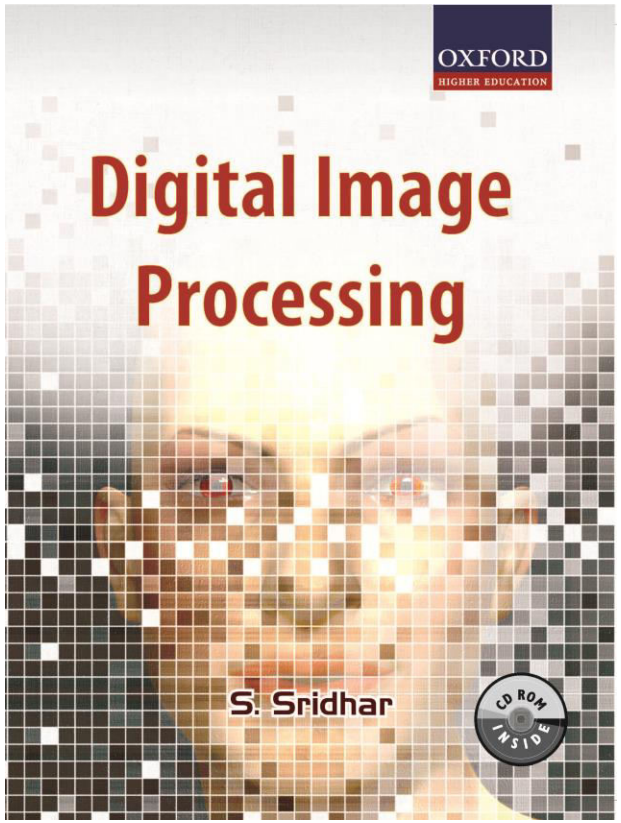
OXFORD
HIGHER EDUCATION



Digital Image Processing

S. Sridhar, Associate Professor,
Department of Information Science and Technology, College of Engineering Guindy Campus, Anna University, Chennai

OXFORD
HIGHER EDUCATION

Chapter 11
Object Recognition

PATTERNS AND PATTERN CLASSES

- The process of assigning a meaningful label to an unknown object is known as *recognition*.

- This process of comparing an unknown object with stored patterns to recognize the unknown object is called *classification*.

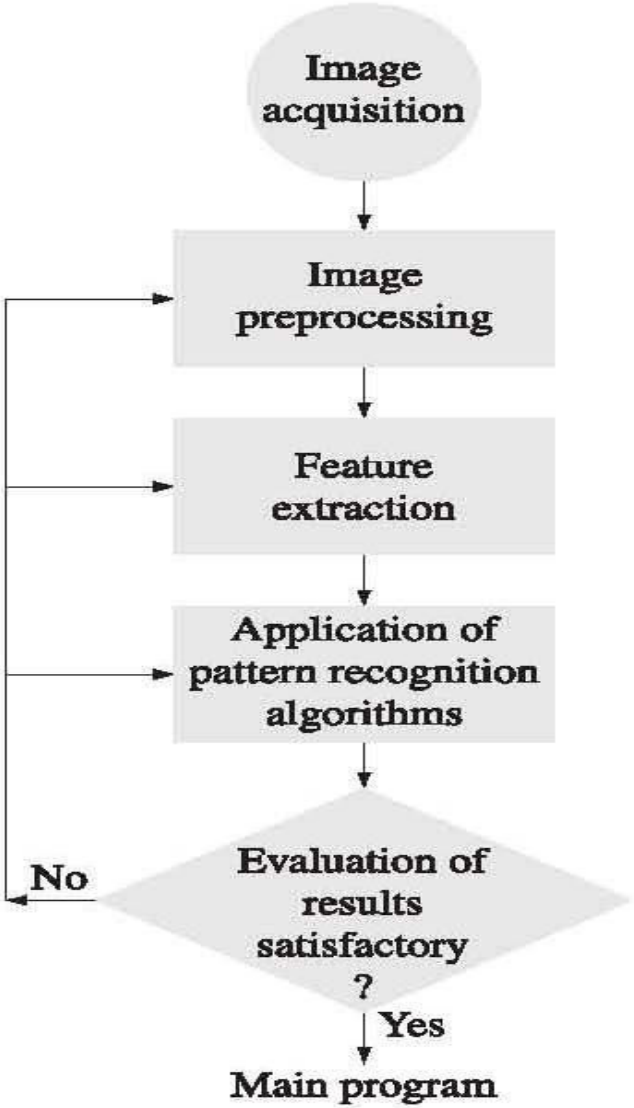


Fig. 11.1 Typical object recognition process

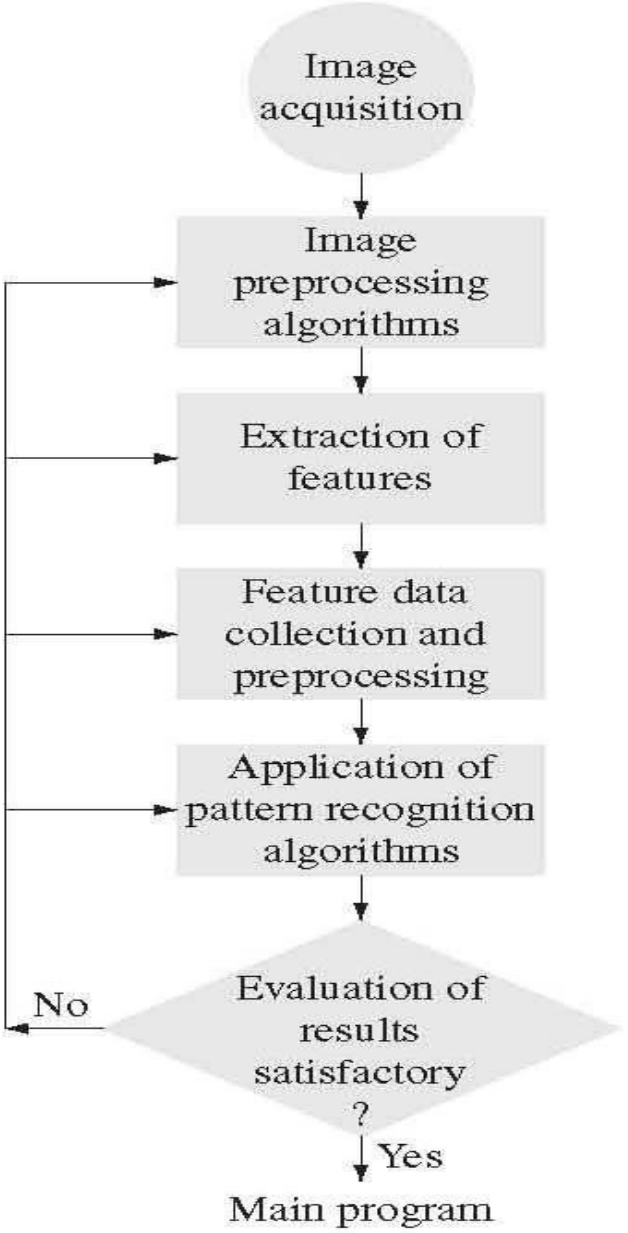


Fig. 11.2 Pattern recognition design cycle

Feature Vector

A feature vector is typically of the form $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}$ or $x = [x_1, x_2, \ldots, x_{n-1}, x_n]^T$

Supervised Learning

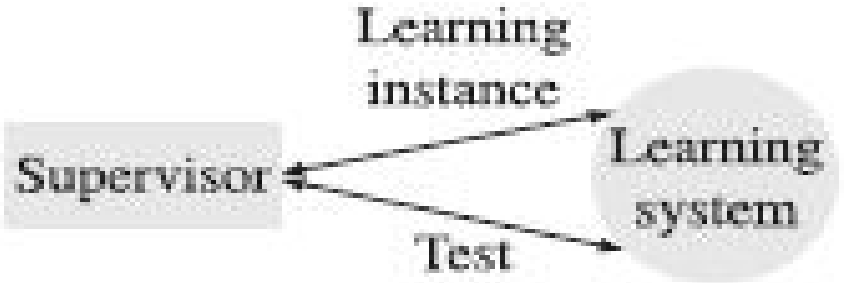


Fig. 11.3 Supervisory system

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.

Stages of Pattern Recognition Design Cycle

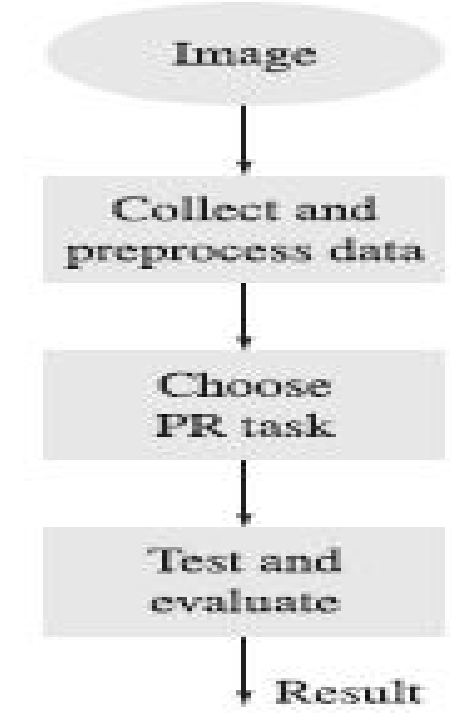


Fig. 11.4 Pattern recognition stages

TEMPLATE MATCHING

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.

Template Matching

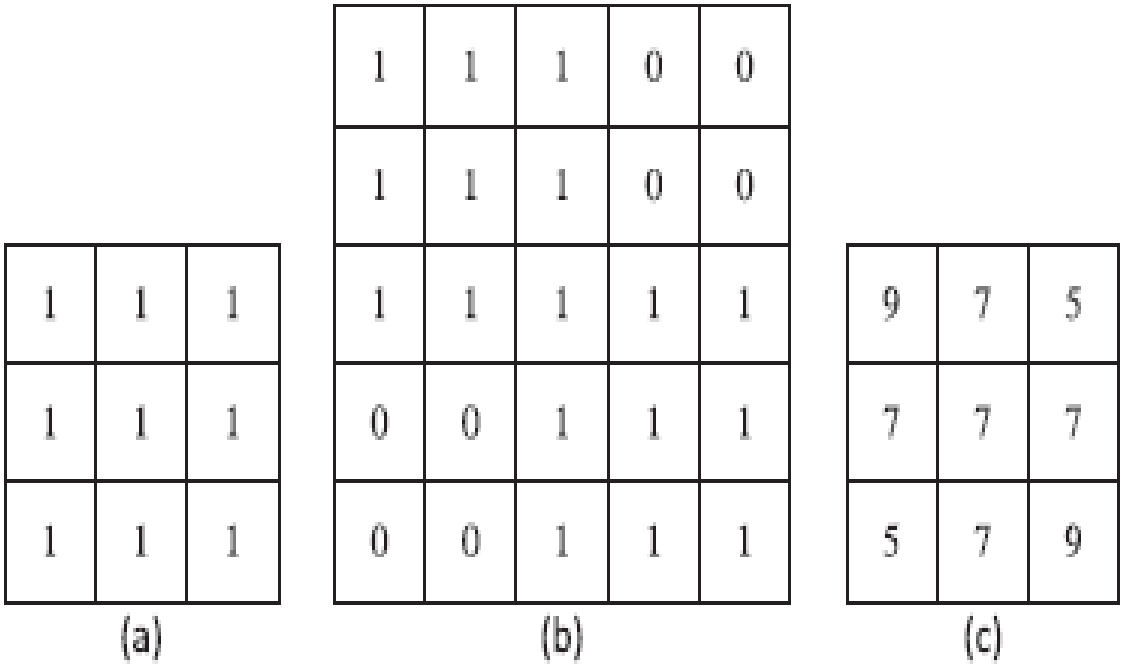


Fig. 11.5 Template matching (a) Template (b) Image array (c) Resultant correlation array

INTRODUCTION TO CLASSIFICATION

- 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.

Sample classification scheme

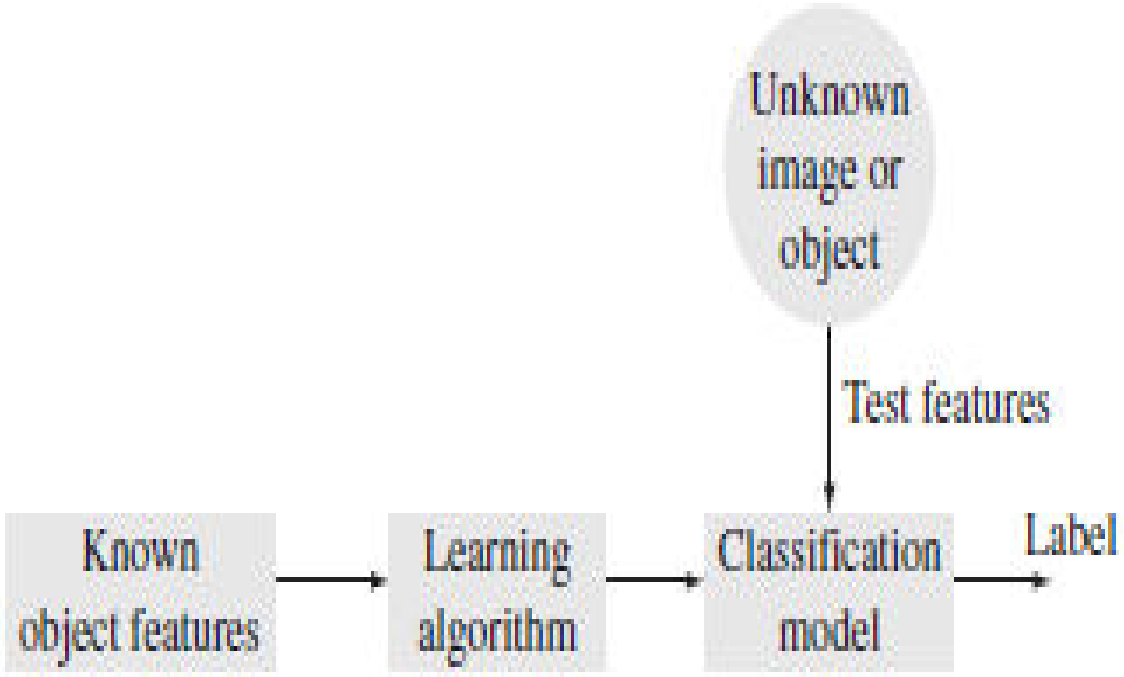


Fig. 11.6 Sample classification scheme

General classification schemes

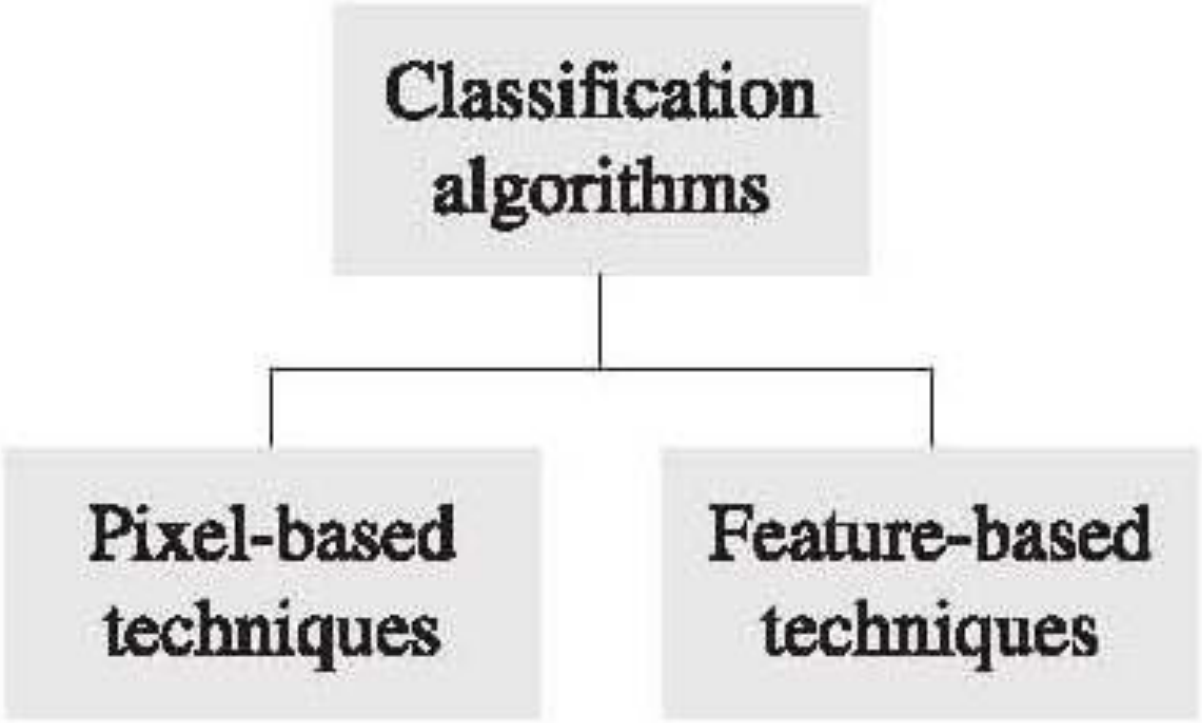


Fig. 11.7 General classification schemes

Classifier Design

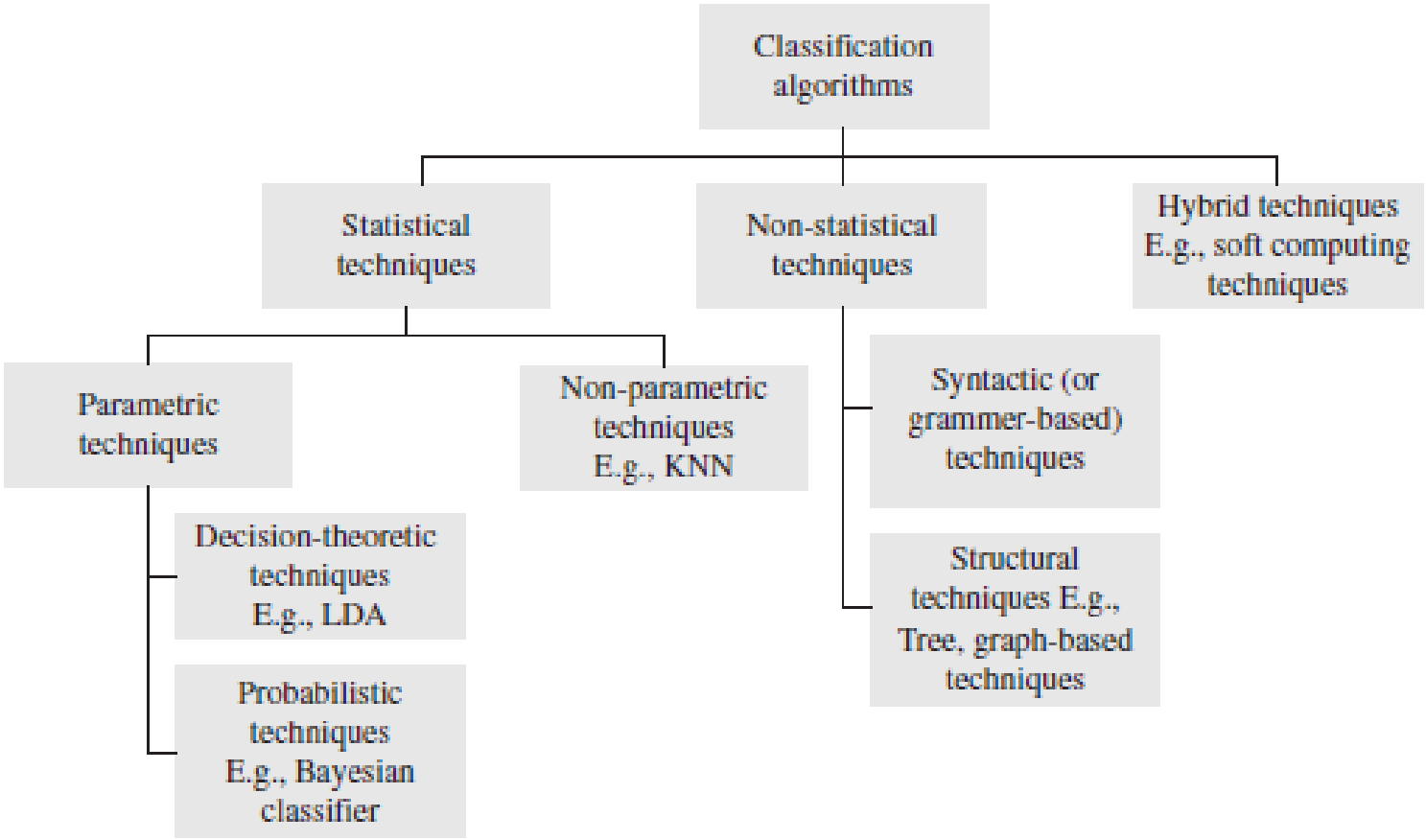


Fig. 11.8 Types of classification algorithms

Decision-theoretic Methods

- The decision-theoretic approach is often called discriminant function analysis.

One of the oldest techniques used in the decision-theoretic approach in pattern recognition is the linear discriminant analysis (LDA)

Linear Discriminant Analysis

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

Then the decision boundary is given as

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

A decision rule can be designed as follows:

Assign the instance to the class i if $d_{ij} > 0$ and assign the instance to 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.

Probabilistic Methods

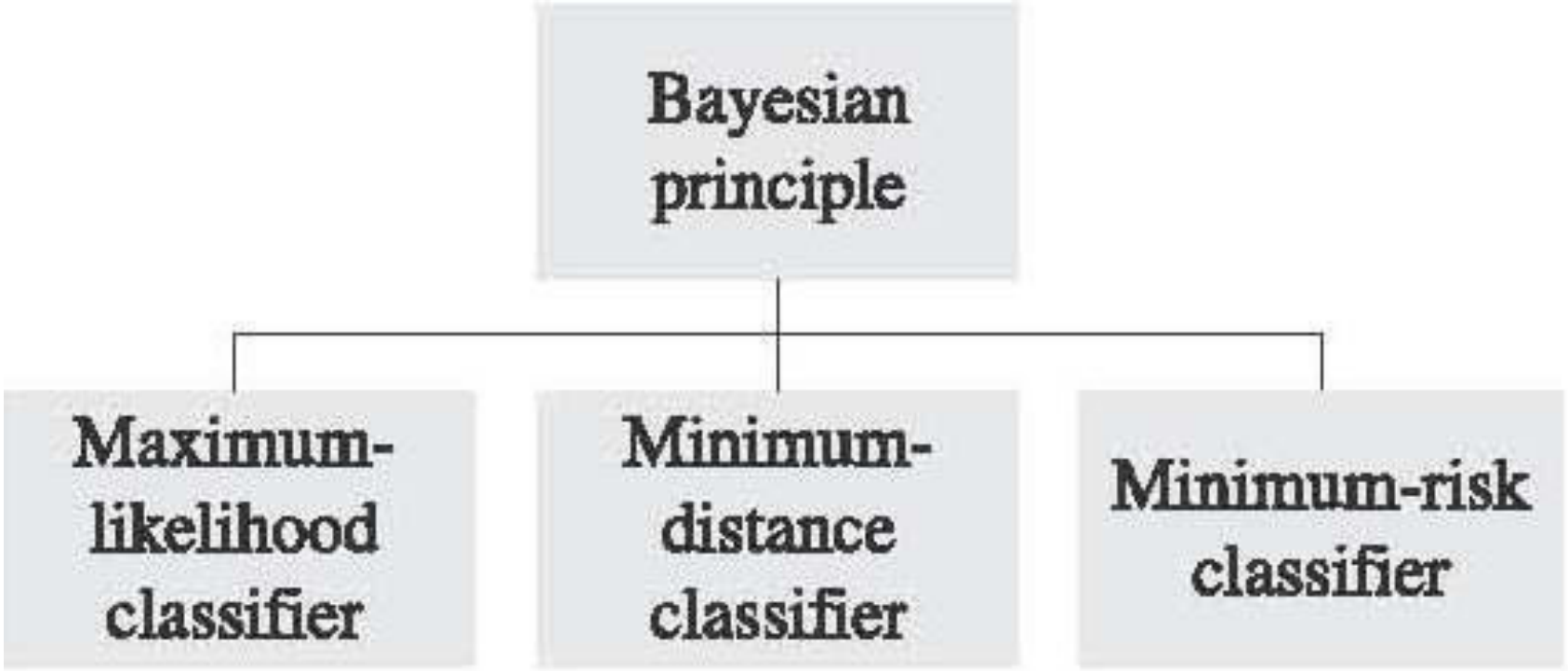


Fig. 11.9 Types of Bayesian classifiers

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

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

Algorithm

- 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.

Example 11.3 Let us assume a simple dataset, as shown in Table 11.3. Let us apply the Bayesian classifier to predict (2, 2).

Table 11.3 Sample dataset

a_1	a_2	Class
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 $P(c_2) = 2/5$ and $P(c_1) = 3/5$. The conditional probability is estimated.

$$P(a_1 = 2/c_1) = 1/2;\ P(a_2 = 2/c_2) = 1/3$$

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

Therefore,

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

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

$$P(x/c_2) = P(a_1 = 2/c_2) \times P(a_2 = 2/c_1)$$

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

This is used to evaluate

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

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

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

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

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

Parametric Bayesian classifier with Gaussian assumption and minimum distance classifier

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

Here m_i and σ_i are the mean and standard deviation of the class i. Since the data is multidimensional, the mean becomes a covariance matrix $\sum_i$. Therefore, the resultant formula is

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

Use the Euclidean distance to compute the distance between the unknown instance x and the mean vector. This is equal to

$$d_i = \|x - m_i\|$$

The norm can be defined as $\|a\| = (a^T a)^{\frac{1}{2}}$. Therefore the norm of d_i is

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

The problem now becomes assigning an instance either to class i or class j. Similarly, the distance function for class i can be calculated as

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

Hence the decision boundary between classes i and j can be calculated as $d_i(x) - d_j(x)$.

This is equivalent to

$$x^{\mathrm{T}} m_i - \frac{1}{2} m_i^{\mathrm{T}} m_i - x^{\mathrm{T}} m_j + \frac{1}{2} m_j^{\mathrm{T}} m_j$$

$$x^{\mathrm{T}} (m_i - m_j) - \frac{1}{2} (m_i - m_j)^{\mathrm{T}} (m_i + m_j) = 0$$

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

Multiple Features

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

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

If this data $P(x)$ is assumed to be a Gaussian distribution, it is given by

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

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

Minimum Risk Classifier

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.

Non-parametric Statistical Methods

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 each class.
3. Classify the test tuple to the appropriate class.

Regression Methods

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,

$$W_1 = \sum_{i=1}^{|D|} (x_i - \overline{x})(y_i - \overline{y}) / \sum_{i=1}^{|D|} (x_i - \overline{x})^2$$

$$W_0 = \overline{y} - \overline{w}_1 \overline{x}$$

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

STRUCTURAL AND SYNTACTIC CLASSIFIER ALGORITHMS

- 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 grammar in formal languages is defined as a 4-tuple structure, $G = \{T, NT, S, P\}$, where T is a set of terminal symbols similar to the alphabets of English or constants of a programming language. NT 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 defines 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, $L(G)$.

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.

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.

Contd…

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.

Rule-based Algorithms

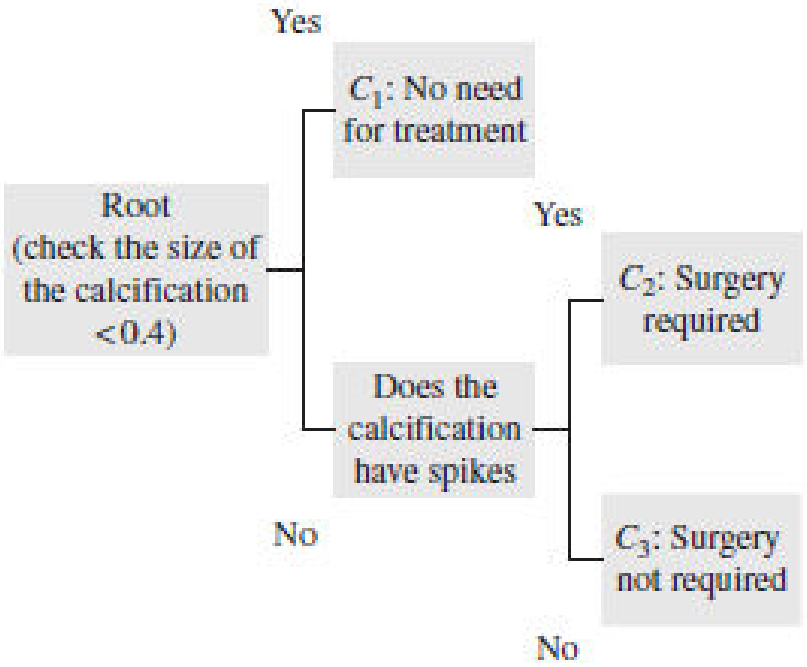


Fig. 11.11 Sample tree classifier

Graph-based Approach

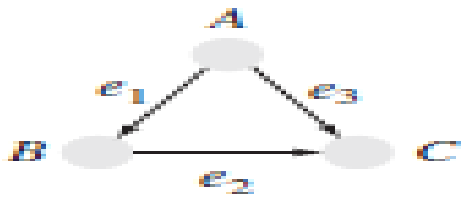
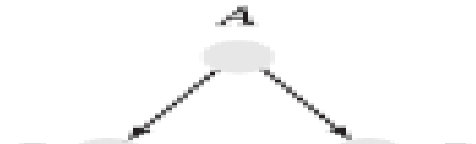


Fig. 11.12 Sample graph



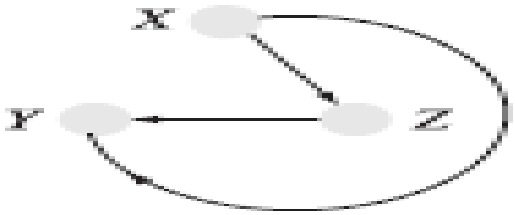


Fig. 11.13 Two similar but structurally different graphs

Algorithm

- 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.

EVALUATION OF CLASSIFIER ALGORITHMS

- **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.

k-fold cross validation

- ***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.

Leave-one-out 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.

- ***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.

True positive rate (TP rate) This metric is also referred 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 (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 (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 (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

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

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. 11.15(a).

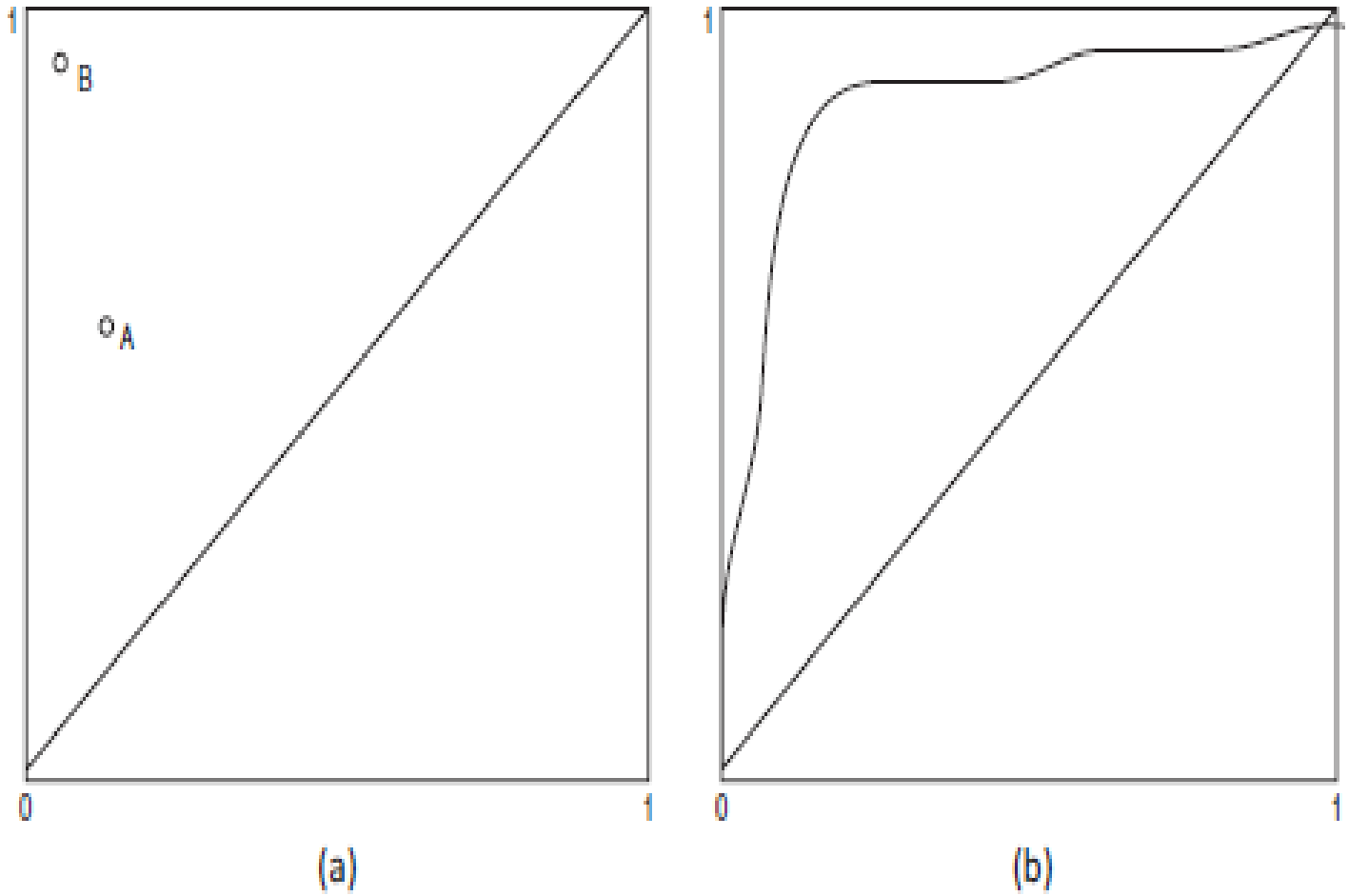


Fig. 11.15 ROC curve (a) Sample graph (diagonal line indicates the performance of the average classifier) (b) Sample ROC curve

BIOMETRICS CASE STUDIES

- An independent biometric system has the following modules/stages:
- 1. Sensor module
- 2. Feature extraction module
- 3. Pattern matching and decision module
- 4. System database module

Types of biometrics

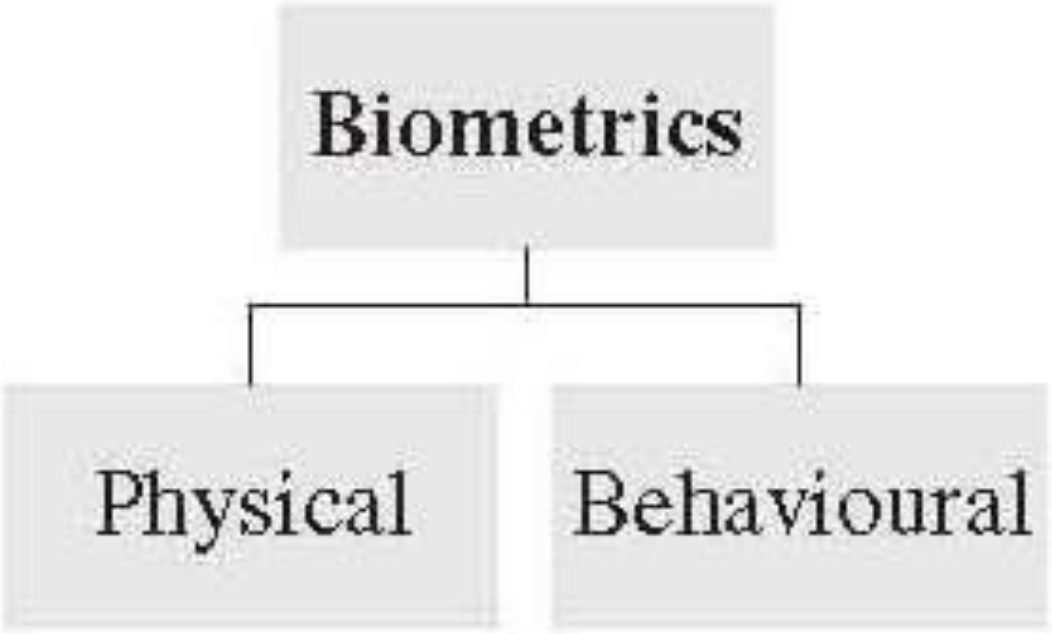


Fig. 11.16 Types of biometrics

Face recognition

- 1. Read the image.
- 2. Calculate the covariance matrix.
- 3. Apply PCA. PCA converts a set of faces of the training set to a single vector. This is described as $y_i = w^t x_i$
- Here *xi* is the set of input faces, *w* is the weight, and *yi* is the output. The weight factors are eigen vectors of the covariance matrix of the training set. Complexity is rec$y_i = w^t x_i$s PCA chooses very few principal components. The output *yi* represents the entire training set. This is called eigen image.
- 4. Check the unknown target face image by projecting it onto an eigen space and comparing its position with the eigen face generated using the training set. If it is possible to match the position, declare the target face as valid.

Iris Recognition

1. Image acquisition
2. Iris localization
3. Pattern recognition

The first step involves the use of a video camera for capturing the image of the retina. The image should be a high quality image capturing the complete details of the iris.

In the second stage, active contour models, which have been discussed in Chapter 7, are used to segment the iris. In the localization process, the surrounding eyelashes are removed using histogram analysis and statistical inference.

Iris recognition is used in 1:M (one to many) identification mode. The processed iris code has at least 200 important points. In the third stage, the test iris is validated with the library of stored patterns by the matching process. The iris matching process involves the Hamming code distance and the result is normalized to prevent false positives.

Image acquisition (especially from non-cooperating people) and the partial occlusion of the iris due to eyelashes and eye glasses are some of the major issues in iris recognition.

Fingerprint Recognition

Fig. 11.17 Fingerprint features

Signature Verification

A standard offline signature verification system consists of four steps:

The first step is image acquisition. In this step, the signatures are obtained. This is easier now as many devices such as pen tablets and touch screens are available. The signatures are then converted to binary form.

In the second step, the binary image is thinned using morphological operators, which have been described in Chapter 9. The thinning operation extracts the medial axis.

The third step involves the extraction of structural features. Watershed-based recognition approach uses watershed segmentation for extracting the signature, and features such as filling time (the time taken to fill a set of connected edges), loop count (the number of simple loops), and water amount (in terms of pixels) can be obtained.

Additionally, the techniques discussed in Chapter 10, such as fitting the signature pattern with a rectangular bounding box and determining the width and height of the bounding box, can be used. The lines can be detected by using the Hough transform.

The final stage is a signature matching scenario where the test signature is matched with the patterns stored in the database. The minimum distance classifier may be used to identify the validity of the signature.

CLUSTERING TECHNIQUES

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.

Distance Measures

Table 11.7 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

Hierarchical clustering methods

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.

Clustering Algorithms

Pixel	X	Y
s1	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. 11.18 Single-linkage algorithm (a) Sample image (b) Euclid distance (c) First iteration

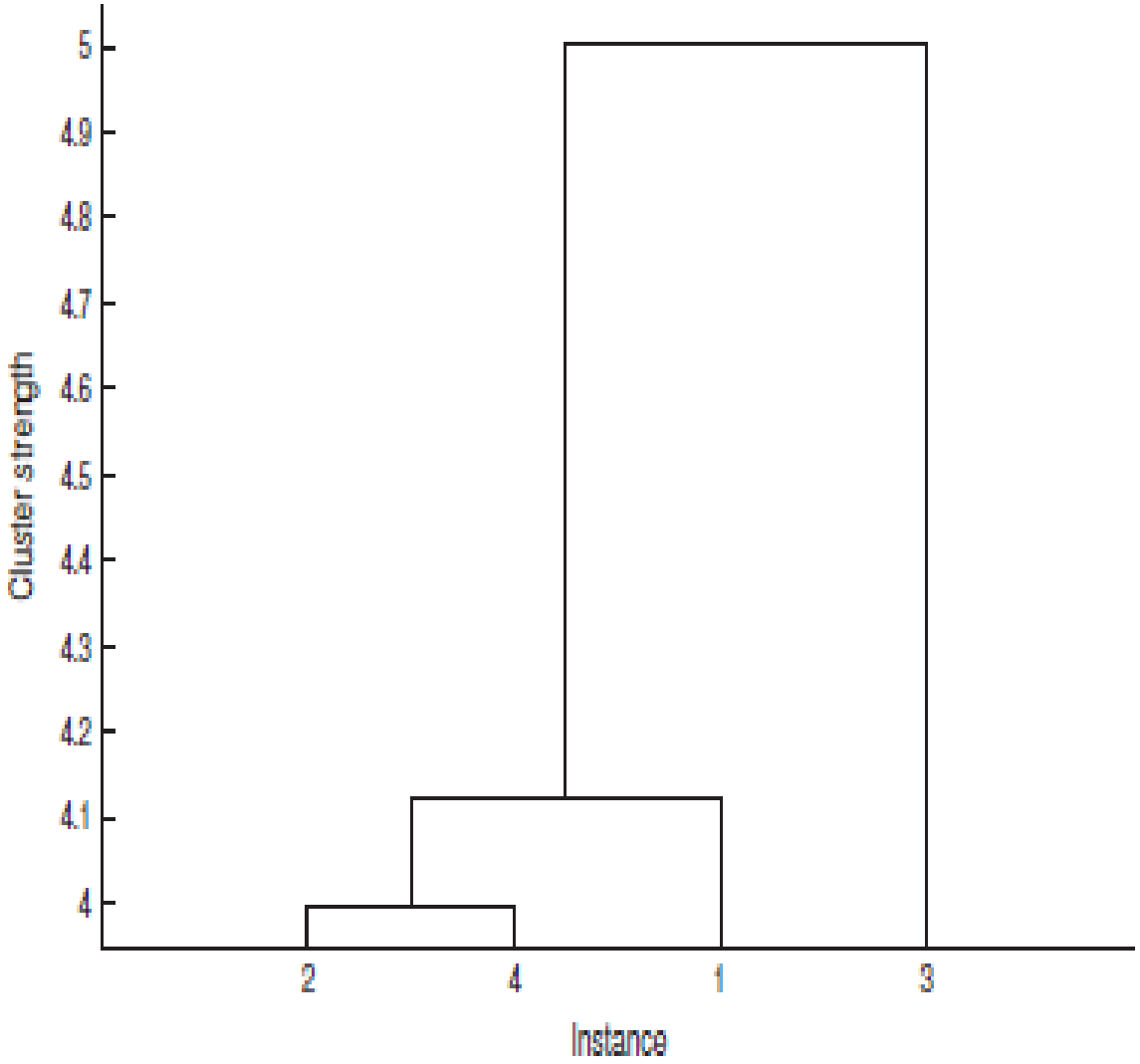


Fig. 11.19 Dendrogram for single-linkage algorithm

Complete-linkage algorithm

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

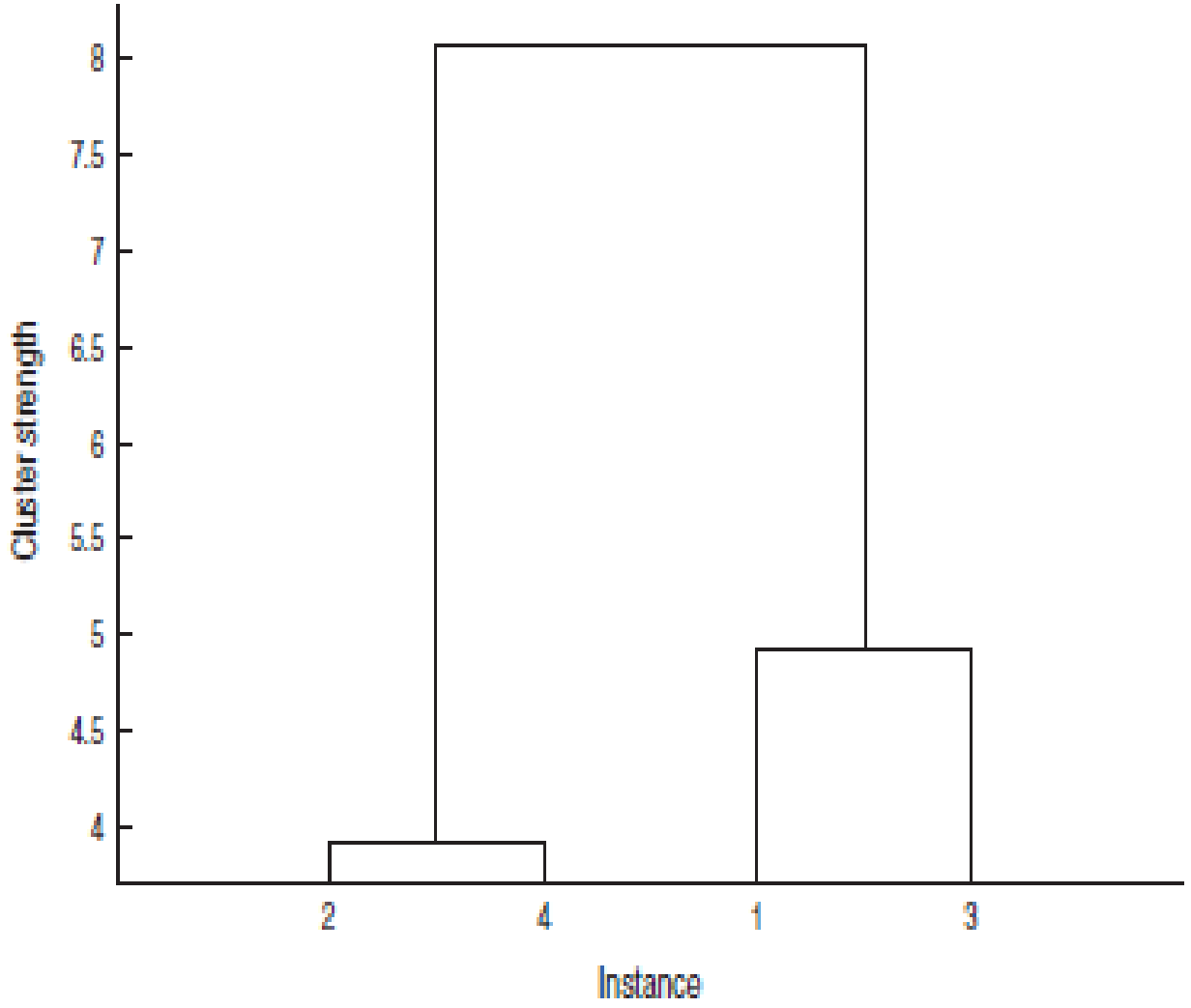


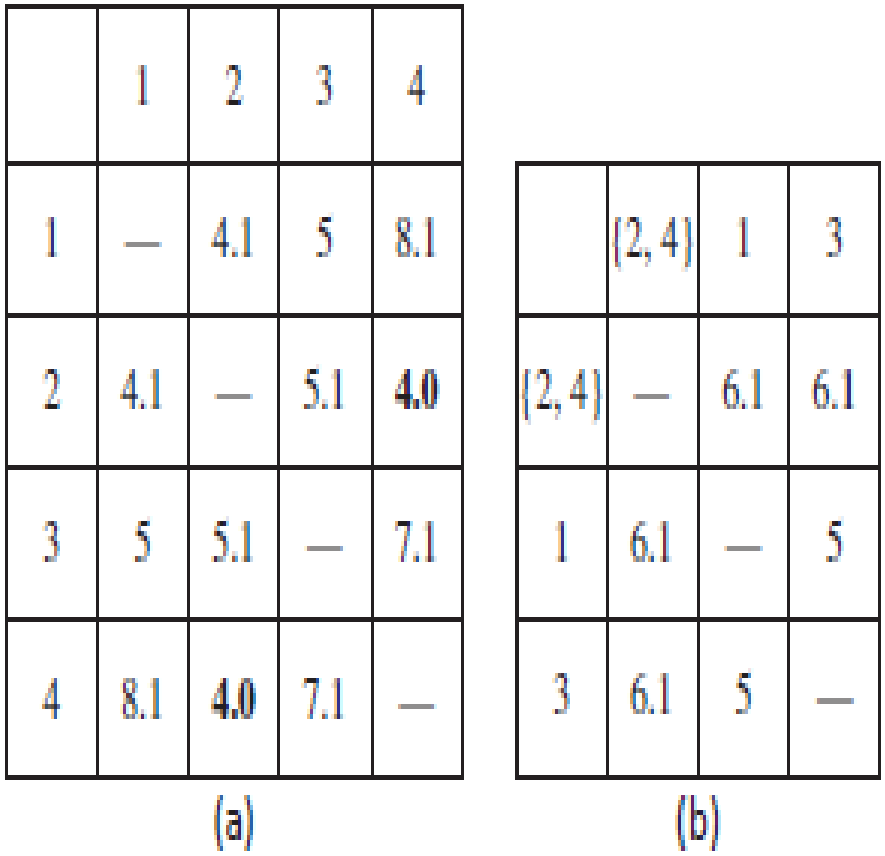
Fig. 11.21 Dendrogram for complete-linkage algorithm

Average-linking 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.



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

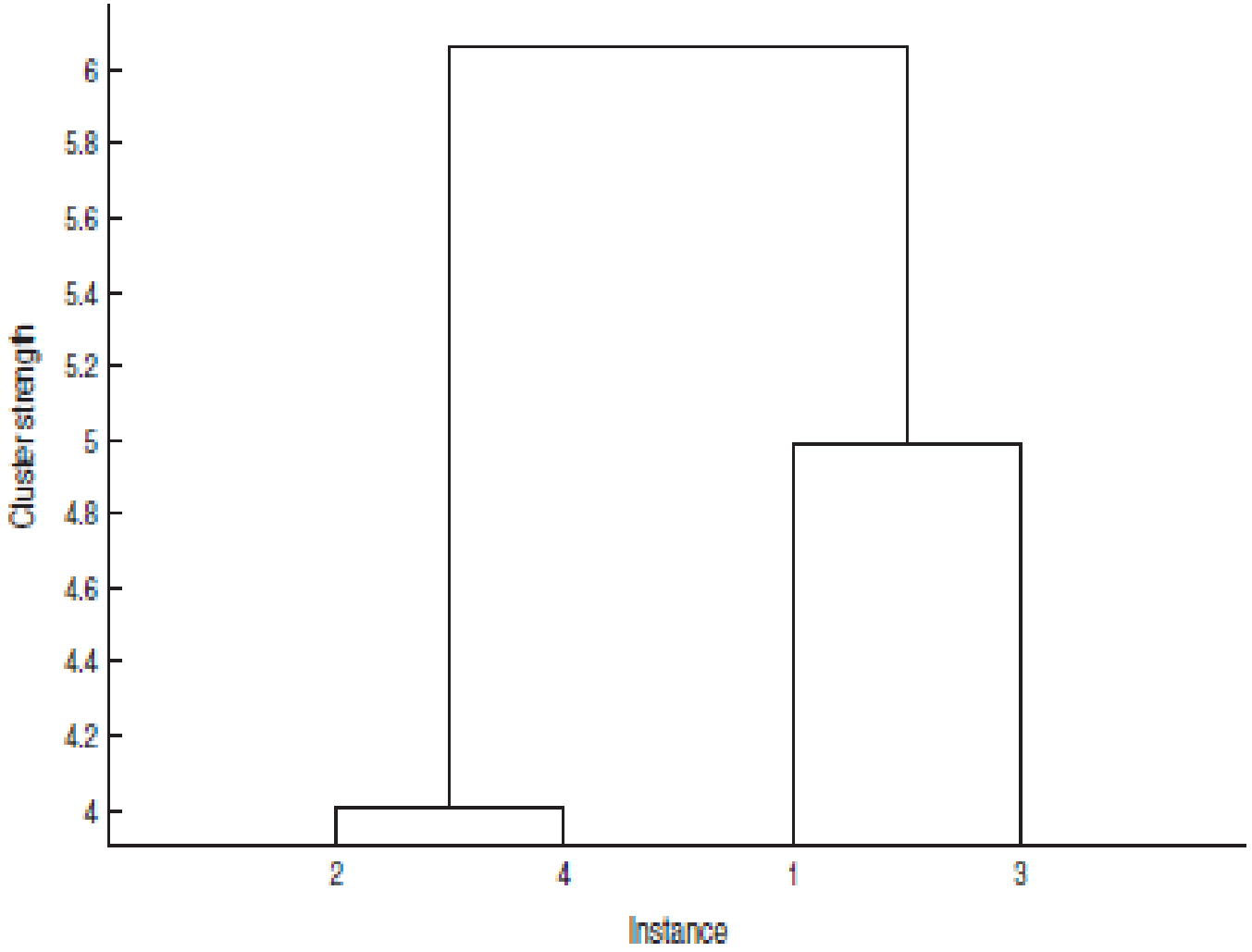


Fig. 11.23 Dendrogram for average-linkage algorithm

Partitional methods

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.

Cluster Evaluation Methods

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

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

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

   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 11.8 can be designed for cluster validation.

Table 11.8 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

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