Chapter-3

3.1 Classification and Prediction:

- Classification and prediction are two forms of data analysis that can be used to extractmodels describing important data classes or to predict future data trends.
- Classificationpredicts categorical (discrete, unordered) labels, *prediction* models continuousvaluedfunctions.
- For example, we can build a classification model to categorize bankloan applications as either safe or risky, or a prediction model to predict the expendituresof potential customers on computer equipment given their income and occupation.
- A predictor is constructed that predicts a continuous-valued function, or ordered value, as opposed to a categorical label.
- Regression analysis is a statistical methodology that is most often used for numeric prediction.
- Many classification and prediction methods have been proposed by researchers in machine learning, pattern recognition, and statistics.
- Most algorithms are memory resident, typically assuming a small data size. Recent data mining research has built on such work, developing scalable classification and prediction techniques capable of handling large disk-resident data.

3.1.1 Issues Regarding Classification and Prediction:

1.Preparing the Data for Classification and Prediction:

The following preprocessing steps may be applied to the data to help improve the accuracy, efficiency, and scalability of the classification or prediction process.

(i)Data cleaning:

- This refers to the preprocessing of data in order to remove or reduce *noise* (by applying smoothing techniques) and the treatment of *missingvalues* (e.g., by replacing a missing value with the most commonly occurring value for that attribute, or with the most probable value based on statistics).
- Although most classification algorithms have some mechanisms for handling noisy or missing data, this step can help reduce confusion during learning.

(ii)Relevance analysis:

- Many of the attributes in the data may be *redundant*.
- Correlation analysis can be used to identify whether any two given attributes are statisticallyrelated.
- For example, a strong correlation between attributes $A1$ and $A2$ would suggest that one of the two could be removed from further analysis.
- A database may also contain *irrelevant* attributes. Attribute subset selection can be used in these cases to find a reduced set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes.
- Hence, relevance analysis, in the form of correlation analysis and attribute subset selection, can be used to detect attributes that do not contribute to the classification or prediction task.
- Such analysis can help improve classification efficiency and scalability.

(iii)Data Transformation And Reduction

- The data may be transformed by normalization, particularly when neural networks or methods involving distance measurements are used in the learning step.
- Normalization involves scaling all values for a given attribute so that they fall within a small specified range, such as -1 to +1 or 0 to 1.
- The data can also be transformed by *generalizing* it to higher-level concepts. Concept hierarchies may be used for this purpose. This is particularly useful for continuous valuedattributes.

- For example, numeric values for the attribute *income* can be generalized to discrete ranges, such as *low, medium*, and *high*. Similarly, categorical attributes, like *street*, can be generalized to higher-level concepts, like *city.*
- Data can also be reduced by applying many other methods, ranging from wavelet transformation and principle components analysis to discretization techniques, such as binning, histogram analysis, and clustering.

3.1.2 Comparing Classification and Prediction Methods:

- **Accuracy:**
  - The accuracy of a classifier refers to the ability of a given classifier to correctly predict the class label of new or previously unseen data (i.e., tuples without class label information).
  - The accuracy of a predictor refers to how well a given predictor can guess the value of the predicted attribute for new or previously unseen data.
- **Speed:**

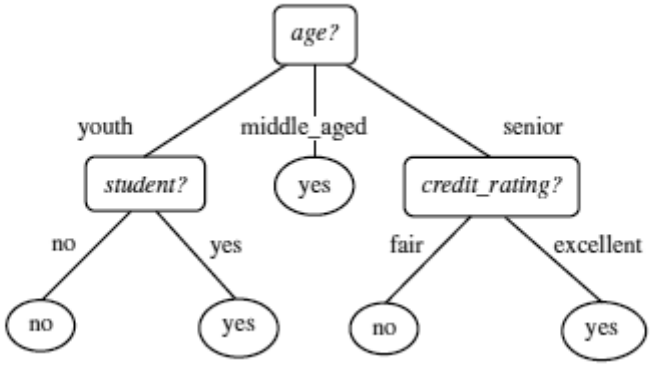
  This refers to the computational costs involved in generating and using the given classifier or predictor.
- **Robustness:**

  This is the ability of the classifier or predictor to make correct predictions given noisy data or data with missing values.
- **Scalability:**

  This refers to the ability to construct the classifier or predictor efficiently given large amounts of data.
- **Interpretability:**
  - This refers to the level of understanding and insight that is providedby the classifier or predictor.
  - Interpretability is subjective and therefore more difficultto assess.

3.2 Classification by Decision Tree Induction:

- Decision tree induction is the learning of decision trees from class-labeled training tuples.
- A decision tree is a flowchart-like tree structure,where
  - Each internal nodedenotes a test on an attribute.
  - Each branch represents an outcome of the test.
  - Each leaf node holds a class label.
  - The topmost node in a tree is the root node.



- The construction of decision treeclassifiers does not require any domain knowledge or parameter setting, and therefore I appropriate for exploratory knowledge discovery.
- Decision trees can handle high dimensionaldata.
- Their representation of acquired knowledge in tree formis intuitive and generallyeasy to assimilate by humans.
- The learning and classification steps of decision treeinduction are simple and fast.
- In general, decision tree classifiers have good accuracy.
- Decision tree induction algorithmshave been used for classification in many application areas, such as medicine,manufacturing and production, financial analysis, astronomy, and molecular biology.

3.2.1 Algorithm For Decision Tree Induction:

Algorithm: Generate_decision_tree. Generate a decision tree from the training tuples of data partition *D*.

Input:

- Data partition, *D*, which is a set of training tuples and their associated class labels;
- *attribute_list*, the set of candidate attributes;
- *Attribute_selection_method*, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a *splitting_attribute* and, possibly, either a *split point* or *splitting subset*.

Output: A decision tree.

Method:

```
(1)  create a node N;
(2)  If tuples in D are all of the same class, C then
(3)      return N as a leaf node labeled with the class C;
(4)  If attribute_list is empty then
(5)      return N as a leaf node labeled with the majority class in D; // majority voting
(6)  apply Attribute_selection_method(D, attribute_list) to find the "best" splitting_criterion;
(7)  label node N with splitting_criterion;
(8)  If splitting_attribute is discrete-valued and
         multiway splits allowed then // not restricted to binary trees
(9)      attribute_list ← attribute_list − splitting_attribute; // remove splitting_attribute
(10) for each outcome j of splitting_criterion
     // partition the tuples and grow subtrees for each partition
(11)     let Dj be the set of data tuples in D satisfying outcome j; // a partition
(12)     If Dj is empty then
(13)         attach a leaf labeled with the majority class in D to node N;
(14)     else attach the node returned by Generate_decision_tree(Dj, attribute_list) to node N;
     endfor
(15) return N;
```

The algorithm is called with three parameters:

- Data partition
- Attribute list
- Attribute selection method

- The parameter attribute list is a list ofattributes describing the tuples.
- Attribute selection method specifies a heuristic procedurefor selecting the attribute that "best" discriminates the given tuples according to class.
- The tree starts as a single node, *N*, representing the training tuples in *D*.

- If the tuples in *D* are all of the same class, then node *N* becomes a leaf and is labeledwith that class .
- Allof the terminating conditions are explained at the end of the algorithm.
- Otherwise, the algorithm calls Attribute selection method to determine the splitting criterion.
- The splitting criterion tells us which attribute to test at node N by determiningthe "best" way to separate or partition the tuples in D into individual classes.

There are three possible scenarios.Let *A* be the splitting attribute. *A* has *v* distinct values, {a_1, a_2, … ,a_v}, based on the training data.

1 A is discrete-valued:

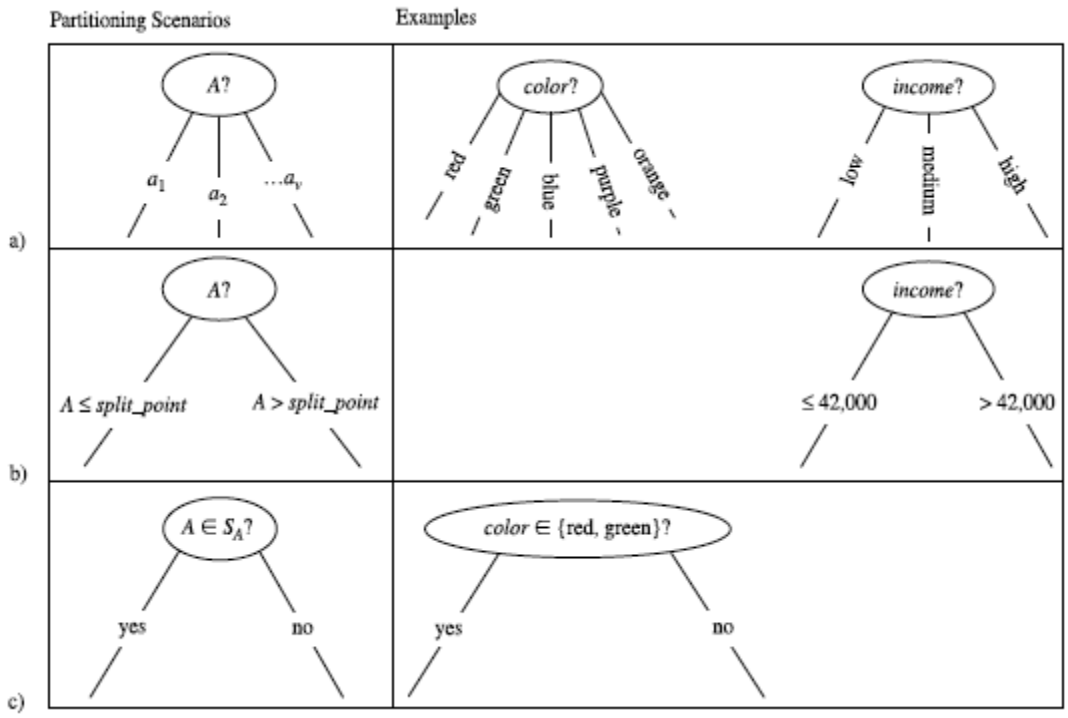
- In this case, the outcomes of the test at node N corresponddirectly to the known values of A.
- A branch is created for each known value, aj, of A and labeled with that value.
- Aneed not be considered in any future partitioning of the tuples.

2 A is continuous-valued:

In this case, the test at node N has two possible outcomes, corresponding to the conditions A <=split point and A >split point, respectively
wheresplit point is the split-point returned by Attribute selection method as part of the splitting criterion.

3 A is discrete-valued and a binary tree must be produced:

The test at node N is of the form"A€SA?".
SA is the splitting subset for A, returned by Attribute selection methodas part of the splitting criterion. It is a subset of the known values of A.



(a)If A is Discrete valued (b)If A is continuous valued (c) IfA is discrete-valued and a binary tree must be produced:

3.3 Bayesian Classification:

- Bayesian classifiers are statistical classifiers.
- They can predictclass membership probabilities, such as the probability that a given tuple belongs toa particular class.
- Bayesian classification is based on Bayes' theorem.

3.3.1 Bayes' Theorem:

- Let X be a data tuple. In Bayesian terms, X is considered "evidence."and it is described by measurements made on a set of n attributes.

- Let H be some hypothesis, such as that the data tuple X belongs to a specified class C.
- For classification problems, we want to determine P(H|X), the probability that the hypothesis H holds given the "evidence" or observed data tuple X.
- P(H|X) is the posterior probability, or a posteriori probability, of H conditioned on X.
- Bayes' theorem is useful in that it providesa way of calculating the posterior probability, $P(H|X)$, from $P(H)$, $P(X|H)$, and $P(X)$.

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}.$$

3.3.2 Naïve Bayesian Classification:

The naïve Bayesian classifier, or simple Bayesian classifier, works as follows:

1.Let D be a training set of tuples and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, X = (x1, x2, …,xn), depicting n measurements made on the tuple from n attributes, respectively, A1, A2, …, An.

2. Suppose that there are m classes, C1, C2, …, Cm. Given a tuple, X, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X.

That is, the naïve Bayesian classifier predicts that tuple X belongs to the class Ci if and only if

$$P(C_i|X) > P(C_j|X) \quad \text{for } 1 \le j \le m, j \ne i.$$

Thus we maximize P(Cij**X**). The classCifor which P(Cij**X**) is maximized is called the maximum posteriori hypothesis. By Bayes' theorem

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}.$$

3.As P(X) is constant for all classes, only P(X|Ci)P(Ci) need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, P(C1) = P(C2) = …= P(Cm), and we would therefore maximize P(X|Ci). Otherwise, we maximize P(X|Ci)P(Ci).

4.Given data sets with many attributes, it would be extremely computationally expensiveto compute P(X|Ci). In order to reduce computation in evaluating P(X|Ci), the naive assumption of class conditional independence is made. This presumes that the values of the attributes areconditionally independent of one another, given the class label of the tuple. Thus,

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$
$$= P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i).$$

We can easily estimate the probabilities $P(x1|Ci)$, $P(x2|Ci)$, : : : , $P(xn|Ci)$ fromthe trainingtuples. For eachattribute, we look at whether the attribute is categorical or continuous-valued. Forinstance, to compute $P(\boldsymbol{X}|Ci)$, we consider the following:

- If $A_k$is categorical, then $P(x_k|Ci)$ is the number of tuples of class Ciin D havingthe value $x_k$for A_k, divided by $|C_{i,D}|$ the number of tuples of class $C_i$in D.
- If $A_k$is continuous-valued, then we need to do a bit more work, but the calculationis pretty straightforward.

A continuous-valued attribute is typically assumed tohave a Gaussian distribution with a mean μ and standard deviation , defined by

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}).$$

5.In order to predict the class label of X, $P(X|Ci)P(Ci)$ is evaluated for each class Ci.

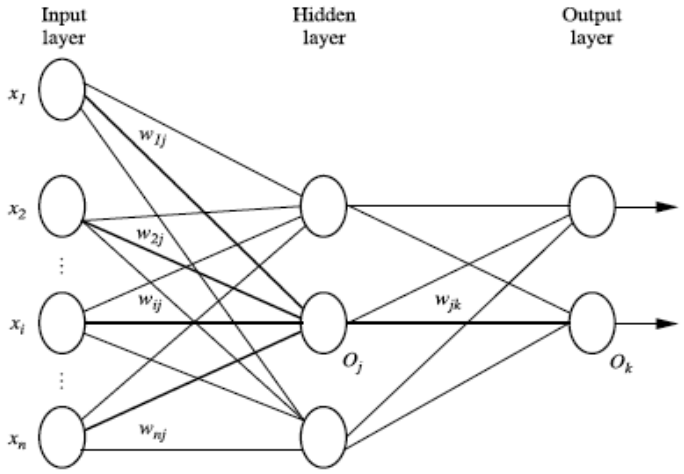
The classifier predicts that the class label of tuple X is the class Ciif and only if

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \quad \text{for } 1 \le j \le m, j \ne i.$$

3.4 A Multilayer Feed-Forward Neural Network:

- The backpropagation algorithm performs learning on a multilayer feed-forward neural network.
- It iteratively learns a set of weights for prediction of the class label of tuples.
- A multilayer feed-forward neural network consists of an input layer, one or more hiddenlayers, and an output layer.

Example:



- The inputs to the network correspond to the attributes measured for each training tuple. The inputs are fed simultaneously into the units making up the input layer. These inputs pass through the input layer and are then weighted and fed simultaneously to a second layer known as a hidden layer.
- The outputs of the hidden layer units can be input to another hidden layer, and so on. The number of hidden layers is arbitrary.
- The weighted outputs of the last hidden layer are input to units making up the output layer, which emits the network's prediction for given tuples

3.4.1 Classification by Backpropagation:

- Backpropagation is a neural network learning algorithm.
- A neural network is a set of connected input/output units inwhich each connection has a weightassociated with it.
- During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input tuples.
- Neural network learning is also referred to as connectionist learning due to the connections between units.

- Neural networks involve long training times and are therefore more suitable for applicationswhere this is feasible.
- Backpropagation learns by iteratively processing a data set of training tuples, comparing the network's prediction for each tuple with the actual known target value.
- The target value may be the known class label of the training tuple (for classification problems) or a continuous value (for prediction).
- For each training tuple, the weights are modified so as to minimize the mean squared errorbetween the network's prediction and the actual target value. These modifications are made in the "backwards" direction, that is, from the output layer, through each hidden layer down to the first hidden layer hence the name is backpropagation.
- Although it is not guaranteed, in general the weights will eventually converge, and the learning process stops.

Advantages:

- It include their high tolerance of noisy data as well as their ability to classify patterns on which they have not been trained.
- They can be used when you may have little knowledge of the relationships between attributesand classes.
- They are well-suited for continuous-valued inputs and outputs, unlike most decision tree algorithms.
- They have been successful on a wide array of real-world data, including handwritten character recognition, pathology and laboratory medicine, and training a computer to pronounce English text.
- Neural network algorithms are inherently parallel; parallelization techniques can be used to speed up the computation process.

Process:

Initialize the weights:

The weights in the network are initialized to small random numbers
ranging from-1.0 to 1.0, or -0.5 to 0.5. Each unit has a *bias* associated with it. The biases are similarly initialized to small random numbers.

Each training tuple, ***X***, is processed by the following steps.

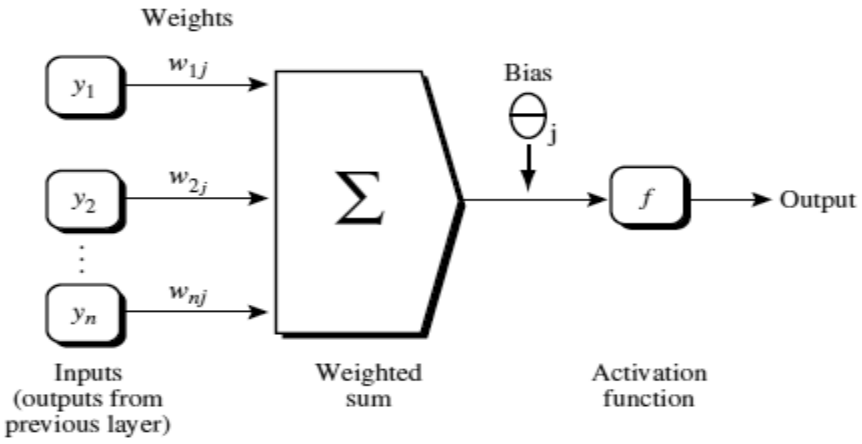
Propagate the inputs forward:

First, the training tuple is fed to the input layer of thenetwork. The inputs pass through the input units, unchanged. That is, for an input unit*j*, its output, *Oj*, is equal to its input value, *Ij*. Next, the net input and output of eachunit in the hidden and output layers are computed. The net input to a unit in the hiddenor output layers is computed as a linear combination of its inputs.
Each such unit has anumber of inputs to it that are, in fact, the outputs of the units connected to it in theprevious layer. Each connection has a weight. To compute the net input to the unit, each input connected to the unit ismultiplied by its correspondingweight, and this is summed.

$$I_j = \sum_i w_{ij} O_i + \theta_j,$$

where$w_{i,j}$is the weight of the connection from unit iin the previous layer to unit j;
$O_i$is the output of unit ifrom the previous layer
$\theta_j$is the bias of the unit & it actsas a threshold in that it serves to vary the activity of the unit.

Each unit in the hidden and output layers takes its net input and then applies an activation function to it.



Backpropagate the error:

The error is propagated backward by updating the weights and biases to reflect the error of the network's prediction. For a unit *j* in the output layer, the error Err_j is computed by

$$Err_j = O_j(1-O_j)(T_j - O_j)$$

where O_j is the actual output of unit j, and T_j is the known target value of the giventraining tuple.

The error of a hidden layerunit *j* is

$$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$$

where w_{jk} is the weight of the connection from unit j to a unit k in the next higher layer, and Err_k is the error of unit k.

Weights are updatedby the following equations, where D*wi j* is the change in weight w_{ij}:

$$\Delta w_{ij} = (l) Err_j O_i$$

$$w_{ij} = w_{ij} + \Delta w_{ij}$$

Biases are updated by the following equations below

$$\Delta\theta_j = (l) Err_j$$

$$\theta_j = \theta_j + \Delta\theta_j$$

Algorithm:

Input:

- D, a data set consisting of the training tuples and their associated target values;
- l, the learning rate;
- *network*, a multilayer feed-forward network.

Output: A trained neural network.

Method:

(1) Initialize all weights and biases in *network*;
(2) **while** terminating condition is not satisfied {
(3) **for** each training tuple $\boldsymbol{X}$ in D {
(4) // Propagate the inputs forward:
(5) **for** each input layer unit j {
(6) $O_j = I_j$; // output of an input unit is its actual input value
(7) **for** each hidden or output layer unit j {
(8) $I_j = \sum_i w_{ij} O_i + \theta_j$; //compute the net input of unit j with respect to the previous layer, i
(9) $O_j = \frac{1}{1+e^{-I_j}}$; } // compute the output of each unit j
(10) // Backpropagate the errors:
(11) **for** each unit j in the output layer
(12) $Err_j = O_j(1 - O_j)(T_j - O_j)$; // compute the error
(13) **for** each unit j in the hidden layers, from the last to the first hidden layer
(14) $Err_j = O_j(1 - O_j) \sum_k Err_k w_{jk}$; // compute the error with respect to the next higher layer, k
(15) **for** each weight w_{ij} in *network* {
(16) $\Delta w_{ij} = (l) Err_j O_i$; // weight increment
(17) $w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update
(18) **for** each bias θ_j in *network* {
(19) $\Delta \theta_j = (l) Err_j$; // bias increment
(20) $\theta_j = \theta_j + \Delta \theta_j$; } // bias update
(21) } }

3.5 k-Nearest-Neighbor Classifier:

- Nearest-neighbor classifiers are based on learning by analogy, that is, by comparing a given test tuplewith training tuples that are similar to it.
- The training tuples are describedby n attributes. Each tuple represents a point in an n-dimensional space. In this way,all of the training tuples are stored in an n-dimensional pattern space. When given anunknown tuple, a ***k***-nearest-neighbor classifier searches the pattern space for the k trainingtuples that are closest to the unknown tuple. These k training tuples are the k nearest neighbors of the unknown tuple.

- Closeness is defined in terms of a distance metric, such as Euclidean distance.
- The Euclidean distance between two points or tuples, say, $\boldsymbol{X}_1 = (x_{11}, x_{12}, \ldots, x_{1n})$ and $\boldsymbol{X}_2 = (x_{21}, x_{22}, \ldots, x_{2n})$, is

$$dist(\boldsymbol{X_1}, \boldsymbol{X_2}) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}.$$

In other words, for each numeric attribute, we take the difference between the corresponding values of that attribute in tuple $\boldsymbol{X_1}$and in tuple $\boldsymbol{X_2}$, square this difference,and accumulate it. The square root is taken of the total accumulated distance count.

Min-Max normalization can be used to transforma value v of a numeric attribute A to v_0 in therange [0, 1] by computing

$$v' = \frac{v - min_A}{max_A - min_A},$$

where$min_A$and $max_A$are the minimum and maximum values of attribute A

- For k-nearest-neighbor classification, the unknown tuple is assigned the mostcommon class among its k nearest neighbors.
- When $k = 1$, the unknown tuple is assigned the class of the training tuple that is closest to it in pattern space.
- Nearestneighborclassifiers can also be used for prediction, that is, to return a real-valued prediction for a given unknown tuple.
- In this case, the classifier returns the averagevalue of the real-valued labels associated with the k nearest neighbors of the unknowntuple.

3.6 Other Classification Methods:

3.6.1 Genetic Algorithms:

Genetic algorithms attempt to incorporate ideas of natural evolution. In general, geneticlearning starts as follows.

- An initial population is created consisting of randomly generated rules. Each rule can be represented by a string of bits. As a simple example, suppose that samples in a given

training set are described by two Boolean attributes, A1 and A2, and that there are two classes,C_1 andC_2.

- The rule "IF A_1 ANDNOT A_2 THENC_2" can be encoded as the bit string "100," where the two leftmost bits represent attributes A_1 and A_2, respectively, and the rightmost bit represents the class.
- Similarly, the rule "IF NOT A_1 AND NOT A_2 THEN C_1" can be encoded as "001."
- If an attribute has k values, where k > 2, then k bits may be used to encode the attribute's values.

Classes can be encoded in a similar fashion.

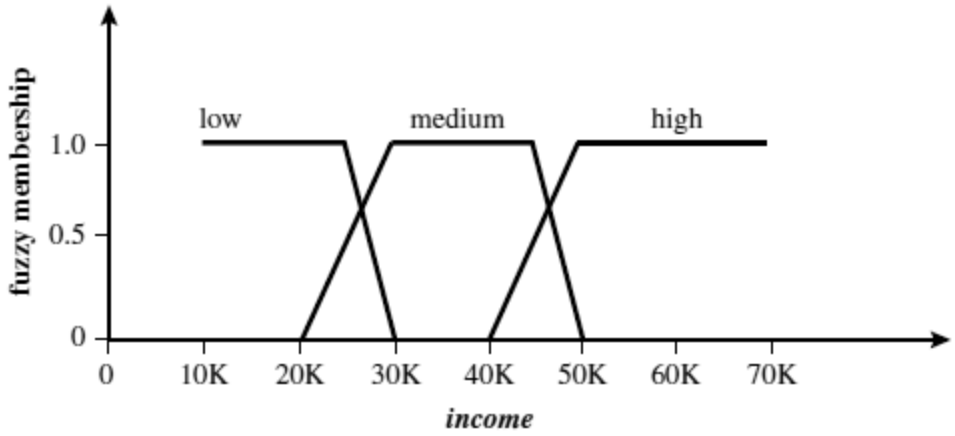
- Based on the notion of survival of the fittest, a new population is formed to consist of the fittest rules in the current population, as well as offspring of these rules.
- Typically, thefitness of a rule is assessed by its classification accuracy on a set of training samples.
- Offspring are created by applying genetic operators such as crossover and mutation.
- In crossover, substrings from pairs of rules are swapped to form new pairs of rules.
- Inmutation, randomly selected bits in a rule's string are inverted.
- The process of generating new populations based on prior populations of rules continues until a population, P, evolves where each rule in P satisfies a pre specified fitness threshold.
- Genetic algorithms are easily parallelizable and have been used for classification as well as other optimization problems. In data mining, they may be used to evaluate the fitness of other algorithms.

3.6.2 Fuzzy Set Approaches:

- Fuzzy logic uses truth values between 0.0 and 1.0 to represent the degree ofmembership that a certain value has in a given category. Each category then represents afuzzy set.
- Fuzzy logic systemstypically provide graphical tools to assist users in converting attribute values to fuzzy truthvalues.
- Fuzzy set theory is also known as possibility theory.

- It was proposed by LotfiZadeh in1965 as an alternative to traditional two-value logic and probability theory.
- It lets usworkat a high level of abstraction and offers a means for dealing with imprecise measurementof data.
- Most important, fuzzy set theory allows us to dealwith vague or inexact facts.
- Unlike the notion of traditional "crisp" sets where anelement either belongs to a set *S* or its complement, in fuzzy set theory, elements canbelong to more than one fuzzy set.
- Fuzzy set theory is useful for data mining systems performing rule-based classification.
- It provides operations for combining fuzzy measurements.
- Several procedures exist for translating the resulting fuzzy output into a *defuzzified*or crisp value that is returned by the system.
- Fuzzy logic systems have been used in numerous areas for classification, including market research, finance, health care, and environmental engineering.

Example:



3.7 Regression Analysis:

- Regression analysis can be used to model the relationship between one or moreindependent or predictor variables and a dependent or response variable which is continuous-valued.
- In the context of data mining, the predictor variables are theattributes of interest describing the tuple (i.e., making up the attribute vector).
- In general,the values of the predictor variables are known.

- The response variable is what we want to predict.

3.7.1 Linear Regression:

- Straight-line regression analysis involves a response variable, *y*, and a single predictor variable x.
- It is the simplest form of regression, and models *y* as a linear function of *x*.

That is, $y = b+wx$

where the variance of *y* is assumed to be constant

*b*and *w* are regression coefficientsspecifying the Y-intercept and slope of the line.

- The regression coefficients, *w* and *b*, can also be thought of as weights, so that we can equivalently write, $y = w_0+w_1x$
- These coefficients can be solved for by the method of least squares, which estimates the best-fitting straight line as the one that minimizes the error between the actual data and the estimate of the line.
- Let *D* be a training set consisting of values of predictor variable, *x*, for some population and their associated values for response variable, *y*. The training set contains |*D*| data points of the form$(x_1, y_1), (x_2, y_2), \ldots, (x_{|D|}, y_{|D|})$.

The regressioncoefficients can be estimated using this method with the following equations:

$$w_1 = \frac{\sum_{i=1}^{|D|}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|}(x_i - \bar{x})^2}$$

$$w_0 = \bar{y} - w_1\bar{x}$$

where *x* is the mean value of $x_1, x_2, \ldots, x_{|D|}$, and *y* is the mean value of $y_1, y_2, \ldots, y_{|D|}$.

The coefficients w_0 and w_1 often provide good approximations to otherwise complicated regression equations.

3.7.2 Multiple Linear Regression:

- It is an extension of straight-line regression so as to involve more than one predictor variable.

- It allows response variable y to be modeled as a linear function of, say, n predictor variables or attributes, A_1, A_2, …, An, describing a tuple, X.
- An example of a multiple linear regression model basedon two predictor attributes or variables, A_1 and A_2, is$y = w_0+w_1x_1+w_2x_2$

where x_1 and x_2 are the values of attributes A_1 and A_2, respectively, in X.

- Multiple regression problemsare instead commonly solved with the use of statistical software packages, such as SAS,SPSS, and S-Plus.

3.7.3 Nonlinear Regression:

- It can be modeled by adding polynomial terms to the basic linear model.
- By applying transformations to the variables, we can convert the nonlinear model into a linear one that can then be solved by the method of least squares.
- Polynomial Regression is a special case of multiple regression. That is, the addition of high-order terms like x^2, x^3, and so on, which are simple functions of the single variable, x, can be considered equivalent to adding new independent variables.

Transformation of a polynomial regression model to a linear regression model:

Consider a cubic polynomial relationship given by

$$y = w_0+w_1x+w_2x_2+w_3x_3$$

To convert this equation to linear form, we define new variables:

$x_1 = x$, $x_2 = x^2$,$x_3 = x^3$

It can then be converted to linear formby applying the above assignments,resulting in the equation$y = w_0+w_1x+w_2x_2+w_3x_3$

which is easily solved by themethod of least squares using software for regression analysis.

3.8 Classifier Accuracy:

- The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier.
- In the pattern recognition literature, this is also referred to as the overall recognition rate of theclassifier, that is, it reflects how well the classifier recognizes tuples of the various classes.

- The error rate or misclassification rate of a classifier,M, which is simply 1-Acc(M), where Acc(M) is the accuracy of M.
- The confusion matrix is a useful tool for analyzing how well your classifier can recognize tuples of different classes.
- True positives refer to the positive tuples that were correctly labeled by the classifier.
- True negatives are the negative tuples that were correctly labeled by the classifier.
- False positives are the negative tuples that were incorrectly labeled.
- How well the classifier can recognize, for this sensitivity and specificity measures can be used.

Accuracy is a function of sensitivity and specificity.

$$accuracy = sensitivity\frac{pos}{(pos+neg)} + specificity\frac{neg}{(pos+neg)}.$$

$$sensitivity = \frac{t\_pos}{pos}$$
$$specificity = \frac{t\_neg}{neg}$$
$$precision = \frac{t\_pos}{(t\_pos + f\_pos)}$$

wheret _posis the number of true positives

posis the number of positive tuples

t _negis the number of true negatives

negis the number of negative tuples,

f _posis the number of false positives

Chapter-4

4.1 Cluster Analysis:

- The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering.
- A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters.
- A cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression.
- Cluster analysis tools based on k-means, k-medoids, and several methods have also been built into many statisticalanalysis software packages or systems, such as S-Plus, SPSS, and SAS.

4.1.1 Applications:

- Cluster analysis has been widely used in numerous applications, including market research, pattern recognition, data analysis, and image processing.
- In business, clustering can help marketers discover distinct groups in their customer bases and characterize customer groups based on purchasing patterns.
- In biology, it can be used to derive plant and animal taxonomies, categorize genes with similar functionality, and gain insight into structures inherent in populations.
- Clustering may also help in the identification of areas of similar land use in an earth observation database and in the identification of groups of houses in a city according to house type, value,and geographic location, as well as the identification of groups of automobile insurance policy holders with a high average claim cost.
- Clustering is also called data segmentation in some applications because clustering partitions large data sets into groups according to their *similarity*.

- Clustering can also be used for outlier detection,Applications of outlier detection include the detection of credit card fraud and the monitoring of criminal activities in electronic commerce.

4.1.2 Typical Requirements Of Clustering InData Mining:

- **Scalability:**

  Many clustering algorithms work well on small data sets containing fewer than several hundred data objects; however, a large database may contain millions of objects. Clustering on a sample of a given large data set may lead to biased results.

  Highly scalable clustering algorithms are needed.

- **Ability to deal with different types of attributes:**

  Many algorithms are designed to cluster interval-based (numerical) data. However, applications may require clustering other types of data, such as binary, categorical (nominal), and ordinal data, or mixtures of these data types.

- **Discovery of clusters with arbitrary shape:**

  Many clustering algorithms determine clusters based on Euclidean or Manhattan distance measures. Algorithms based on such distance measures tend to find spherical clusters with similar size and density.

  However, a cluster could be of any shape. It is important to develop algorithms thatcan detect clusters of arbitrary shape.

- **Minimal requirements for domain knowledge to determine input parameters:**

  Many clustering algorithms require users to input certain parameters in cluster analysis (such as the number of desired clusters). The clustering results can be quite sensitive to input parameters. Parameters are often difficult to determine, especially for data sets containing high-dimensional objects. This not only burdens users, but it also makes the quality of clustering difficult to control.

- **Ability to deal with noisy data:**

  Most real-world databases contain outliers or missing, unknown, or erroneous data.

  Some clustering algorithms are sensitive to such data and may lead to clusters of poor quality.

- **Incremental clustering and insensitivity to the order of input records:**

  Some clustering algorithms cannot incorporate newly inserted data (i.e., database updates) into existing clustering structures and, instead, must determine a new clustering from scratch. Some clustering algorithms are sensitive to the order of input data.

  That is, given a set of data objects, such an algorithm may return dramatically different clusterings depending on the order of presentation of the input objects.

  It is important to develop incremental clustering algorithms and algorithms thatare insensitive to the order of input.

- **High dimensionality:**

  A database or a data warehouse can contain several dimensionsor attributes.Many clustering algorithms are good at handling low-dimensional data,involving only two to three dimensions. Human eyes are good at judging the qualityof clustering for up to three dimensions. Finding clusters of data objects in highdimensionalspace is challenging, especially considering that such data can be sparseand highly skewed.

- **Constraint-based clustering:**

  Real-world applications may need to perform clustering under various kinds of constraints. Suppose that your job is to choose the locations for a given number of new automatic banking machines (ATMs) in a city. To decide upon this, you may cluster households while considering constraints such as the city's rivers and highway networks, and the type and number of customers per cluster. A challenging task is to find groups of data with good clustering behavior that satisfy specified constraints.

- **Interpretability and usability:**

  Users expect clustering results to be interpretable, comprehensible, and usable. That is, clustering may need to be tied to specific semantic interpretations and applications. It is important to study how an application goal may influence the selection of clustering features and methods.

4.2 Major Clustering Methods:

- Partitioning Methods
- Hierarchical Methods

- Density-Based Methods
- Grid-Based Methods
- Model-Based Methods

4.2.1 Partitioning Methods:

A partitioning method constructs *k* partitions of the data, where each partition represents a cluster and *k* <= *n*. That is, it classifies the data into *k* groups, which together satisfy the following requirements:

- Each group must contain at least one object, and
- Each object must belong to exactly one group.

A partitioning method creates an initial partitioning. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another.

The general criterion of a good partitioning is that objects in the same cluster are close or related to each other, whereas objects of different clusters are far apart or very different.

4.2.2 Hierarchical Methods:

A hierarchical method creates a hierarchical decomposition ofthe given set of data objects. A hierarchical method can be classified as being eitheragglomerative or divisive, based on howthe hierarchical decomposition is formed.

- Theagglomerative approach, also called the bottom-up approach, starts with each objectforming a separate group. It successively merges the objects or groups that are closeto one another, until all of the groups are merged into one or until a termination condition holds.
- The divisive approach, also calledthe top-down approach, starts with all of the objects in the same cluster. In each successiveiteration, a cluster is split up into smaller clusters, until eventually each objectis in one cluster, or until a termination condition holds.

Hierarchical methods suffer fromthe fact that once a step (merge or split) is done,it can never be undone. This rigidity is useful in that it leads to smaller computationcosts by not having toworry about a combinatorial number of different choices.

There are two approachesto improving the quality of hierarchical clustering:

- Perform careful analysis ofobject "linkages" at each hierarchical partitioning, such as in Chameleon, or
- Integratehierarchical agglomeration and other approaches by first using a hierarchicalagglomerative algorithm to group objects into microclusters, and then performingmacroclustering on the microclusters using another clustering method such as iterative relocation.

4.2.3 Density-based methods:

- Most partitioning methods cluster objects based on the distance between objects. Such methods can find only spherical-shaped clusters and encounter difficulty at discovering clusters of arbitrary shapes.
- Other clustering methods have been developed based on the notion of density. Their general idea is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold; that is, for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points. Such a method can be used to filter out noise (outliers)and discover clusters of arbitrary shape.
- DBSCAN and its extension, OPTICS, are typical density-based methods that growclusters according to a density-based connectivity analysis. DENCLUE is a methodthat clusters objects based on the analysis of the value distributions of density functions.

4.2.4 Grid-Based Methods:

- Grid-based methods quantize the object space into a finite number of cells that form a grid structure.

- ❖ All of the clustering operations are performed on the grid structure i.e., on the quantized space. The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space.
- ❖ STING is a typical example of a grid-based method. Wave Cluster applies wavelet transformation for clustering analysis and is both grid-based and density-based.

4.2.5 Model-Based Methods:

- ❖ Model-based methods hypothesize a model for each of the clusters and find the best fit of the data to the given model.
- ❖ A model-based algorithm may locate clusters by constructing a density function that reflects the spatial distribution of the data points.
- ❖ It also leads to a way of automatically determining the number of clusters based on standard statistics, taking "noise" or outliers into account and thus yielding robust clustering methods.

4.3 Tasks in Data Mining:

- ➢ Clustering High-Dimensional Data
- ➢ Constraint-Based Clustering

4.3.1 Clustering High-Dimensional Data:

- It is a particularly important task in cluster analysis because many applications require the analysis of objects containing a large number of features or dimensions.
- For example, text documents may contain thousands of terms or keywords as features, and DNA micro array data may provide information on the expression levels of thousands of genes under hundreds of conditions.
- Clustering high-dimensional data is challenging due to the curse of dimensionality.
- Many dimensions may not be relevant. As the number of dimensions increases, thedata become increasingly sparse so that the distance measurement between pairs ofpoints become meaningless and the average density of points anywhere in the data islikely to be low. Therefore, a different clustering methodology needs to be developedfor high-dimensional data.

- CLIQUE and PROCLUS are two influential subspace clustering methods, which search for clusters in subspaces ofthe data, rather than over the entire data space.
- Frequent pattern–based clustering,another clustering methodology, extractsdistinct frequent patterns among subsets ofdimensions that occur frequently. It uses such patterns to group objects and generatemeaningful clusters.

4.3.2 Constraint-Based Clustering:

- It is a clustering approach that performs clustering by incorporation of user-specified or application-oriented constraints.
- A constraint expresses a user's expectation or describes properties of the desired clustering results, and provides an effective means for communicating with the clustering process.
- Various kinds of constraints can be specified, either by a user or as per application requirements.
- Spatial clustering employs with the existence of obstacles and clustering under user-specified constraints. In addition, semi-supervised clusteringemploys forpairwise constraints in order to improvethe quality of the resulting clustering.

4.4 Classical Partitioning Methods:

The mostwell-known and commonly used partitioningmethods are

- ❖ The *k*-Means Method
- ❖ k-Medoids Method

4.4.1 Centroid-Based Technique: The *K*-Means Method:

The k-means algorithm takes the input parameter, k, and partitions a set of n objects intok clusters so that the resulting intracluster similarity is high but the intercluster similarity is low.

Cluster similarity is measured in regard to the mean value of the objects in a cluster, which can be viewed as the cluster's centroid or center of gravity.

The *k*-means algorithm proceeds as follows.

- First, it randomly selects *k* of the objects, each of which initially represents a cluster mean or center.
- For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean.
- It then computes the new mean for each cluster.
- This process iterates until the criterion function converges.

Typically, the square-error criterion is used, defined as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2,$$

whereE is the sum of the square error for all objects in the data set

pis the point in space representing a given object

$m_i$is the mean of cluster C_i

4.4.1 The k-means partitioning algorithm:

The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

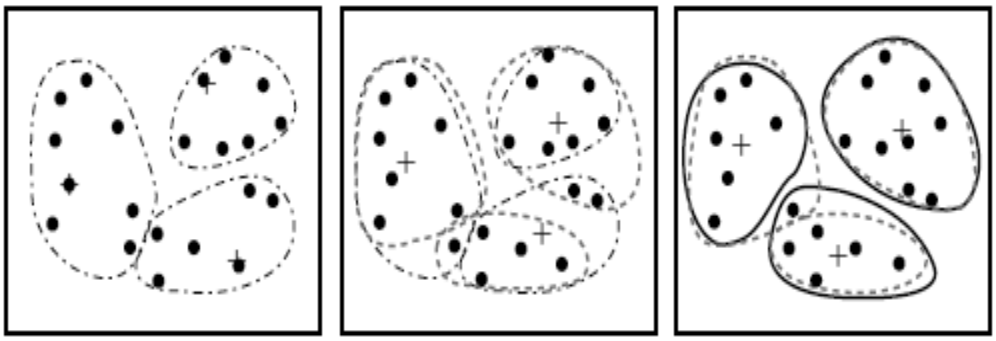
Input:

- *k*: the number of clusters,
- *D*: a data set containing *n* objects.

Output: A set of *k* clusters.

Method:

(1) arbitrarily choose *k* objects from *D* as the initial cluster centers;
(2) **repeat**
(3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
(4) update the cluster means, i.e., calculate the mean value of the objects for each cluster;
(5) **until** no change;



Clustering of a set of objects based on the *k*-means method

4.4.2 The *k*-Medoids Method:

- The k-means algorithm is sensitive to outliers because an object with an extremely large value may substantially distort the distribution of data. This effect is particularly exacerbated due to the use of the square-error function.
- Instead of taking the mean value of the objects in a cluster as a reference point, we can pick actual objects to represent the clusters, using one representative object per cluster. Each remaining object is clustered with the representative object to which it is the most similar.
- Thepartitioning method is then performed based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point. That is, an absolute-error criterion is used, defined as

$$E = \sum_{j=1}^{k} \sum_{p \in C_j} |p - o_j|,$$

where*E* is the sum of the absolute error for all objects in the data set

pis the point inspace representing a given object in clusterC_j

$\mathbf{o_j}$is the representative object of C_j

- The initial representative objects are chosen arbitrarily. The iterative process of replacing representative objects by non representative objects continues as long as the quality of the resulting clustering is improved.
- This quality is estimated using a cost function that measures the average dissimilaritybetween an object and the representative object of its cluster.
- To determine whether a non representative object, oj random, is a good replacement for a current representativeobject, oj, the following four cases are examined for each of the nonrepresentative objects.

Case 1:

pcurrently belongs to representative object, o_j. If $o_j$is replaced by o_{random}asa representative object and p is closest to one of the other representative objects, o_i,$i \neq j$, then p is reassigned to o_i.

Case 2:

pcurrently belongs to representative object, oj. If ojis replaced by o_{random}asa representative object and p is closest to o_{random}, then p is reassigned to o_{random}.

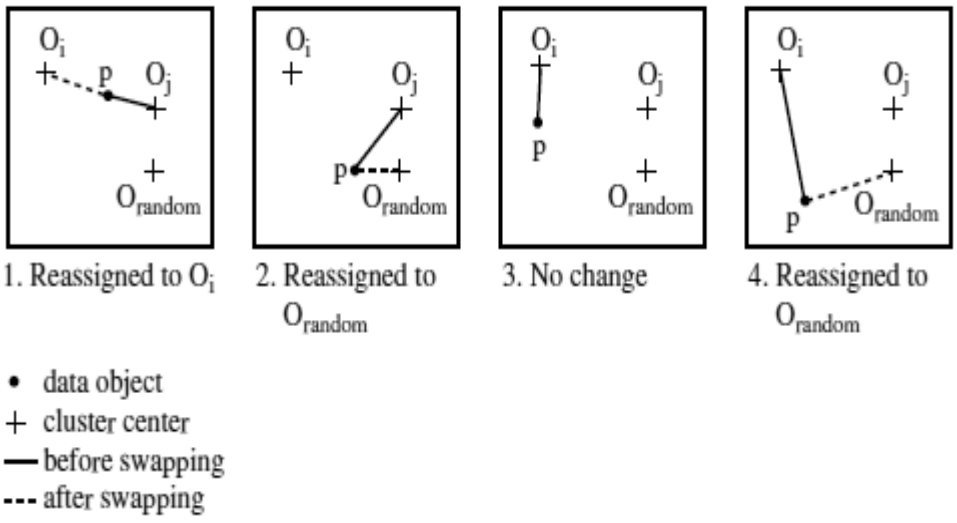
Case 3:

pcurrently belongs to representative object, o_i, $i \neq j$. If ojis replaced by o_{random}as a representative object and p is still closest to o_i, then the assignment does notchange.

Case 4:

pcurrently belongs to representative object, oi, $i \neq j$. If $o_j$is replaced byo_{random}as a representative object and p is closest to o_{random}, then p is reassigned

too$_{random}$.



Four cases of the cost function for *k*-medoids clustering

4.4.2 The*k*-MedoidsAlgorithm:

The k-medoids algorithm for partitioning based on medoid or central objects.

Input:

- *k*: the number of clusters,
- *D*: a data set containing *n* objects.

Output: A set of *k* clusters.

Method:

(1) arbitrarily choose *k* objects in *D* as the initial representative objects or seeds;
(2) **repeat**
(3) assign each remaining object to the cluster with the nearest representative object;
(4) randomly select a nonrepresentative object, o_{random};
(5) compute the total cost, *S*, of swapping representative object, o_j, with o_{random};
(6) **if** $S < 0$ **then** swap o_j with o_{random} to form the new set of *k* representative objects;
(7) **until** no change;

The *k*-medoids method ismore robust than *k*-means in the presence of noise and outliers, because a medoid is lessinfluenced by outliers or other extreme values than a mean. However, its processing ismore costly than the *k*-means method.

4.5 Hierarchical Clustering Methods:

- A hierarchical clustering method works by grouping data objects into a tree of clusters.
- The quality of a pure hierarchical clusteringmethod suffers fromits inability to performadjustment once amerge or split decision hasbeen executed. That is, if a particular merge or split decision later turns out to have been apoor choice, the method cannot backtrack and correct it.

Hierarchical clustering methods can be further classified as either agglomerative or divisive, depending on whether the hierarchical decomposition is formed in a bottom-up or top-down fashion.

4.5.1 Agglomerative hierarchical clustering:

- This bottom-up strategy starts by placing each object in its own cluster and then merges these atomic clusters into larger and larger clusters, until all of the objects are in a single cluster or until certain termination conditions are satisfied.
- Most hierarchical clustering methods belong to this category. They differ only in their definition of intercluster similarity.

4.5.2 Divisive hierarchical clustering:

- This top-down strategy does the reverse of agglomerativehierarchical clustering by starting with all objects in one cluster.
- It subdividesthe cluster into smaller and smaller pieces, until each object forms a cluster on itsown or until it satisfies certain termination conditions, such as a desired number ofclusters is obtained or the diameter of each cluster is within a certain threshold.

4.6 Constraint-Based Cluster Analysis:

Constraint-based clustering finds clusters that satisfy user-specified preferences orconstraints. Depending on the nature of the constraints, constraint-based clusteringmay adopt rather different approaches.

There are a few categories of constraints.

- **Constraints on individual objects:**

  We can specify constraints on the objects to beclustered. In a real estate application, for example, one may like to spatially cluster only those luxury mansions worth over a million dollars. This constraint confines the setof objects to be clustered. It can easily be handled by preprocessing after which the problem reduces to an instance ofunconstrained clustering.

- **Constraints on the selection of clustering parameters:**

  A user may like to set a desired range for each clustering parameter. Clustering parameters are usually quite specific to the given clustering algorithm. Examples of parameters include k, the desired numberof clusters in a k-means algorithm; or e the radius and the minimum number of points in the DBSCAN algorithm. Although such user-specified parameters may strongly influence the clustering results, they are usually confined to the algorithm itself. Thus, their fine tuning and processing are usually not considered a form of constraint-based clustering.

- **Constraints on distance or similarity functions:**

  We can specify different distance orsimilarity functions for specific attributes of the objects to be clustered, or differentdistance measures for specific pairs of objects.When clustering sportsmen, for example,we may use different weighting schemes for height, body weight, age, and skilllevel. Although this will likely change the mining results, it may not alter the clusteringprocess per se. However, in some cases, such changes may make the evaluationof the distance function nontrivial, especially when it is tightly intertwined with the clustering process.

- **User-specified constraints on the properties of individual clusters:**
  A user may like tospecify desired characteristics of the resulting clusters, which may strongly influencethe clustering process.
- **Semi-supervised clustering based on partial supervision:**
  The quality of unsupervisedclustering can be significantly improved using some weak form of supervision.This may be in the formof pairwise constraints (i.e., pairs of objects labeled as belongingto the same or different cluster). Such a constrained clustering process is calledsemi-supervised clustering.

4.7 Outlier Analysis:

- There exist data objects that do not comply with the general behavior or model of the data. Such data objects, which are grossly different from or inconsistent with the remaining set of data, are called outliers.
- Many data mining algorithms try to minimize the influence of outliers or eliminate them all together. This, however, could result in the loss of important hidden information because one person's noise could be another person's signal. In other words, the outliers may be of particular interest, such as in the case of fraud detection, where outliers may indicate fraudulent activity. Thus, outlier detection and analysis is an interesting data mining task, referred to as outlier mining.
- It can be used in fraud detection, for example, by detecting unusual usage of credit cards or telecommunication services. In addition, it is useful in customized marketing for identifying the spending behavior of customers with extremely low or extremely high incomes, or in medicalanalysis for finding unusual responses to various medical treatments.

Outlier mining can be described as follows: Given a set of *n* data points or objectsand *k*, the expected number of outliers, find the top *k* objects that are considerablydissimilar, exceptional, or inconsistent with respect to the remaining data. The outliermining problem can be viewed as two subproblems:

- Define what data can be considered as inconsistent in a given data set, and
- Find an efficient method to mine the outliers so defined.

Types of outlier detection:

- Statistical Distribution-Based Outlier Detection
- Distance-Based Outlier Detection
- Density-Based Local Outlier Detection
- Deviation-Based Outlier Detection

4.7.1 Statistical Distribution-Based Outlier Detection:

The statistical distribution-based approach to outlier detection assumes a distributionor probability model for the given data set (e.g., a normal or Poisson distribution) andthen identifies outliers with respect to the model using a discordancy test. Application ofthe test requires knowledge of the data set parameters knowledge of distribution parameters such as the mean and variance and theexpected number of outliers.

A statistical discordancy test examines two hypotheses:

- A working hypothesis
- An alternative hypothesis

A working hypothesis, H, is a statement that the entire data set of n objects comes from an initial distribution model, F, that is,

$$H : o_i \in F, \quad \text{where } i = 1, 2, \ldots, n.$$

The hypothesis is retained if there is no statistically significant evidence supporting its rejection. A discordancy test verifies whether an object, oi, is significantly large (or small) in relation to the distribution F. Different test statistics have been proposed for use as a discordancy test, depending on the available knowledge of the data. Assuming that some statistic, T, has been chosen for discordancy testing, and the value of the statistic for object oi is vi, then the distribution of T is constructed. Significance probability, SP(vi)=Prob(T > vi), is evaluated. If SP(vi) is sufficiently small, then oi is discordant and the working hypothesis is rejected.

An alternative hypothesis, H, which states that o_i comes from another distribution model, G, is adopted. The result is very much dependent on which model F is chosen because $o_i$may be an outlier under one model and a perfectly valid value under another. The

alternative distribution is very important in determining the power of the test, that is, the probability that the working hypothesis is rejected when oi is really an outlier.

There are different kinds of alternative distributions.

- **Inherent alternative distribution:**

  In this case, the working hypothesis that all of the objects come from distribution F is rejected in favor of the alternative hypothesis that all of the objects arise from another distribution, G:

  H :oi € G, where i = 1, 2,…, n

  F and G may be different distributions or differ only in parameters of the same distribution.

  There are constraints on the form of the *G* distribution in that it must have potential to produce outliers. For example, it may have a different mean or dispersion, or a longer tail.

- **Mixture alternative distribution:**

  The mixture alternative states that discordant values are not outliers in the F population, but contaminants from some other population,

  G. In this case, the alternative hypothesis is

$$\overline{H} : o_i \in (1-\lambda)F + \lambda G, \quad \text{where } i = 1, 2, \ldots, n.$$

- **Slippage alternative distribution:**

  This alternative states that all of the objects (apart from some prescribed small number) arise independently from the initial model, F, with its given parameters, whereas the remaining objects are independent observations from a modified version of F in which the parameters have been shifted.

There are two basic types of procedures for detecting outliers:

Block procedures:

In this case, either all of the suspect objects are treated as outliersor all of them are accepted as consistent.

Consecutive procedures:

An example of such a procedure is the *insideout*procedure. Its main idea is that the object that is least likely to be an outlier istested first. If it is found to be an outlier, then all of the

more extreme values are alsoconsidered outliers; otherwise, the next most extreme object is tested, and so on. Thisprocedure tends to be more effective than block procedures.

4.7.2 Distance-Based Outlier Detection:

The notion of distance-based outliers was introduced to counter the main limitationsimposed by statistical methods. An object, o, in a data set, D, is a distance-based (DB)outlier with parameters pct and dmin,that is, a DB(pct;dmin)-outlier, if at least a fraction,pct, of the objects in D lie at a distance greater than dmin from o. In other words, rather thatrelying on statistical tests, we can think of distance-based outliers as thoseobjects that do not have enoughneighbors, where neighbors are defined based ondistance from the given object. In comparison with statistical-based methods, distancebased outlier detection generalizes the ideas behind discordancy testing for various standarddistributions. Distance-based outlier detection avoids the excessive computationthat can be associated with fitting the observed distribution into some standard distributionand in selecting discordancy tests.

For many discordancy tests, it can be shown that if an object, o, is an outlier accordingto the given test, then o is also a DB(pct, dmin)-outlier for some suitably defined pct anddmin.

For example, if objects that lie three or more standard deviations from the mean
are considered to be outliers, assuming a normal distribution, then this definition can
be generalized by a DB(0.9988, 0.13s) outlier.

Several efficient algorithms for mining distance-based outliers have been developed.

Index-based algorithm:

Given a data set, the index-based algorithm uses multidimensionalindexing structures, such as R-trees or k-d trees, to search for neighbors of eachobject ***o*** within radius *dmin*around that object. Let *M*be the maximum number ofobjects within the *dmin*-neighborhood of an outlier. Therefore, once*M*+1 neighborsof object ***o*** are found, it is clear that ***o*** is not an outlier. This algorithm has a worst-casecomplexity of $O(n2k)$, where n is the number of objects in the data set and k is thedimensionality. The index-based algorithm scales well as k increases. However, thiscomplexity evaluation takes only the search time into account, even though the taskof building an index in itself can be computationally intensive.

Nested-loop algorithm:

The nested-loop algorithm has the same computational complexityas the index-based algorithm but avoids index structure construction and triesto minimize the number of I/Os. It divides the memory buffer space into two halvesand the data set into several logical blocks. By carefully choosing the order in whichblocks are loaded into each half, I/O efficiency can be achieved.

Cell-based algorithm:

To avoid$O(n^2)$ computational complexity, a cell-based algorithm was developed for memory-resident data sets. Its complexity is $O(c^k+n)$, where c is a constant depending on the number of cells and k is the dimensionality.

In this method, the data space is partitioned into cells with a side length equal to $\frac{dmin}{2\sqrt{k}}$. Eachcell has two layers surrounding it. The first layer is one cell thick, while the secondis $\lceil 2\sqrt{k}-1 \rceil$ cells thick, rounded up to the closest integer. The algorithm countsoutliers on a cell-by-cell rather than an object-by-object basis. For a given cell, itaccumulates three counts—the number of objects in the cell, in the cell and the firstlayer together, and in the cell and both layers together. Let's refer to these counts ascell count, cell + 1 layer count, and cell + 2 layers count, respectively.

Let Mbe the maximum number ofoutliers that can exist in the dmin-neighborhood of an outlier.

- An object, **o**, in the current cell is considered an outlier only if cell + 1 layer countis less than or equal to M. If this condition does not hold, then all of the objectsin the cell can be removed from further investigation as they cannot be outliers.
- If cell_+ 2_layers_count is less than or equal to M, then all of the objects in thecell are considered outliers. Otherwise, if this number is more than M, then itis possible that some of the objects in the cell may be outliers. To detect theseoutliers, object-by-object processing is used where, for each object, **o**, in the cell,objects in the second layer of **o** are examined. For objects in the cell, only thoseobjects having no more than M points in their dmin-neighborhoods are outliers.The dmin-neighborhood of an object consists of the object's cell, all of its firstlayer, and some of its second layer.

A variation to the algorithm is linear with respect to n and guarantees that no morethan three passes over the data set are required. It can be used for large disk-residentdata sets, yet does not scale well for high dimensions.

4.7.3 Density-Based Local Outlier Detection:

Statistical and distance-based outlier detection both depend on the overall or globaldistribution of the given set of data points, D. However, data are usually not uniformlydistributed. These methods encounter difficulties when analyzing data with rather different

density distributions.

To define the local outlier factor of an object, we need to introduce the concepts ofk-distance, k-distance neighborhood, reachability distance,13 and local reachability density.

These are defined as follows:

The k-distance of an object p is the maximal distance that p gets from its k-nearestneighbors. This distance is denoted as k-distance(p). It is defined as the distance,

d(p, o), between p and an object o 2 D, such that for at least k objects, o_0 2 D, it holds that d(p, o·)_d(p, o). That is, there are at least k objects inDthat are as close asor closer to p than o, and for at most k-1 objects, o00 2 D, it holds that d(p;o'') <d(p, o).

That is, there are at most k-1 objects that are closer to p than o. You may bewondering at this point how k is determined. The LOF method links to density-basedclustering in that it sets k to the parameter rMinPts,which specifies the minimumnumberof points for use in identifying clusters based on density.

Here, MinPts (as k) is used to define the local neighborhood of an object, p.

The k-distance neighborhood of an object p is denoted $N_{kdistance(p)(}p)$, or $N_k(p)$for short. By setting k to MinPts, we get $N_{MinPts}(p)$. It contains the MinPts-nearestneighbors of p. That is, it contains every object whose distance is not greater than theMinPts-distance of p.

The reachability distance of an object p with respect to object o (where o is within

theMinPts-nearest neighbors of p), is defined as reach

distMinPts(p, o) = max{MinPtsdistance(o), d(p, o)}.

Intuitively, if an object p is far away , then the reachabilitydistance between the two is simply their actual distance. However, if they are sufficientlyclose (i.e., where p is within the MinPts-distance neighborhood of o), thenthe actual distance is replaced by the MinPts-distance of o. This helps to significantlyreduce the statistical fluctuations of d(p, o) for all of the p close to o.

The higher thevalue of MinPts is, the more similar is the reachability distance for objects withinthe same neighborhood.

Intuitively, the local reachability density of p is the inverse of the average reachability density based on the MinPts-nearest neighbors of p. It is defined as

$$lrd_{MinPts}(\boldsymbol{p}) = \frac{|N_{MinPts}(\boldsymbol{p})|}{\sum_{\boldsymbol{o} \in N_{MinPts}(\boldsymbol{p})} reach\_dist_{MinPts}(\boldsymbol{p}, \boldsymbol{o})}.$$

The local outlier factor (LOF) of $\boldsymbol{p}$ captures the degree to which we call $\boldsymbol{p}$ an outlier.

It is defined as

$$LOF_{MinPts}(\boldsymbol{p}) = \frac{\sum_{\boldsymbol{o} \in N_{MinPts}(\boldsymbol{p})} \frac{lrd_{MinPts}(\boldsymbol{o})}{lrd_{MinPts}(\boldsymbol{p})}}{|N_{MinPts}(\boldsymbol{p})|}.$$

It is the average of the ratio of the local reachability density of $\boldsymbol{p}$ and those of $\boldsymbol{p}$'s *MinPts*-nearest neighbors. It is easy to see that the lower $\boldsymbol{p}$'s local reachability density is, and the higher the local reachability density of $\boldsymbol{p}$'s *MinPts*-nearest neighbors are, the higher $LOF(\boldsymbol{p})$ is.

4.7.4 Deviation-Based Outlier Detection:

Deviation-based outlier detection does not use statistical tests or distance-basedmeasures to identify exceptional objects. Instead, it identifies outliers by examining themain characteristics of objects in a group.Objects that "deviate" fromthisdescription areconsidered outliers. Hence, in this approach the term deviations is typically used to referto outliers. In this section, we study two techniques for deviation-based outlier detection.The first sequentially compares objects in a set, while the second employs an OLAPdata cube approach.

Sequential Exception Technique:

The sequential exception technique simulates the way in which humans can distinguishunusual objects from among a series of supposedly like objects. It uses implicit redundancyof the data. Given a data set, D, of n objects, it builds a sequence of subsets,{D1, D2, …,Dm}, of these objects with 2<=m <= n such that

$$D_{j-1} \subset D_j, \quad \text{where } D_j \subseteq D.$$

Dissimilarities are assessed between subsets in the sequence. The technique introducesthe following key terms.

Exception set:

This is the set of deviations or outliers. It is defined as the smallestsubset of objects whose removal results in the greatest reduction of dissimilarity in the residual set.

Dissimilarity function:

This function does not require a metric distance between theobjects. It is any function that, if given a set of objects, returns a lowvalue if the objectsare similar to one another. The greater the dissimilarity among the objects, the higherthe value returned by the function. The dissimilarity of a subset is incrementally computedbased on the subset prior to it in the sequence. Given a subset of n numbers, $\{x_1, \ldots, x_n\}$, a possible dissimilarity function is the variance of the numbers in theset, that is,

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2,$$

where x is the mean of the n numbers in the set. For character strings, the dissimilarityfunction may be in the form of a pattern string (e.g., containing wildcard charactersthat is used to cover all of the patterns seen so far. The dissimilarity increases when the pattern covering all of the strings in D_{j-1} does not cover any string in D_j that isnot in D_{j-1}.

Cardinality function:

This is typically the count of the number of objects in a given set.

Smoothing factor:

This function is computed for each subset in the sequence. Itassesses how much the dissimilarity can be reduced by removing the subset from theoriginal set of objects.