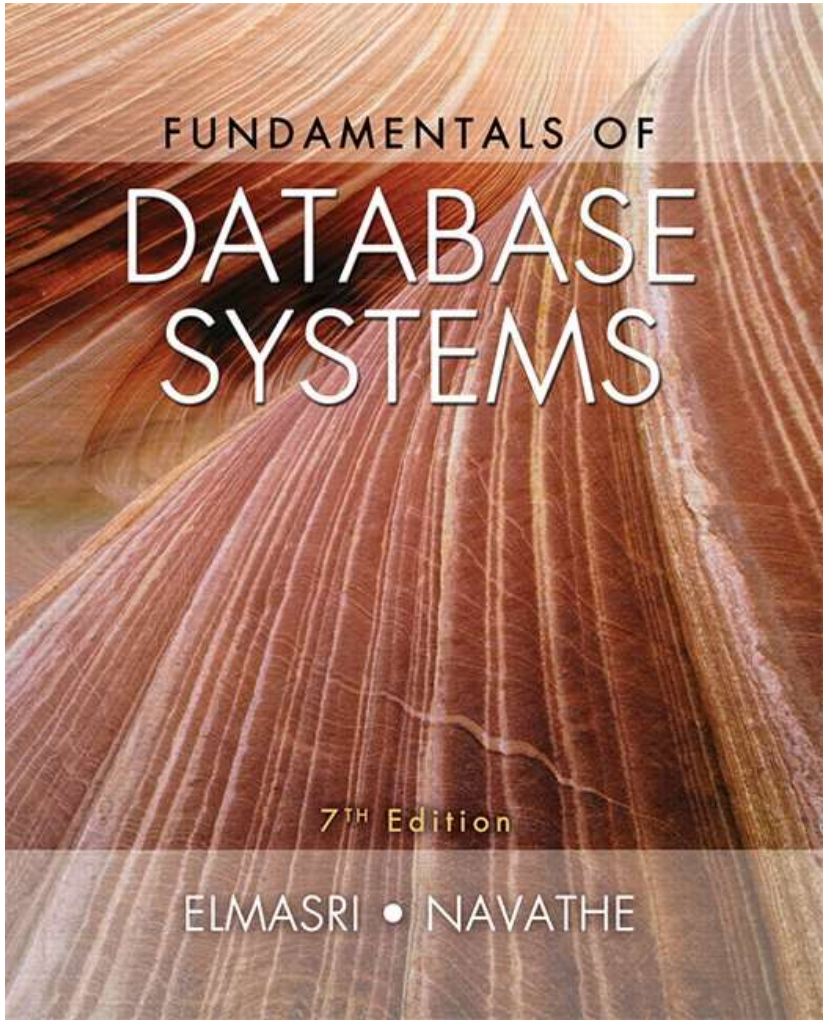


CHAPTER 28

Data Mining Concepts

Chapter Outline

- 1 Data Mining
  - 1.1. Data Mining vs. Data Warehousing
  - 1.2 Knowledge Discovery in Databases (KDD)
  - 1.3 Goals of Data Mining and Knowledge Discovery
  - 1.4 Types of Knowledge Discovered
- 2 Association Rules
  - 2.1 Association Rules Confidence and Support
  - 2.2 Apriori Algorithm
  - 2.3 Sampling Algorithm
  - 2.4 Frequent Pattern (FP) Tree and Growth Algorithms
  - 2.5 Partition Algorithm
  - 2.6 Other types of Association Rules
  - 2.7 Additional Considerations

Chapter Outline

- 3 Classification
- 4 Clustering
- 5 Other Data Mining Algorithms
  - 5.1 Sequential pattern discovery
  - 5.2 Time Series Analysis
  - 5.3 Regression
  - 5.4 Neural Networks
  - 5.5 Genetic Algorithms
- 6 Data Mining Applications
- 7 Commercial Data Mining Tools

1 Definitions of Data Mining

- The discovery of new information in terms of patterns or rules from vast amounts of data.
- The process of finding interesting structure in data.
- The process of employing one or more computer learning techniques to automatically analyze and extract knowledge from data.

1.1 Data Mining vs. Data Warehousing

- The data warehouse is a historical database designed for decision support.
- Data mining can be applied to the data in a warehouse to help with certain types of decisions.
- Proper construction of a data warehouse is fundamental to the successful use of data mining.
- Most enterprises (Fortune 500 companies) already have data warehouses. They are now trying very hard to benefit from that data with mining techniques.

1.2 Knowledge Discovery in Databases (KDD)

- Data mining is actually one step of a larger process known as **knowledge discovery in databases** (KDD).
- The KDD process model comprises six phases
  - Data selection
  - Data cleansing
  - Enrichment
  - Data transformation or encoding
  - Data mining
  - Reporting and displaying discovered knowledge

1.3 Goals of Data Mining and Knowledge Discovery (PICO)

- **Prediction**:
  - Determine how certain attributes will behave in the future.
- **Identification**:
  - Identify the existence of an item, event, or activity.
- **Classification**:
  - Partition data into classes or categories.
- **Optimization**:
  - Optimize the use of limited resources.

1.4 Types of Discovered Knowledge

- Association Rules
- Classification Hierarchies
- Sequential Patterns
- Patterns Within Time Series
- Clustering

2 Association Rules

- Association rules are frequently used to generate rules from **market-basket data**.
  - A market basket corresponds to the sets of items a consumer purchases during one visit to a supermarket.
- The set of items purchased by customers is known as an **itemset**.
- An **association rule** is of the form X=>Y, where X ={x_1, x_2, …., x_n }, and Y = {y_1,y_2, …., y_n} are sets of items, with x_i and y_i being distinct items for all i and all j.
  - For an association rule to be of interest, it must satisfy a minimum support and confidence.

2.1 Association Rules Confidence and Support

- **Support**:
  - The minimum percentage of instances in the database that contain all items listed in a given association rule.
  - Support is the percentage of transactions that contain all of the items in the itemset, LHS U RHS.
- **Confidence**:
  - Given a rule of the form A=>B, rule confidence is the conditional probability that B is true when A is known to be true.
  - Confidence can be computed as
    - support(LHS U RHS) / support(LHS)

2.2 Generating Association Rules and Apriori Algorithm

- The general algorithm for generating association rules is a two-step process.
  - Generate all itemsets that have a support exceeding the given threshold. Itemsets with this property are called **large** or **frequent itemsets**.
  - Generate rules for each itemset as follows:
    - For itemset X and Y a subset of X, let Z = X – Y;
    - If support(X)/Support(Z) > minimum confidence, the rule Z=>Y is a valid rule.

Reducing Association Rule Complexity

- Two properties are used to reduce the search space for association rule generation.
  - **Downward Closure**
    - A subset of a large itemset must also be large
  - **Anti-monotonicity**
    - A superset of a small itemset is also small. This implies that the itemset does not have sufficient support to be considered for rule generation.

The Apriori Algorithm (1)

- The **Apriori algorithm** was the first algorithm used to generate association rules.
  - The Apriori algorithm uses the general algorithm for creating association rules together with downward closure and anti-monotonicity.

The Apriori Algorithm (2)

- **Algorithm 28.1.** Apriori Algorithm for Finding Frequent (Large) Itemsets
- **Input:** Database of *m* transactions, *D*, and a minimum support, *mins*, represented as a fraction of *m.*
- **Output:** Frequent itemsets, L_1, L_2, ..., L_k
- **Begin** /* steps or statements are numbered for better readability */
- **1.** Compute support(i_j) = count(i_j)/*m* for each individual item, i_1, i_2, ..., i_n by scanning the database once and counting the number of transactions that item i_j appears in (that is, count(i_j));

The Apriori Algorithm (3)

- **2.** The candidate frequent 1-itemset, C_1, will be the set of items i_1, i_2, ..., i_n;
- **3.** The subset of items containing i_j from C_1 where support(i_j) >= mins becomes the frequent 1-itemset, L_1;
- **4.** k = 1;

  termination = false;
- **repeat**

  …..

  until termination (for this part see the next slide)

The Apriori Algorithm (4)

- **repeat**
- **1.** L_{k+1} = (empty set) ;
- **2.** Create the candidate frequent (k+1)-itemset, C_{k+1}, by combining members of L_k that have k–1 items in common (this forms candidate frequent (k+1)-itemsets by selectively extending frequent k-itemsets by one item);
- **3.** In addition, only consider as elements of C_{k+1} those k+1 items such that every subset of size k appears in L_k;
- **4.** Scan the database once and compute the support for each member of C_{k+1}; if the support for a member of C_{k+1} >= mins then add that member to L_{k+1};
- **5.** If L_{k+1} is empty then termination = true
  else $k = k + 1$;
  until termination;
- **End;**

2.3 The Sampling Algorithm

- The **sampling algorithm** selects samples from the database of transactions that individually fit into memory. Frequent itemsets are then formed for each sample.
  - If the frequent itemsets form a superset of the frequent itemsets for the entire database, then the real frequent itemsets can be obtained by scanning the remainder of the database.
- The negative border with respect to a frequent itemset, *S*, and set of items, *I*, is the minimal itemsets contained in PowerSet(*I*) and not in *S*.

2.3 The Sampling Algorithm (2)

- The negative border of a set of frequent itemsets contains the closest itemsets that could also be frequent.
- Consider the case where a set *X* is not contained in the frequent itemsets. If all subsets of *X* are contained in the set of frequent itemsets, then *X* would be in the negative border.
- If X in the negative border is in the frequent itemsets then there is potential for a superset of X to be in frequent itemsets. In such cases a second pass over the database is needed to make sure all frequent datasets are found.

2.4 Frequent-Pattern Tree Algorithm

- The **Frequent-Pattern Tree** Algorithm reduces the total number of candidate itemsets by producing a compressed version of the database in terms of an FP-tree.
- The FP-tree stores relevant information and allows for the efficient discovery of frequent itemsets.
- The algorithm consists of two steps:
  - Step 1 builds the FP-tree.
  - Step 2 uses the tree to find frequent itemsets.

Step 1: Building the FP-Tree

- First, frequent 1-itemsets along with the count of transactions containing each item are computed.
- The 1-itemsets are sorted in non-increasing order.
- The root of the FP-tree is created with a “null” label.
- For each transaction T in the database, place the frequent 1-itemsets in T in sorted order. Designate T as consisting of a head and the remaining items, the tail.
- Insert itemset information recursively into the FP-tree as follows:
  - if the current node, N, of the FP-tree has a child with an item name = head, increment the count associated with N by 1 else create a new node, N, with a count of 1, link N to its parent and link N with the item header table.
  - if tail is nonempty, repeat the above step using only the tail, i.e., the old head is removed and the new head is the first item from the tail and the remaining items become the new tail.

Step 2: The FP-growth Algorithm For Finding Frequent Itemsets

Algorithm 28.2 FP-Growth Algorithm Input: Fp-tree and minimum support, mins
Output: frequent patterns (itemsets)
procedure FP-growth (tree, alpha);
Begin
if tree contains a single path P then
for each combination, beta of the nodes in the path
generate pattern (beta ∪ alpha)
with support = minimum support of nodes in beta
else
for each item, i, in the header of the tree do
begin
generate pattern beta = (i ∪ alpha) with support = i.support;
construct beta's conditional pattern base;
construct beta's conditional FP-tree, beta_tree;
if beta_tree is not empty then
FP-growth(beta_tree, beta);
end;
End;

2.5 The Partition Algorithm

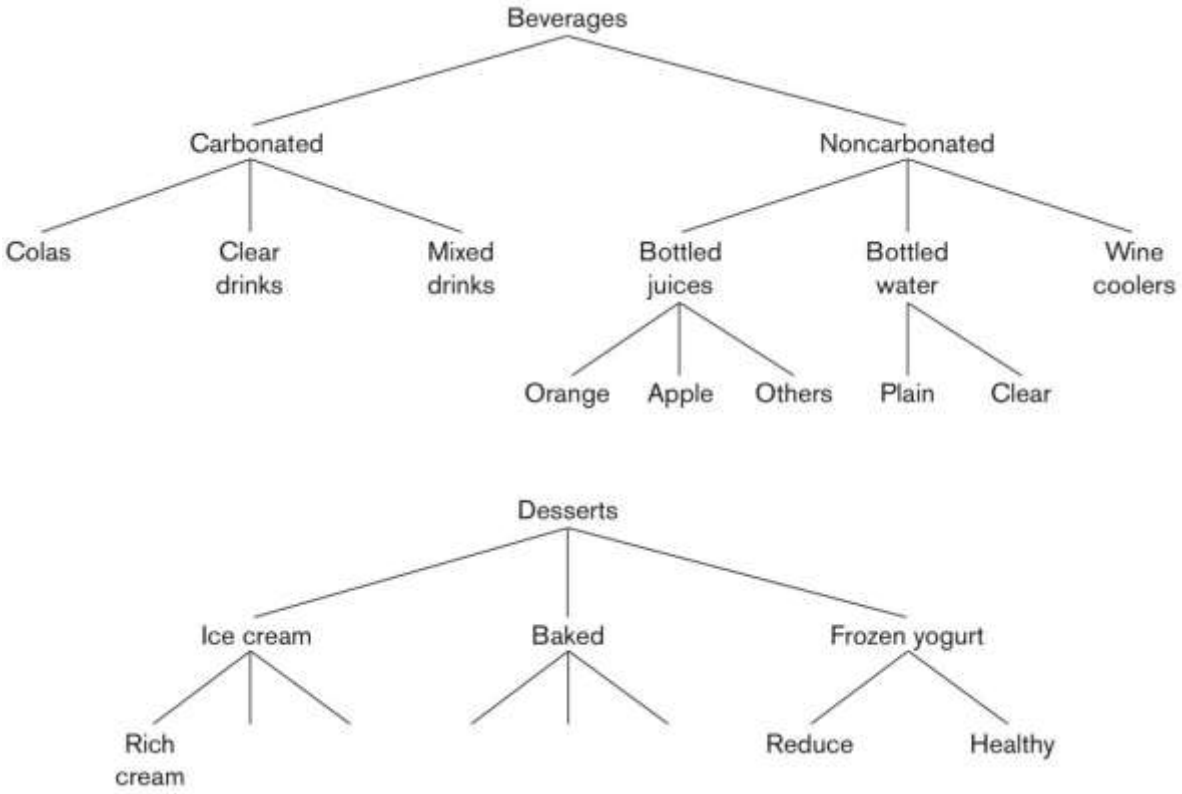
- Divide the database into non-overlapping subsets.
- Treat each subset as a separate database where each subset fits entirely into main memory.
- Apply the Apriori algorithm to each partition.
- Take the union of all frequent itemsets from each partition.
- These itemsets form the global candidate frequent itemsets for the entire database.
- Verify the global set of itemsets by having their actual support measured for the entire database.

Association Rules among hierarchies (1)

- Certain types of associations occur among hierarchies of items.
- For example, foods in a supermarket, items in a department store, or articles in a sports shop can be categorized into classes and subclasses that give rise to hierarchies.
- Figure 28.3 shows the taxonomy of items in a supermarket, with two hierarchies—beverages and desserts.

Association Rules among hierarchies (2)

- **Figure 28.3** Taxonomy of items in a supermarket.



Association Rules among hierarchies (3)

- The entire groups may not produce associations of the form
  - beverages => desserts, or desserts => beverages.
- However, associations of the type

  Healthy-brand frozen yogurt => bottled water, or
  Rich cream-brand ice cream => wine cooler

may produce enough confidence and support to be valid association rules.

Multi-dimensional Association Rules

- The following rule is an example of including the label of the single dimension:
  - Items_bought(milk) => Items_bought(juice).
- It may be of interest to find association rules that involve multiple dimensions, for example,
  - Time(6:30...8:00) => Items_bought(milk).
- Rules like these are called *multidimensional association rules*. The dimensions represent attributes of records of a file or, in terms of relations, columns of rows of a relation, and can be categorical or quantitative

Multi-dimensional and Negative Association Rules

- One approach to handling a quantitative attribute is to partition its values into non-overlapping intervals.
- E.g., consider the salary attribute:
  - (0 < Salary < 29,999) = > {Kia, Hyundai}
  - (30,000 < salary < 99,999) => {Honda, Toyota}
  - (100,000 < salary < 149,000) => {Lexus, Mercedes}
  - (150,000 < salary < 300,000) => {Porsche, BMW}
- Negative Association Rules: E.g.. those who buy

Soft_drink “Wakeup” are not likely to buy chips “Nightos”. It is very difficult to discover these because support is very low by definition. See the discussion in the text and

Savasere, Omiecinski and Navathe, “Mining Strong Negative Association in a Large Database of Customer Transactions,” Int. Conf. on Data Mining, 1998.

Complications seen with Association Rules

- The cardinality of itemsets in most situations is extremely large.
- Association rule mining is more difficult when transactions show variability in factors such as geographic location and seasons.
- Item classifications exist along multiple dimensions. Those must be taken into account.
- Data quality is variable; data may be missing, erroneous, conflicting, as well as redundant.

3. Classification

- **Classification** is the process of learning a model that is able to describe different classes of data.
- Learning is **supervised** as the classes to be learned are predetermined.
- Learning is accomplished by using a training set of pre-classified data.
- The model produced is usually in the form of a decision tree or a set of rules.

Decision Tree Construction (1)

- **Algorithm 28.3.** Algorithm for Decision Tree Induction
- **Input:** Set of training data records: R_1, R_2, ..., R_m and set of attributes: A_1, A_2, ..., A_n
- **Output:** Decision tree
- procedure Build_tree (records, attributes);
- **Begin**

    create a node *N*;

    if all records belong to the same class, *C* then

        return *N* as a leaf node with class label *C*;

    if attributes is empty then return *N* as a leaf node with class label *C*, such that the majority of records belong to it;

    select attribute A_i (*with the highest information gain*) from attributes;

    label node *N* with A_i;

Decision Tree Construction (2)

- for each known value, v_j, of A_i do

begin

add a branch from node N for the condition $A_i = v_j$;

S_j = subset of records where $A_i = v_j$;

if S_j is empty then

add a leaf, L, with class label C, such that the majority of records belong to it and return L

else add the node returned by Build_tree(S_j, attributes – A_i);

end;

End;

-

Figure 28.7 Decision tree based on sample training data

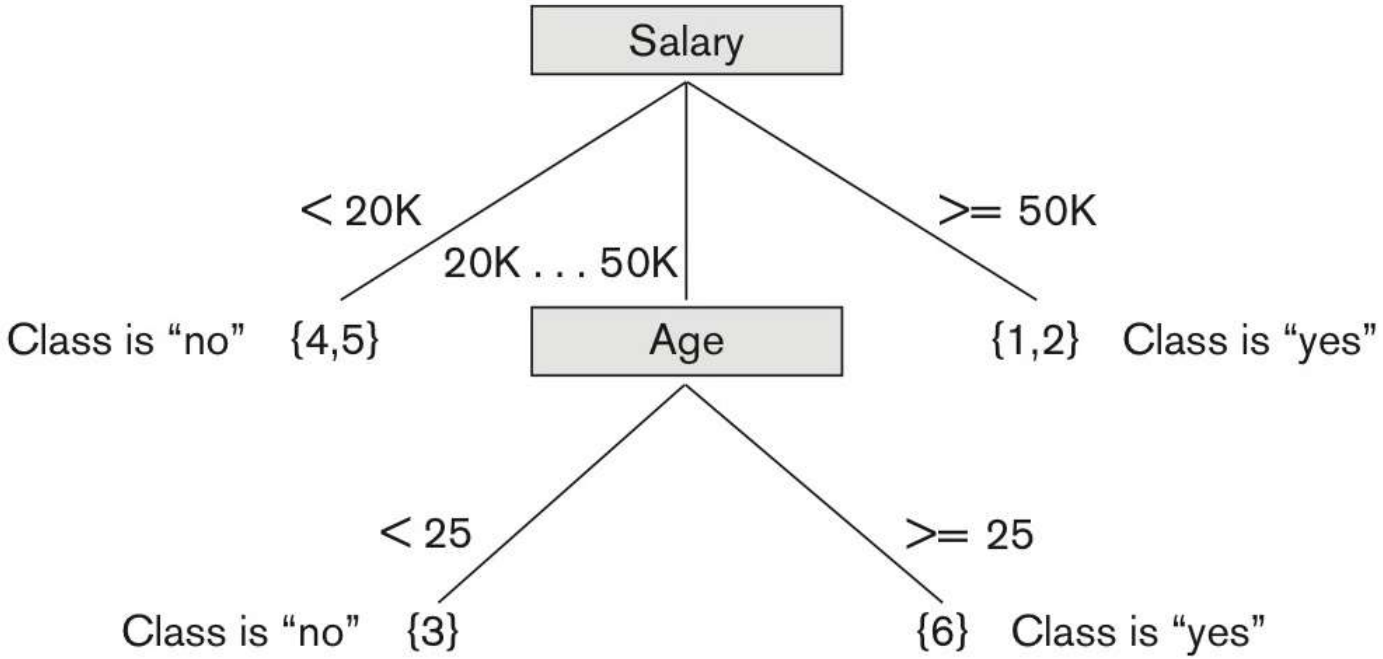


Figure 28.7 Decision tree based on sample training data where the leaf nodes are represented by a set of RIDs of the partitioned records.

An Example Rule

- Here is one of the rules extracted from the decision tree of Figure 28.7.

  IF 50K > salary >= 20K

  AND age >=25

  THEN class is “yes”

- Another rule

  IF 50K > salary >= 20K

  AND age < 25

  THEN class is “no”

4 Clustering

- **Unsupervised** learning or clustering builds models from data without predefined classes.
- The goal is to place records into groups where the records in a group are highly similar to each other and dissimilar to records in other groups.
- The **k-Means** algorithm is a simple yet effective clustering technique.

K-Means Clustering

- **Algorithm 28.4.** *k*-Means Clustering Algorithm
- **Input:** a database *D*, of *m* records, r_1, ..., r_m and a desired number of clusters *k*
- **Output:** set of *k* clusters that minimizes the squared error criterion
- **Begin**

  randomly choose *k* records as the centroids for the *k* clusters;

  repeat

  assign each record, r_i, to a cluster such that the distance between r_i and the cluster centroid (mean) is the smallest among the *k* clusters;

  recalculate the centroid (mean) for each cluster based on the records assigned to the cluster;

  until no change;
- **End;**

5 Additional Approaches for Data Mining

- **Sequential pattern analysis**
  - Looks for a **sequence of itemsets**
  - Transactions ordered by time of purchase form a sequence of itemsets.
  - The **support** for a sequence *S* of itemsets is the percentage of the given set *U* of sequences of which *S* is a subsequence
  - Assuming we are analyzing sequences of items as a time series for a customer, The sequence S_1, S_2, S_3, ... is a **predictor** of the fact that a customer who buys itemset S_1 is likely to buy itemset S_2 and then S_3, and so on.

5.2 Time Series Analysis

- **Time Series analysis**
  - **Time series** are sequences of events; each event may be a given fixed type of a transaction.
    - For example, the closing price of a stock or a fund is an event that occurs every weekday for each stock and fund. The sequence of these values per stock or fund constitutes a time series.
    - For a time series, one may look for a variety of patterns by analyzing sequences and subsequences as we did above
    - Time series may be compared by establishing measures of similarity to identify companies whose stocks behave in a similar fashion
- Time series analysis is an extended functionality of **temporal** data management.

5.3 Regression (1)

- **Regression**

Regression is a special application of the classification rule.

- If a classification rule is regarded as a function over the variables that maps these variables into a target class variable, the rule is called a **regression rule**.
- A **regression equation** (or function) estimates a **dependent** variable using a set of **independent** variables and a set of constants.
  - For example, consider a relation

    LAB_TESTS (patient ID, test 1, test 2, ..., test *n*)

    which contains values that are results from a series of *n* tests for one patient.
- The target variable that we wish to predict is *P*, the probability of survival of the patient. Then the rule for regression takes the form:

Regression (2)

- **Regression**
- (test 1 in range$_1$) and (test 2 in range$_2$) and ... (test n in range$_n$) $\Rightarrow$ P = x, or $x < P <= y$
- In the above example, we regard P as a function:

  $P = f$ (test 1, test 2, ..., test n)
- If the function appears as

  $Y = f(x_1, x_2, ..., x_n)$,

  and f is linear in the domain variables x_i, the process of deriving f from a given set of tuples for $<x_1, x_2, ..., x_n, y>$ is called **linear regression**.
- Linear regression is a commonly used statistical technique for fitting a set of observations or points in n dimensions with the target variable y.

5.4 Neural Networks

- A **Neural Network** is a set of interconnected nodes designed to imitate the functioning of the brain.
- **Node connections** have **weights** which are modified during the learning process.
- Neural networks are of two types: supervised and unsupervised.
- Adaptive methods that attempt to reduce the output error are **supervised learning** methods
- Those that develop internal representations without sample outputs are called **unsupervised learning** methods.
- The output of a supervised neural network is **quantitative** and not easily understood.
- Unsupervised networks learn; but they do not provide a good representation of what they learn
- Although neural networks suffer from above shortcomings and cannot easily deal with time series data, they are still popular.

5.5 Genetic Algorithms (1)

- **Genetic algorithms** (GAs) are a class of randomized search procedures capable of adaptive and robust search over a wide range of search space topologies.
- Modeled after the adaptive emergence of biological species from evolutionary mechanisms, **Genetic learning** is based on the theory of evolution.
- An initial population of several candidate solutions is provided to the learning model.
- A fitness function defines which solutions survive from one generation to the next.
- **Crossover, mutation** and **selection** are used to create new population elements.

Genetic Algorithms (2)

- The solutions produced by GAs are distinguished from most other search techniques by the following characteristics:
  - A GA search uses a set of solutions during each generation rather than a single solution.
  - The search in the string-space represents a much larger parallel search in the space of encoded solutions.
  - The memory of the search done is represented solely by the set of solutions available for a generation.
  - A genetic algorithm is a randomized algorithm since search mechanisms use probabilistic operators.
  - While progressing from one generation to the next, a GA finds near-optimal balance between knowledge acquisition and exploitation by manipulating encoded solutions.

6 Data Mining / Business Analytics (B.I.) Applications

- **Marketing**
  - Marketing strategies and consumer behavior. Advertising, Service location, Customer Segmentation, Targeted mailing
- **Finance**
  - Fraud detection, creditworthiness and investment performance analysis
- **Manufacturing**
  - Resource optimization of men, machines and materials
- **Health**
  - Image analysis, genetic analysis, side effects of drug, and treatment effectiveness, process and cost optimization in healthcare services

7 Data Mining Tools

- Currently, commercial data mining tools use several common techniques to extract knowledge. These include association rules, clustering, neural networks, sequencing, and statistical analysis.
- Also used are decision trees, which are a representation of the rules used in classification or clustering, and statistical analyses, which may include different types of regression and many other techniques.
- Some commercial products use advanced techniques such as genetic algorithms, case-based reasoning, Bayesian networks, nonlinear regression, combinatorial optimization, pattern matching, and fuzzy logic.
- A set of representative tools are presented in Table 28.2

Some Commercial Data Mining Tools

Table 28.1 Some Representative Data Mining Tools

Company	Product	Technique	Platform	Interface*
IBM	Intelligent Miner	Classification, association rules, predictive models	UNIX (AIX)	IBM DB2
Megaputer Intelligence	PolyAnalyst	Symbolic knowledge acquisition, evolutionary programming	Windows OS/2	ODBC Oracle DB2
NCR	Management Discovery Tool (MDT)	Association rules	Windows	ODBC
Purple Insight	MineSet	Decision trees, association rules	UNIX (Irix)	Oracle Sybase Informix
SAS	Enterprise Miner	Decision trees, association rules, Nneural nets, regression, clustering	UNIX (Solaris) Windows Macintosh	ODBC Oracle AS/400

*ODBC: Open Database Connectivity
ODMG: Object Data Management Group

Future of Data Mining

- Data mining tools are continually evolving, building on ideas from the latest scientific research.
  - Many of these tools incorporate the latest algorithms taken from artificial intelligence (AI), machine learning (which is a very parallel term), statistics, and optimization.
- The primary direction for data mining is to analyze terabytes and petabytes of data in the so-called Big Data systems which we presented in Chapter 25.
- Big Data systems are being equipped with their own tools and libraries for data mining such as Mahout which runs on top of Hadoop.
- The data mining area will also be closely tied to data that will be housed in the cloud in data warehouses and brought into service for mining operations as needed using OLAP (Online Analytical Processing) servers.
- Business Analytics is the primary goal of the currently advancing Spark technology.

Recap

- Data Mining
- Data Warehousing
- Knowledge Discovery in Databases (KDD)
- Goals of Data Mining and Knowledge Discovery
- Association Rules
- Classification
- Clusering
- Additional Data Mining Algorithms
  - Sequential pattern analysis
  - Time Series Analysis
  - Regression
  - Neural Networks
  - Genetic Algorithms
- Data Mining Applications
- Commercial Data Mining Tools and Future Directions.