important distance measures and can be calculated as follows:

$$\text{Distance}(O_i, O_j) = \sqrt{\left(\sum O_{ik} - O_{jk}\right)^2}$$

The advantage of the Euclidean distance is that it does not change with the addition of new objects. However, if the units change, the resultant change is enormous. Another distance that is useful is city block or Manhattan distance. This is the average distance across dimensions. It dampens the effects of the outliers. Manhattan distance as an average across dimensions can be expressed as

$$\text{Manhattan average distance } (O_i, O_j) = \frac{1}{n}\sum_{k=1}^{n}\left|O_{ik} - O_{jk}\right|$$

Both the distances can be combined into a single form called Minkowski distance and can be expressed for two objects O_i and O_j having attributes $(x_1, x_2, \ldots, x_n)$ and $(y_1, y_2, \ldots, y_n)$ as

$$D(x, y) = \left(\left|x_1 - y_1\right|^q + \left|x_2 - y_2\right|^q + \ldots + \left|x_n - y_n\right|^q\right)^{\frac{1}{q}}$$

When $q = 1$, $D(x, y)$ becomes Manhattan distance and when $q = 2$, $D(x, y)$ is called Euclid distance. When q becomes infinite, the distance is called Chebychev distance.

Example 13.12 Consider the following two objects O_1 and O_2 with two attributes x_1 and x_2. Find the Euclid, Manhattan average, and Chebychev distances.

Solution The calculations are as follows:

Object/Variable	x_1	x_2
O_1	5	6
O_2	2	3

$$\text{Euclid distance } D(O_1, O_2) = \sqrt{(5-2)^2 + (6-3)^2} = \sqrt{9+9} = \sqrt{18}$$

$$\text{Manhattan average distance}(O_i, O_j) = \frac{1}{n}\sum_{k=1}^{n}\left|O_{jk} - O_{jk}\right| = \frac{1}{2}(|5-2| + |6-3|)$$

$$= \frac{1}{2}(6) = 3$$

$$\text{Chebychev distance}(O_j, O_i) = \max_k(|O_{ik} - O_{jk}|)$$

$$= \max(|5-2|, |6-3|) = \max(3, 3) = 3$$

Ordinal variables Ordinal variables involve ranking among the attributes. The distance for ranked attributes can be calculated as $Z_i = (r_i - 1)/(M - 1)$; where r_i is the rank and M is the maximum rank.

Interval and ratio variables If the difference between two measurements of a variable is meaningful, it is called an interval variable. For example, the difference between 15 and 20 of a size variable is meaningful. Ratio variable is the same as interval variable, but there is no definition for the value of zero. For example, zero weight or zero height has no meaning. Hence weight and height are examples of ratio variables. Minkowski distance functions can be used for these variables.

13.10.2 Categories of Clustering Algorithms

Broadly speaking, clustering algorithms can be classified into two categories—hierarchical methods and partitional methods.

13.10.2.1 Hierarchical clustering methods

Hierarchical methods produce a recursive partition of the set of objects and the results are shown visually as a dendrogram. These methods are divided into two categories—agglomerative methods and divisive methods.

Agglomerative algorithms treat each individual object as a cluster. They are then merged with other clusters and this process is continued to ultimately get a single cluster. Divisive methods, on the other hand, take whole objects as a single cluster and partition the cluster. This process is continued till the cluster results in smaller clusters.

Agglomerative methods employ the following procedure:

1. Create a separate cluster for every data instance.
2. Repeat the following steps till a single cluster is obtained:

   (a) Determine the two most similar clusters using similarity measures.
   (b) Merge the two clusters into a single cluster.
3. Choose a cluster formed by one of the step 2 results as final, if no more merging is possible.

The advantages of hierarchical methods include the fact that there is no need for vector representation for each object. These algorithms are easy to understand and interpret, and are intuitive and simple. Some of the popular algorithms are as follows:

Single-linkage algorithm A single-linkage algorithm is an agglomerative algorithm that takes a single instance and merges it with a cluster with which it is closer This process is continued till no more merging is possible. Consider the sample image shown in Fig. 13.14(a).

The Euclidean distance can be calculated between the pixel values for deciding the merging process. The Euclidean distance is shown in Fig. 13.14(b).

The minimum is 4.0, which is the distance between 2 and 4. Therefore, these two instances are merged in the next step as shown in Fig. 13.14(c).

Pixel	X	Y
1	4	4
2	8	3
3	7	8
4	12	3

(a)

	1	2	3	4
1	—	4.1	5	8.1
2	4.1	—	5.1	**4.0**
3	5	5.1	—	7.1
4	8.1	**4.0**	7.1	—

(b)

	{2, 4}	1	3
{2, 4}	—	4.1	5.1
1	4.1	—	5
3	5.1	5	—

(c)

Fig. 13.14 Single-linkage algorithm (a) Sample image (b) Euclid distance (c) First iteration

Distance $(P + Q, R) = \min(\text{Distance}(P, R), \text{Distance}(Q, R))$. Accordingly, the distance between {1} and {2, 4} is the minimum of the distance between {1, 2} and {1, 4}. Here the minimum is 4.1. Therefore, the resulting cluster is {1, 2, 4} and {3}. There is no point in performing the next iteration as it results in the merging of all the instances. The process therefore ends with two clusters. The results of this process can be shown visually as a dendrogram (Fig. 13.15).

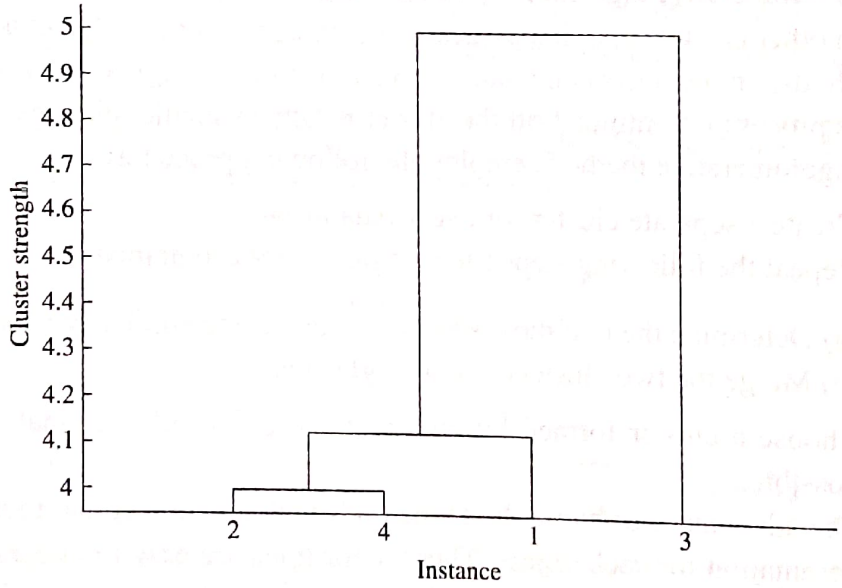


Fig. 13.15 Dendrogram for single-linkage algorithm

Complete-linkage algorithm This algorithm is similar to the single-linkage algorithm in most aspects. However, the difference lies in the calculation of the distance between the instance and the cluster. In complete-linkage algorithm, the distance is $\text{Distance}(P + Q, R) = \max\{\text{Distance}(P, Q), \text{Distance}(Q, R)\}$. The data in Fig. 13.14(a) is taken as input and the complete linkage algorithm is applied. The initial Euclidean distance is shown in Fig. 13.16(a).

The minimum is 4.0, which is the distance between 2 and 4. Therefore, these two instances are merged in the next step and are shown in Fig. 13.16(b).

	1	2	3	4
1	—	4.1	5	8.1
2	4.1	—	5.1	**4.0**
3	5	5.1	—	7.1
4	8.1	**4.0**	7.1	

(a)

	{2, 4}	1	3
{2, 4}	—	8.1	7.1
1	8.1	—	5
3	7.1	5	—

(b)

Fig. 13.16 Complete-linkage Algorithm (a) Euclidean distance table (b) First iteration

The distance$(P + Q, R)$ can be calculated as max(Distance(P, R), Distance(Q, R)). Accordingly, the distance between $\{1\}$ and $\{2, 4\}$ is the maximum of the distance between $\{1, 2\}$ and $\{1, 4\}$. Here, the maximum is 8.1. Therefore, the resulting cluster is $\{1, 3\}$ and $\{2, 4\}$. There is no point in performing the next iteration as it results in the merging of all the instances. Therefore, the process ends with two clusters. The results of this process can be shown visually as a dendrogram (Fig. 13.17).

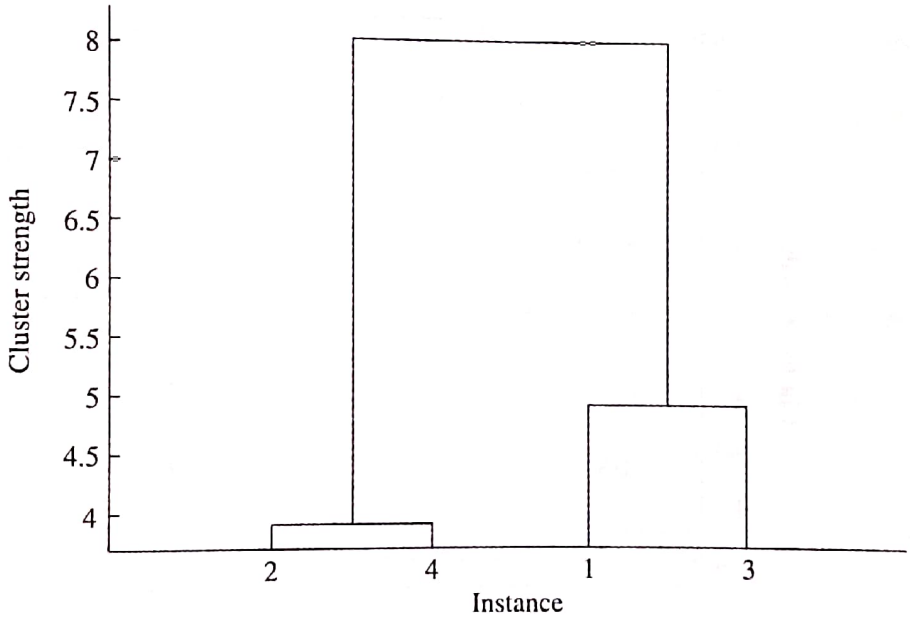


Fig. 13.17 Dendrogram for complete-linkage algorithm

Average-linking algorithm The average-linking algorithm is also similar to single-linkage and complete-linkage algorithms. However, the distance between the instance and the cluster is calculated as the average of the individual distances, that is, the distance is

$$D_{\text{average-linkage}} = \frac{1}{n_i n_j} \sum_{a \in C_i, b \in C_j} d(a,b)$$

where $d(a, b)$ is the distance between objects a and b ($a \in C_i$, $b \in C_j$), n_i and n_j are the number of objects in the clusters C_i and C_j, respectively.

For the original data in Fig. 13.14(a), the initial Euclid distance is shown in Fig. 13.18(a). The minimum is 4.0, which is the distance between 2 and 4. These two instances are merged in the next step as shown in Fig. 13.18(b).

	1	2	3	4
1	—	4.1	5	8.1
2	4.1	—	5.1	**4.0**
3	5	5.1	—	7.1
4	8.1	**4.0**	7.1	—

(a)

	{2, 4}	1	3
{2, 4}	—	6.1	6.1
1	6.1	—	5
3	6.1	5	—

(b)

Fig. 13.18 Average-linking algorithm (a) Euclid distance table (b) First iteration

The distance between {1} and {2, 4} is the average of Distance(P, R) and Distance (Q, R). The results of this process can be shown visually as a dendrogram (Fig. 13.19).

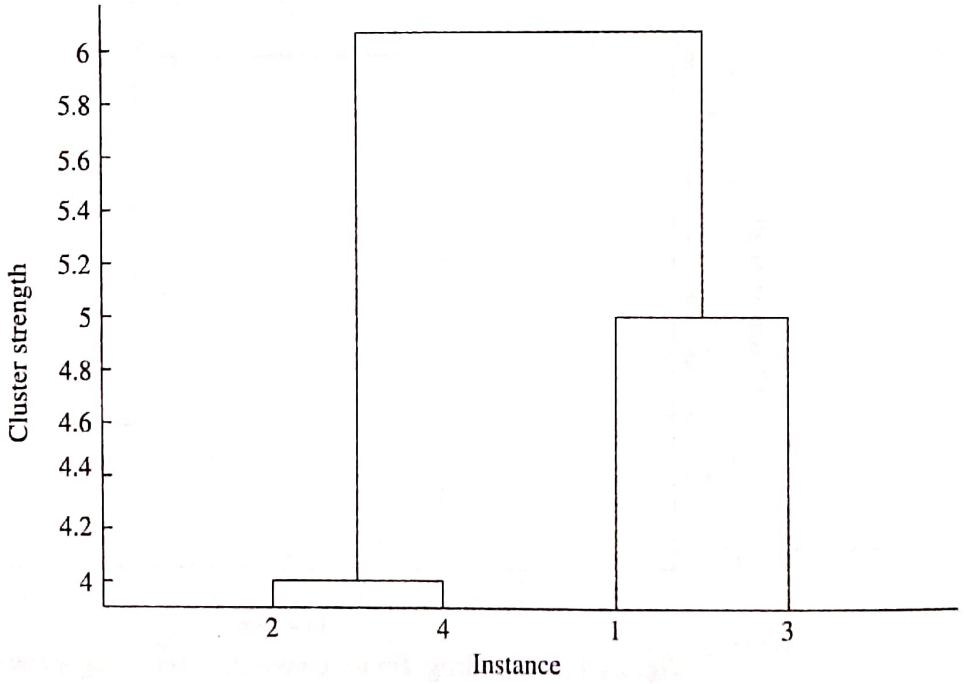


Fig. 13.19 Dendrogram for average-linkage algorithm

The advantages of the hierarchical algorithms are that they normally yield the correct number of clusters, are helpful in identifying the outliers, and are easy to understand.

13.10.2.2 Partitional methods

Partitional methods are 'greedy' procedures that are used iteratively to obtain a single level of partition. Being based on the 'greedy approach', these produce locally optimal or suboptimal solutions.

k-means algorithm is a partitional algorithm. The user has to specify the number of clusters initially. Let it be k. Then the algorithm randomly creates k cluster centres. It then assigns each point to the k clusters if the distance between the point and the cluster mean is minimum. The algorithm recomputes the centroid of the cluster as soon as the instance is added to it. The updated centroid for k clusters is used for the next iteration.

This process is continued till no further change of instances to clusters is noticed. This algorithm is summarized as follows:

1. Obtain the expected number of clusters k from the user.
2. Based on the value, choose k instances of the training image set randomly. These are initial cluster centres.
3. Assign the instances to the closest cluster based on the similarity measure. As soon as the instance is added, compute the new mean.
4. Perform the iterations till there is no change in the centroid value. Otherwise, choose a new mean and go to step 2.

Example 13.13 Let us assume that the pixels of an image are {(1, 1),(3, 3),(7, 8), (9, 9)}. Apply k-means algorithm. Let $k = 2$ and assume that the initial seeds are (1, 1) and (9, 9).

Solution Let the cluster 1 be (1, 1) and cluster 2 be (9, 9).

Now {3, 3} needs to be assigned to either cluster 1 or 2. So find the Euclidean distances between the pixel values.

The Euclidean distance between {1, 1} and {3, 3} is 2 and that between (1, 1) and (9, 9) is $8\sqrt{2}$. So {3, 3} is assigned to cluster 1. Now the new centroid is calculated as {(1 + 3/2), (1 + 3/2)} = {2, 2}.

Now {7, 8} needs to be assigned to either cluster 1 or 2. So find the Euclidean distance between (7, 8) and the new centroid, that is, {7, 8} and {2, 2} which is 7.81. The distance between (7, 8) and (9, 9) is 2.2. So (7, 8) is added to the cluster (9, 9). Now the new centroid is calculated as {(7 + 9/2), (8 + 9/2)} = {8, 8.5}.

Now again consider (3, 3) with the new centroid (8, 8.5). The distance is 7.4, which is larger then the distance between (3, 3) and the cluster centroid of cluster 1.

So the resulting clusters are {(1, 1), (3, 3)} and {(7, 8), (9, 9)}.

Example 13.14 Consider the following points:

(1, 1) (1, 3) (2, 4) (4, 8) (12, 8)

Cluster it using simple linkage algorithm.

Solution Compute the distance between the following points:

Points	*x*	*y*
1	1	1
2	1	3
3	2	4
4	4	8
5	12	8

$$(1) \text{ and } (2) = \sqrt{(1-1)^2 + (1-3)^2} = \sqrt{(0+4)} = \sqrt{4} = 2$$

$$(1) \text{ and } (3) = \sqrt{(1-2)^2 + (1-4)^2} = \sqrt{(1+9)} = \sqrt{10} = 3.11$$

$$(1) \text{ and } (4) = \sqrt{(1-4)^2 + (1-8)^2} = \sqrt{(9+49)} = \sqrt{58} = 7.62$$

$$(1) \text{ and } (5) = \sqrt{(1-12)^2 + (1-8)^2} = \sqrt{(121+49)} = \sqrt{170} = 13.04$$

$$(2) \text{ and } (3) = \sqrt{(1-2)^2 + (3-4)^2} = \sqrt{(1+1)} = \sqrt{2} = 1.414$$

$$(2) \text{ and } (4) = \sqrt{(1-4)^2 + (3-8)^2} = \sqrt{(9+25)} = \sqrt{34} = 5.83$$

$$(2) \text{ and } (5) = \sqrt{(1-12)^2 + (3-8)^2} = \sqrt{(121+25)} = \sqrt{146} = 12.08$$

$$(3) \text{ and } (4) = \sqrt{(2-4)^2 + (4-8)^2} = \sqrt{(4+16)} = \sqrt{20} = 4.472$$

$$(3) \text{ and } (5) = \sqrt{(2-12)^2 + (4-8)^2} = \sqrt{(100+16)} = \sqrt{116} = 10.77$$

$$(4) \text{ and } (5) = \sqrt{(4-12)^2 + (8-8)^2} = \sqrt{(64+0)} = \sqrt{64} = 8$$

After the computation, this can be plotted as a table

	1	2	3	4	5
1	–	2	3.11	7.62	13.04
2		–	**1.414**	5.83	12.08
3			–	4.472	10.77
4				–	8
5					–

The minimum distance is 1.414. Therefore, points 2 and 3 are grouped together. This results in the following table.

	2,3	1	4	5
2.3	–	**2**	4.472	10.77
1		–	7.62	13.04
4			–	8
5				–

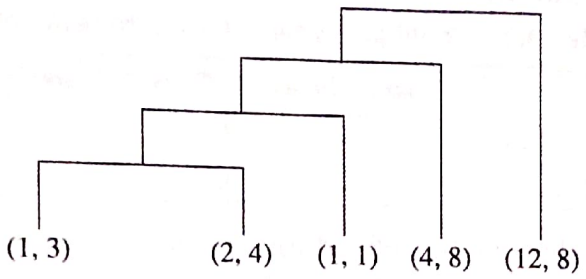
The minimum distance is 2. Therefore, points 1, 2, and 3 are grouped together. This results in the following table.

	1,2,3	4	5
1,2,3	–	**4.472**	10.77
4		–	8
5			–

The minimum distance is 4.472. Therefore, points 1, 2, 3, and 4 are grouped together. This results in the following table.

	1,2,3,4	5
1,2,3,4	–	8
5		–

The resulting dendrogram is shown here:



13.10.3 Cluster Evaluation Methods

The essential conditions to be satisfied by a good clustering algorithm include robust features such as scale-invariance, ability to obtain good clusters on all attribute values/ methods, and consistency. Evaluation of clustering is difficult as no benchmark data is available as in classification, since the domain of clustering has no prior knowledge of data. However, the characteristics of a good clustering algorithm are as follows:

1. Efficiency of the clustering algorithm
2. Ability to handle missing data in the dataset
3. Ability to handle noisy and outlier data
4. Ability to handle different attribute types

A cluster is supposed to have all elements that are similar and each element should be different from the elements of other clusters. Cluster cohesion is a measure that indicates how similar the elements are to each other in a cluster, and cluster separation is a measure that indicates how distinct a cluster is from other clusters. Some of the useful metrics are the following:

1. Purity: This is a measure of the extent to which a cluster consists of elements belonging to a single class. For example, let us assume that there are three clusters—cluster 1 with ten x elements and two y elements, cluster 2 with nine x elements and one y element and cluster 3 with two x elements and ten y elements. Then purity can be measured as

$$\frac{1}{\text{Total elements}}(\text{Sum of the majority elements of all the clusters})$$

$$=\frac{1}{34}(10+9+10)=\frac{29}{34}\cong 0.85$$

Purity values range from 0–1 and is high when clusters have more coherent values.

2. Precision and recall as discussed in classifier evaluation are useful in clustering also. These measures indicate the fraction and the extent of the specific class present in a cluster.
3. Similarity-based measures are useful for cluster evaluation. The idea behind similarity oriented measures is that if elements A and B belong to a same class then they should also belong to the same cluster. A contingency table shown in Table 13.8 can be designed for cluster validation.

Table 13.8 Contingency table for cluster evaluation

	Same cluster	Different cluster
Same class	A	B
Different class	C	D

Then two metrics that can be defined for cluster evaluation are as follows:

1. Jaccard coefficient $= \dfrac{A}{B+C+D}$
2. Rand statistic $= \dfrac{A+B}{A+B+C+D}$

The value of these metrics is in the range 0–1, and a high value indicates better clustering.

SUMMARY

- Classification is a supervisory method whose purpose is to assign a label to an unknown instance.
- Regression differs from classification in that it predicts the continuous variable.
- Bayesian classifier is a probabilistic model that is very popular as well as effective and uses Bayesian principle.
- Regression analysis models the relationship between independent and dependent variables.
- Nearest neighbour techniques use distance measures to predict the class of the unknown instances.
- The classifiers can be evaluated based on the confusion matrix. Parameters such as sensitivity, specificity, and accuracy can be determined from the confusion matrix.
- Clustering uses similarity measures for clustering, and dendrogram for visualization of the results.
- Clustering algorithms are evaluated using efficiency, ability of the algorithm to handle missing/noisy data, and ability to handle different attribute types/magnitude.

KEY TERMS

Classification It is the technique of categorizing an object and assigning a label to the unknown instance.

Clustering It is the technique of partitioning a group of images into meaningful disjoint subgroups.

Decision theoretic technique It is a technique of designing a decision or discriminant function for classifying input data.

Distance functions These are metrics that quantify the similarity or dissimilarity of images.

Feature vector It is a set of features of an image and is also known as pattern vector.

Hierarchical methods These are methods that use a recursive partition of the set of objects.

Object recognition It is the technique of identifying an object and assigning it a label.

Pattern It is a group of vectors that characterize the object.

Regression It is a technique for predicting a continuous variable.

ROC It is the acronym for 'receiver operating characteristic' curve, which is used to visually evaluate the performance of the classifier.

Supervised learning It means learning of a system with the assistance of a supervisor.

Syntactic method It is a formal language technique of recognizing a string using grammar rules.

Template matching It is a technique of using image correlation to recognize the presence or absence of a target object.

Unsupervised learning It is a technique to group objects using trial and error without the assistance of a training phase.

REVIEW QUESTIONS

1. Define and distinguish the following terms—classification, regression, and clustering.
2. What are the disadvantages of template matching?
3. State Bayesian principle.
4. What are the advantages and disadvantages of Bayesian classification?
5. What is the difference between statistical and syntactic classifiers?
6. What are the advantages and disadvantages of clustering schemes?
7. What are the problems associated with clustering large data?

NUMERICAL PROBLEMS

1. Use Naïve Bayes classifier and classify the unknown pixel X in the following image, which contains two types of pixels # and *.

$$\begin{pmatrix} \# & \# & * & * \\ \# & X & \# & * \\ * & \# & * & * \\ \# & \# & * & * \end{pmatrix}$$

Consider the 4-neighbourhood of X and determine the class of X.

2. Consider the following data. Use Naïve Bayesian classifier to classify the instance (1, 1).

S. no.	X	Y	Class
1	0	0	C_1
2	1	0	C_1
3	0	1	C_2
4	1	0	C_1

3. Consider the following data:

(2 3 4) and (1 5 6)
(2 2 9) and (7 8 9)

Calculate the Euclidean and average Manhattan distances.

4. Apply linear regression to the following data.

S. no.	X	Y
1	3	5
2	7	8
3	12	5
4	16	9
5	20	8

Select initial seeds randomly.

5. Let us assume the classifier performance with respect to a dataset is as follows. Evaluate the performance of the classifier.

Expert vs classifier		Predicted class as per classifier	
		+	–
Actual class as per the expert	+	9 (TP)	0 (FN)
	–	1 (FP)	3 (TN)

6. Cluster the following data using hierarchical methods and show the dendrogram.

S. no.	X	Y
1	3	5
2	7	8
3	12	5
4	16	9
5	20	8

7. Let us assume five clusters with dominant elements spread as 3, 4, 5, 6, and 2. What is the purity of the cluster?

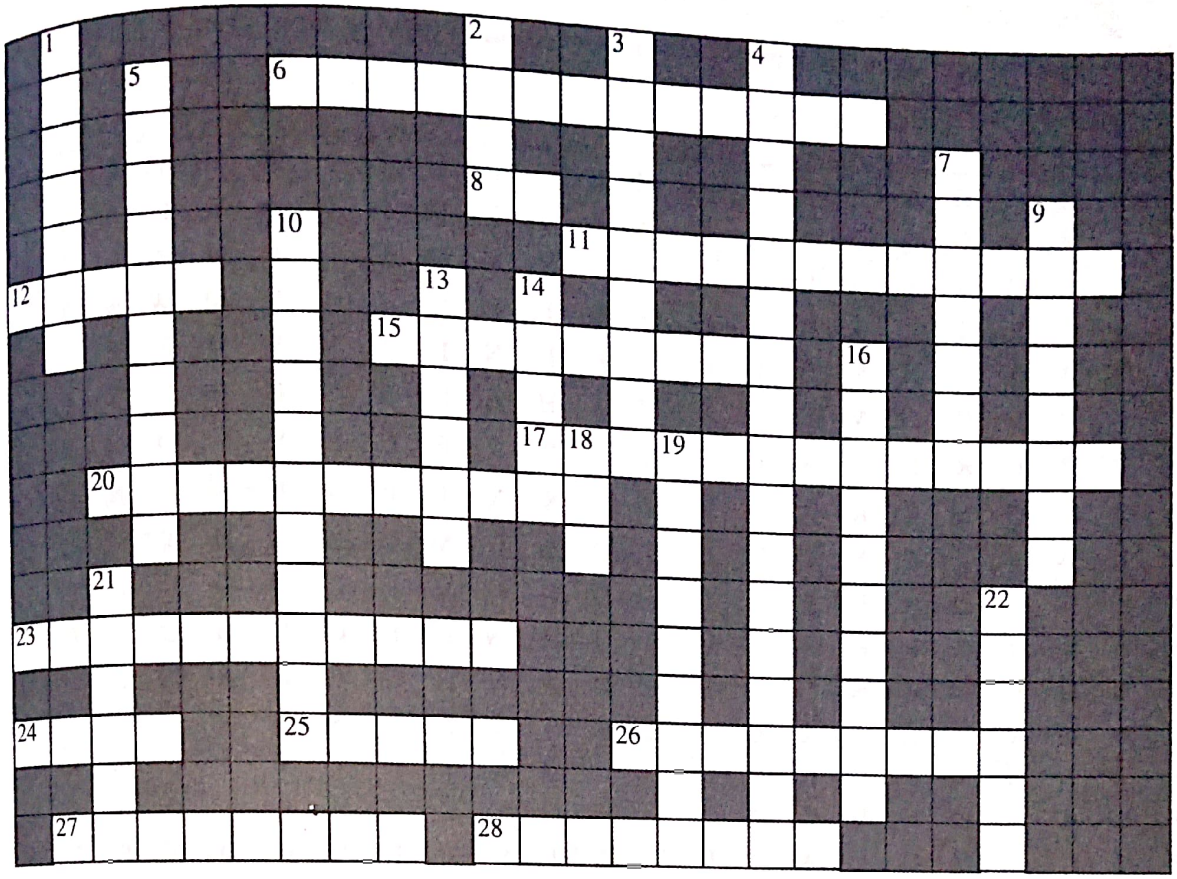
8. Consider the following data. Use k-means algorithm with $k = 2$ and show the result.

S. no.	X	Y
1	3	5
2	7	8
3	12	5
4	16	9

9. A 2×2 contingency table of a clustering algorithm is given as follows. What is the value of Jaccard coefficient and Rand statistic?

	Same cluster	Different cluster
Same class	5	3
Different class	3	5

CROSSWORD



Across

6. A single-linkage algorithm is _____ algorithm.
8. The factor TP/P is called _____ rate.
11. LDA uses _____ functions to classify the data.
12. Classification means assigning a _____ to the object.
15. Regression is used for _____ prediction.
17. KNN is a _____ (parametric/nonparametric) classifer.
20. Object similarity can be found using _____ (correlation/convolution) technique.
23. Shape number is _____ data.
24. Cost function is also known as _____ function.
25. Iris flower dataset is a _____ class problem.
26. Two graphs are isomorphic if there is a _____ function between them.
27. The phases of a classifier design are _____ and testing.
28. KNN uses _____ vote technique to find class.

Down

1. ID number is _____ variable.
2. Minimum risk classifer assigns a _____function to the classification process.
3. F1 score is a combination of _____ and Recall.
4. BLOB is _____.
5. In a numerical prediction, one should use _____ (regression/classification).
7. The resultant of the learning process is called _____ or model.
9. If the strings are same, then the value of degree of similarity is _____.
10. Naïve Bayesian classifier assumes that the features are _____ of each other.
13. The extent of cluster elements belonging to a single cluster is indicated by _____.
14. Can nature of data affect classifier design? (Yes/No)
16. Rand metric is a _____ based measure for cluster evaluation.
18. The Euclidean distance between the given classifier and the best classifier is _____ for best classifier.
19. Bayes theorem finds _____ probability using prior probability.
21. Syntactic methods use _____ to decompose the objects into a set of primitives.
22. Partitional methods use _____ method as a design technique.

WORD SEARCH PUZZLE

Some of the important terms in this chapter are present in the following word jumble. Identify the words. Diagonal words are possible.

T	M	V	H	N	O	N	P	A	R	A	M	E	T	R	I	C	I	N	O	M	I	N	A	L
C	L	I	R	W	U	Q	G	R	E	E	D	Y	S	I	M	I	L	A	R	I	T	Y	I	I
C	O	S	T	D	S	J	A	L	A	A	Y	E	S	N	O	M	A	J	O	R	I	T	Y	I
B	I	N	A	R	Y	L	A	R	G	E	O	B	J	E	C	T	N	V	M	W	M	Y	U	B
T	Q	W	B	S	T	H	R	E	E	I	N	F	I	N	I	T	Y	W	Q	F	R	T	L	Y
V	B	Y	D	I	C	D	M	Z	R	Z	M	A	H	A	L	A	N	O	B	I	S	K	K	R
E	V	B	P	U	R	I	T	Y	E	X	A	F	P	R	E	C	I	S	I	O	N	K	Q	S
A	G	G	L	O	M	E	R	A	T	I	V	E	S	X	T	R	A	I	N	I	N	G	Q	U
P	E	Z	W	V	Z	G	Y	C	D	I	S	C	R	I	M	I	N	A	N	T	L	D	R	P
O	T	F	D	O	G	R	K	B	I	J	E	C	T	I	V	E	I	O	Q	E	Z	W	L	E
S	M	P	A	R	S	E	R	K	F	O	S	Q	K	W	F	E	C	O	N	C	E	P	T	R
T	I	N	D	E	P	E	N	D	A	N	T	D	E	N	D	O	G	R	A	M	U	U	D	V
E	H	M	O	R	I	K	J	R	E	G	R	E	S	S	I	O	N	H	N	A	D	E	I	I
R	T	W	J	N	M	C	G	O	Y	N	L	C	O	R	R	E	L	A	T	I	O	N	O	S
I	T	U	X	K	E	G	W	D	Z	J	R	E	C	E	I	V	E	R	L	O	S	S	L	E
O	P	C	I	L	H	Q	U	A	L	I	T	A	T	I	V	E	D	O	O	N	E	L	I	D
R	D	W	D	J	J	V	N	U	M	E	R	I	C	A	L	Z	E	E	B	L	A	B	E	L

Hints

1. Classification assigns a _____ to unknown objects.
2. BLOB is an abbreviation of _____.
3. _____ learning is used in classification process.
4. _____ is a statistical operation used to find similar features.
5. Classification process generates a _____ so that unknown instances can be classified.
6. _____ and testing phases are essential for classification process.
7. Decision or _____ functions are used to classify objects.
8. Bayes theorem finds _____ probability using prior probability.
9. _____ distance measure is used by Bayesian minimum distance classifier.
10. Risk-based classifier assigns _____ for decisions to minimize risk.
11. KNN takes _____ class of the neighbours for finding class of the unknown instance.
12. _____ process is used for numerical predictions.
13. _____ mapping is used for graph-based approaches.
14. _____ is the probability that the object is classified correctly.
15. Shape number is a _____ variable.
16. Identification number is a _____ variable.
17. _____ is used to show clustering visually.
18. _____ algorithms treat every object as a cluster.
19. Hierarchical algorithms use _____ approach for constructing dendrograms.
20. The _____ of a cluster is an indicator of the quality of the clustering algorithm.
21. Clustering algorithms use _____ and dissimilarity for clustering of data.