UNIT 1

Business intelligence

Business intelligence may be defined as a set of mathematical models and analysis methodologies that exploit the available data to generate information and knowledge useful for complex decision-making processes.
Loads of heterogeneous data available everywhere. It is possible to convert such data into information and knowledge that can then be used by decision makers. Business intelligence may be defined as a set of mathematical models and analysis methodologies that exploit the available data to generate information and knowledge useful for complex decision-making processes. The main purpose of business intelligence systems is to provide knowledge workers with tools and methodologies that allow them to make effective and timely decisions.

Effective and timely decisions

The main purpose of business intelligence systems is to provide knowledge workers with tools and methodologies that allow them to make **effective** and **timely** decisions.

Effective decisions

The application of rigorous analytical methods allows decision makers to rely on information and knowledge which are more dependable. As a result, they are able to make better decisions and devise action plans that allow their objectives to be reached in a more effective way. Indeed, turning to formal analytical methods forces decision makers to explicitly describe both the criteria for evaluating alternative choices and the mechanisms regulating the problem under investigation. Furthermore, the ensuing in-depth examination and thought lead to a deeper awareness and comprehension of the underlying logic of the decision-making process.

Timely decisions

Enterprises operate in economic environments characterized by growing levels of competition and high dynamism. As a consequence, the ability to rapidly react to the actions of competitors and to new market conditions is a critical factor in the success or even the survival of a company.
The major benefits that a given organization may draw from the adoption of a business intelligence system. When facing problemssuch as those described in Examples 1.1 and 1.2 above, decision makers ask themselves a series of questions and develop the corresponding analysis. Hence, they examine and compare several options, selecting among them the best decision, given the conditions at hand.
If decision makers can rely on a business intelligence system facilitating their activity, we can expect that the overall quality of the decision-making process will be greatly improved. With the help of mathematical models and algorithms, it is actually possible to analyze a larger number of alternative actions, achieve more accurate conclusions and reach effective and timely decisions.

We may therefore conclude that the major advantage deriving from theadoption of a business intelligence system is found in the increased effectivenessof the decision-making process.

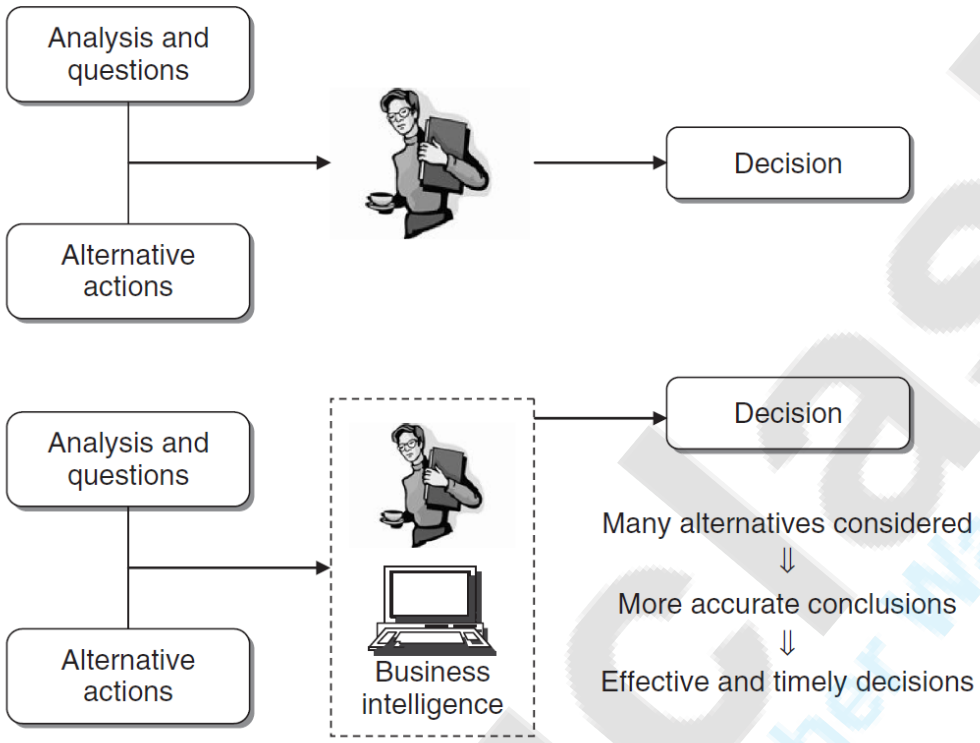


Figure 1.1 Benefits of a business intelligence system

Data, information and knowledge

Data

Generally, data represent a structured codification of single primary entities, as well as of transactions involving two or more primary entities. For example, for a retailer data refer to primary entities such as customers, points of sale and items, while sales receipts represent the commercial transactions.

Information

Information is the outcome of extraction and processing activities carried out on data, and it appears meaningful for those who receive it in a specific domain. For example, to the sales manager of a retail company, the proportion of sales receipts in the amount of over 100 per week, or the number of customers holding a loyalty card who have reduced by more than 50% the monthly amount spent in the last three months, represent meaningful pieces of information that can be extracted from raw stored data.

Knowledge

Information is transformed into knowledge when it is used to make decisions and develop the corresponding actions. Therefore, we can think of knowledge as consisting of information put to work into a specific domain, enhanced by the experience and competence of decision makers in tackling and solving complex problems. For a retail company, a sales analysis may detect that a group of customers, living in an area where a competitor has recently opened a new point of sale, have reduced their usual amount of business.
The knowledge extracted in this way will eventually lead to actions aimed at solving the problem detected, for example by introducing a new free home delivery service for the customers residing in that specific area. We wish to point out that knowledge can be extracted from data both in a *passive* way, through the analysis criteria suggested by the decision makers, or through the *active* application of mathematical models, in the form of inductive learning or optimization,
Several public and private enterprises and organizations have developed in recent years formal and systematic mechanisms to gather,store and share their wealth of knowledge, which is now perceived as an invaluable intangible asset.
The activity of providing support to knowledge workers through the integration of decision-making processes and enabling information technologies is usually referred to as **knowledge management**.
It is apparent that business intelligence and knowledge management share some degree of similarity in their objectives. The main purpose of both disciplines is to develop environments that can support knowledge workers in decision-making processes and complex problem-solving activities. To draw a boundary between the two approaches, we may observe that knowledge management methodologies primarily focus on the treatment of information that is usually unstructured, at times implicit, contained mostly in documents, conversations and past experience. Conversely, business intelligence systems are based on structured information, most often of a quantitative nature and usually organized in a database. However, this distinction is a somewhat fuzzy one: for example, the ability to analyze emails and web pages through text mining methods progressively induces business intelligence systems to deal with unstructured information.

Business intelligence architectures

The architecture of a business intelligence system, includes three major components.

Data sources

In a first stage, it is necessary to gather and integrate the data stored in the various primary and secondary sources, which are heterogeneous in origin and type. The sources consist for the most part of data belonging to operational systems, but may also include unstructured documents, such as emails and data received from external providers. Generally speaking, a major effort is required to unify and integrate the different data sources

Data warehouses and data marts

Using extraction and transformation tools known as *extract, transform, load* (ETL), the data originating from the different sources are stored in databases intended to support business intelligence analyses. These databases are usually referred to as *data warehouses* and **data marts**

Business intelligence methodologies

Data are finally extracted and used to feed mathematical models and analysis methodologies intended to support decision makers. In a business intelligence system, several decision support applications may be implemented, most of which will be described in the following chapters:

- Multidimensional cube analysis;
- Exploratory data analysis;
- Time series analysis;
- Inductive learning models for data mining;
- Optimization models.

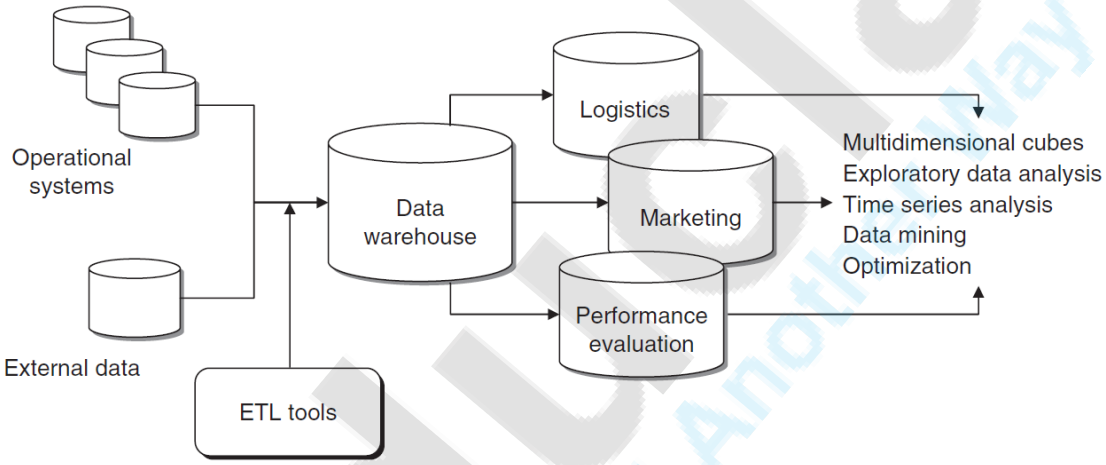


Figure 1.2 A typical business intelligence architecture

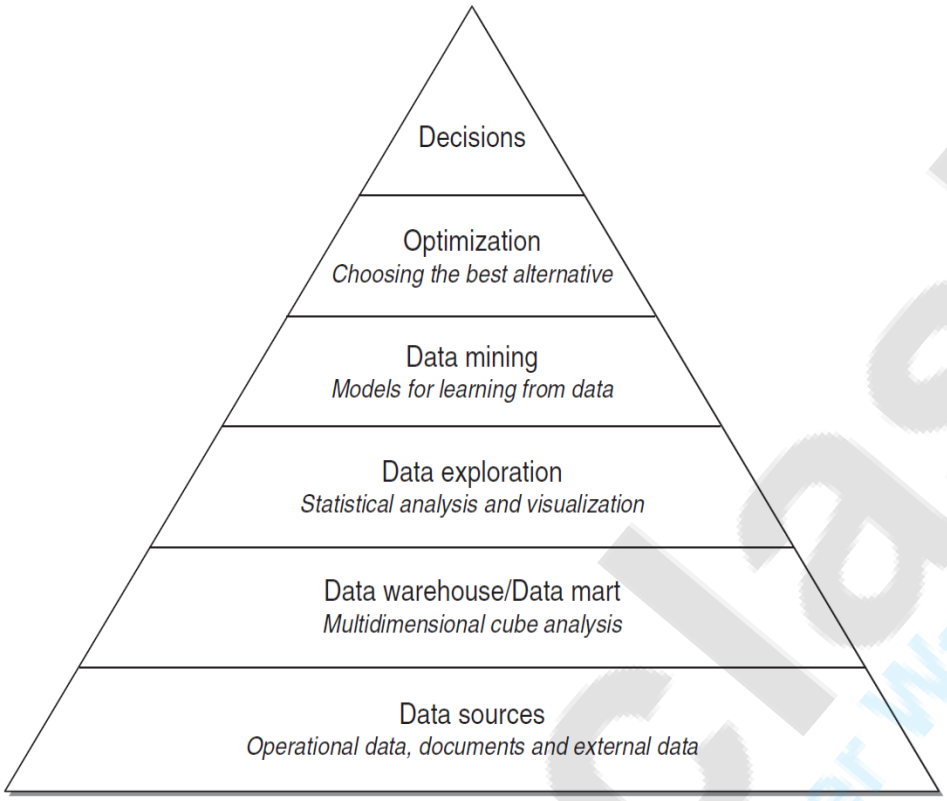


Figure 1.3 The main components of a business intelligence system

Data exploration

At the third level of the pyramid we find the tools for performing a *passive* business intelligence analysis, which consist of query and reporting systems, as well as statistical methods. These are referred to as passive methodologies because decision makers are requested to generate prior hypotheses or define data extraction criteria, and then use the analysis tools to find answers and confirm their original insight. For instance, consider the sales manager of a company who notices that revenues in a given geographic area have dropped for a specific group of customers. Hence, she might want to bear out her hypothesis by using extraction and visualization tools, and then apply a statistical test to verify that her conclusions are adequately supported by data.

Data mining

The fourth level includes *active* business intelligence methodologies, whose purpose is the extraction of information and knowledge from data. These include mathematical models for pattern recognition, machine learning and data mining techniques, which will be dealt with in Part II of this book. Unlike the tools described at the previous level of the pyramid, the models of an active kind do not require decision makers to formulate any prior hypothesis to be later verified. Their purpose is instead to expand the decision makers' knowledge.

Optimization
By moving up one level in the pyramid we find optimization models that allow us to determine the best solution out of a set of alternative actions, which is usually fairly extensive and sometimes even infinite.

Decisions
Finally, the top of the pyramid corresponds to the choice and the actual adoption of a specific decision, and in some way represents the natural conclusion of the decision-making process. Even when business intelligence methodologies are available and successfully adopted, the choice of a decision pertains to the decision makers, who may also take advantage of informal and unstructured information available to adapt and modify the recommendations and the conclusions achieved through the use of mathematical models.

Ethics and business intelligence

The adoption of business intelligence methodologies, data mining methods and decision support systems raises some ethical problems that should not be overlooked. Indeed, the progress toward the information and knowledge society opens up countless opportunities, but may also generate distortions and risks which should be prevented and avoided by using adequate control rules and mechanisms. Usage of data by public and private organizations that is improper and does not respect the individuals' right to privacy should not be tolerated.

More generally, we must guard against the excessive growth of the political and economic power of enterprises allowing the transformation processes outlined above to exclusively and unilaterally benefit such enterprises themselves, at the expense of consumers, workers and inhabitants of the Earth ecosystem.

However, even failing specific regulations that would prevent the abuse of data gathering and invasive investigations, it is essential that business intelligence analysts and decision makers abide by the ethical principle of respect for the personal rights of the individuals. The risk of overstepping the boundary between correct and intrusive use of information is particularly high within the relational marketing and web mining fields, Respect for the right to privacy is not the only ethical issue concerning the use of business intelligence systems.
There has been much discussion in recent years of the social responsibilities of enterprises, leading to the introduction of the new concept of *stakeholders*. This term refers to anyone with any interest in the activities of a given enterprise, such as investors, employees, labor unions and civil society as a whole. There is a diversity of opinion on whether a company should pursue the short-term maximization of profits, acting exclusively in the interest of shareholders, or should instead adopt an approach that takes into account the social consequences of its decisions.
For example, is it right to develop an optimization model with the purpose of distributing costs on an international scale in order to circumvent the tax systems of certain countries? Is it legitimate to make a decision on the optimal position of the tank in a vehicle in order to minimize production costs, even if this may cause serious harm to the passengers in the event

of a collision? As proven by these examples, analysts developing a mathematical model and those who make the decisions cannot remain neutral, but have the moral obligation to take an ethical stance.

The Balanced Scorecard and the Business-Centric BI Architecture

Sometimes when an organization begins a business intelligence initiative (BI) they are so excited about data visualization and data transparency in the form of dashboards that the first thing they want to do is start measuring everything. I believe that strategy comes before measures and those organizations that thoughtfully and purposefully align what they are measuring to their strategic plan achieve more meaningful long-term results from their BI initiative.

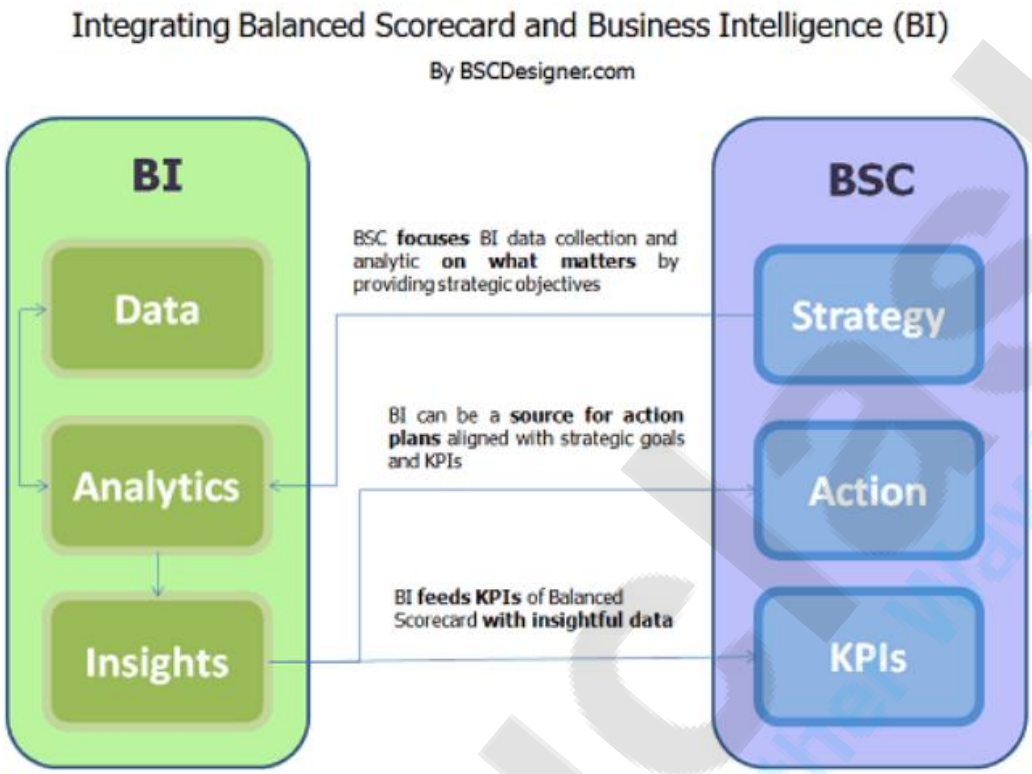
The Balanced Scorecard is a performance management system designed to align, measure, and communicate how well an organization's activities are supporting the strategic vision and mission of the organization.
It was originated by Drs. Robert Kaplan (Harvard Business School) and David Norton as a performance measurement framework that added strategic non-financial performance measures to traditional financial metrics to create a more 'balanced' view of organizational performance. Four strategic perspectives are addressed within the Balanced Scorecard framework:

1. Customer
2. Financial
3. Internal Processes – commonly includes technology, systems, etc.
4. Learning and Growth (aka "Organization Capacity") – commonly includes people, training, etc.

Objectives (goals) are set for each perspective, measures (numbers) that represent things to be measured (such as sales, customers, returns) are identified and can then be transformed into ratios or counts, which serve as Key Performance Indicators (KPIs). Initiatives (projects) are undertaken in order to "move the needle" in a positive direction on the KPI gage for that measure.

Balanced Scorecard dashboards include both leading and lagging indicators. For example, customer and financial KPIs are traditionally lagging indicators – the numbers indicate *what has already happened.* KPIs for the two perspectives of internal processes and learning/growth are leading indicators. This is because positive results achieved with respect to internal processes and learning/growth initiatives should lead to a positive result in the customer and financial KPIs.

Types of consumer segmentation



Market segmentation

Market segmentation is the process of dividing a broad consumer or business market, normally consisting of existing and potential customers, into sub-groups of consumers (known as *segments*) based on some type of shared characteristics. In dividing or segmenting markets, researchers typically look for common characteristics such as shared needs, common interests, similar lifestyles or even similar demographic profiles. The overall aim of segmentation is to identify *high yield segments* – that is, those segments that are likely to be the most profitable or that have growth potential – so that these can be selected for special attention (i.e. become target markets).

Basis of Market Segmentation

Segmenting is dividing a group into subgroups according to some set 'basis'. These bases range from age, gender, etc. to psychographic factors like attitude, interest, values, etc.

Gender

Gender is one of the most simple yet important basis of market segmentation. The interests, needs and wants of males and females differ at many levels. Thus, marketers focus on different marketing and communication strategies for both. This type of segmentation is usually seen in the case of cosmetics, clothing, and jewellery industry, etc.

Age group

Segmenting market according to the age group of the audience is a great strategy for personalized marketing. Most of the products in the market are not universal to be used by all the age groups. Hence, by segmenting the market according to the target age group, marketers create better marketing and communication strategies and get better conversion rates.

Income

Income decides the purchasing power of the target audience. It is also one of the key factors to decide whether to market the product as a need, want or a luxury. Marketers usually segment the market into three different groups considering their income. These are

- High Income Group
- Mid Income Group
- Low Income Group

This division also varies according to the product, its use, and the area the business is operating in.

Place

The place where the target audience lives affects the buying decision the most. A person living on mountains will have less or no demand for ice cream than the person living in a desert.

Occupation

Occupation, just like income, influences the purchase decision of the audience. A need of an entrepreneur might be a luxury for a government sector employee. There are even many products which cater to an audience engaged in a specific occupation.

Usage

Product usage also acts as a segmenting basis. A user can be labelled as heavy, medium or light user of a product. The audience can also be segmented on the basis of their awareness of the product.

Lifestyle

Other than physical factors, marketers also segment the market on the basis of lifestyle. Lifestyle includes subsets like marital status, interests, hobbies, religion, values, and other psychographic factors which affect the decision making of an individual.

Types of Market Segmentation

Geographic Segmentation

Geographic segmentation divides the market on the basis of geography. This type of market segmentation is important for the marketers as people belonging to different regions may have different requirements. For example, water might be scarce in some regions which inflates the demand for bottled water but, at the same time, it might be in abundance in other regions where the demand for the same is very less.

People belonging to different regions may have different reasons to use the same product as well. Geographic segmentation helps marketer draft personalized marketing campaigns for everyone.

Demographic Segmentation

Demographic segmentation divides the market on the basis of demographic variables like age, gender, marital status, family size, income, religion, race, occupation, nationality, etc. This is one of the most common segmentation practice among the marketers. Demographic segmentation is seen almost in every industry like automobiles, beauty products, mobile phones, apparels, etc and is set on a premise that the customers' buying behaviour is hugely influenced by their demographics.

Behavioral Segmentation

The market is also segmented based on audience's behaviour, usage, preference, choices and decision making. The segments are usually divided based on their knowledge of the product and usage of the product. It is believed that the knowledge of the product and its use affects the buying decision of an individual. The audience can be segmented into –

- Those who know about the product,
- Those who don't know about the product,
- Ex-users,
- Potential users,
- Current Users,
- First time users, etc.

People can be labelled as brand loyal, brand-neutral, or competitor loyal. They can also be labelled according to their usage. For example, a sports person may prefer an energy drink as elementary (heavy user) and a not so sporty person may buy it just because he likes the taste (light/medium user).

Psychographic Segmentation

Psychographic Segmentation divides the audience on the basis of their personality, lifestyle and attitude. This segmentation process works on a premise that consumer buying behaviour can be influenced by his personality and lifestyle. Personality is the combination of characteristics that form an individual's distinctive character and includes habits, traits, attitude, temperament, etc. Lifestyle is how a person lives his life.

Personality and lifestyle influence the buying decision and habits of a person to a great extent. A person having a lavish lifestyle may consider having an air conditioner in every room as a need, whereas a person living in the same city but having a conservative lifestyle may consider it as a luxury.

Examples of market segmentation

Market segmentation is a common practice among all the industries. It is not possible for a marketer to address the mass with same marketing strategy. Here are some examples of market segmentation to prove this point.

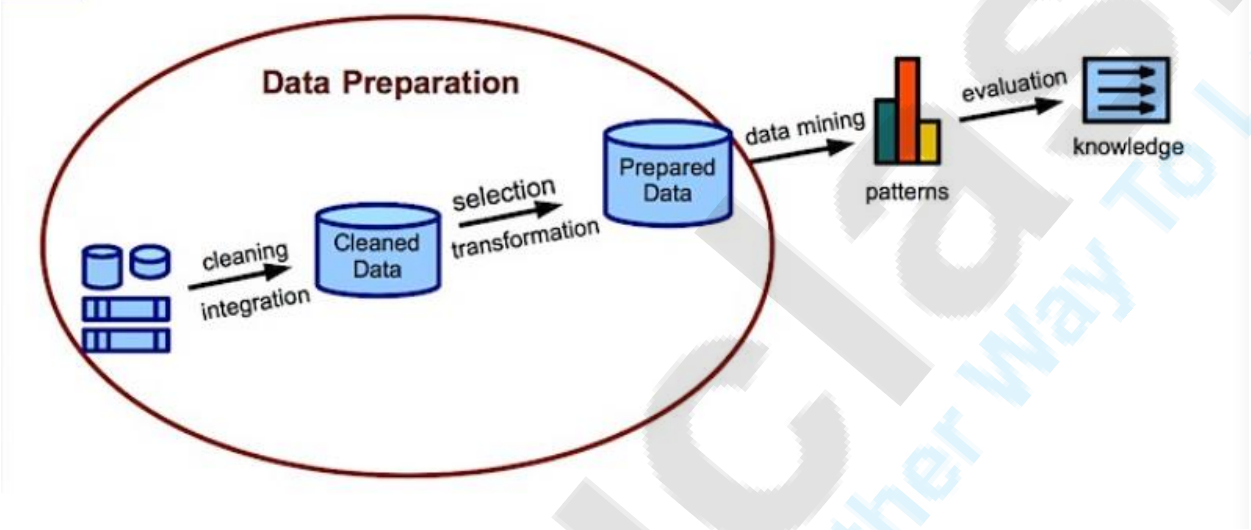
- Marketers will only waste their time and might end up making fun of themselves if they don't segment the market while marketing beauty products.
- A company that sells nutritious food might market the product to the older people while fast-food chains target the working demographic or teens.
- Sports brands often segment the market based on the sports they play which help them market the sports specific products to the right audience.

Market Segmentation is a convenient method marketers use to cut costs and boost their conversions. It allows them to be specific in their planning and thus provide better results. It ultimately helps them to target the niche user base by making smaller segments.

UNIT 2

What is Data Preparation?

Data preparation (or data preprocessing) in this context means manipulation of data into a form suitable for further analysis and processing. It is a process that involves many different tasks and which cannot be fully automated. Many of the data preparation activities are routine, tedious, and time consuming. It has been estimated that data preparation accounts for 60%-80% of the time spent on a data mining project.



Data preparation is essential for successful data mining. Poor quality data typically result in incorrect and unreliable data mining results.

Data preparation improves the quality of data and consequently helps improve the quality of data mining results. The well known saying "garbage-in garbage-out" is very relevant to this domain.

Our aim is to develop tools to facilitate the tasks of data preparation.

Prediction

Prediction attempts to form patterns that permit it to predict the next event(s) given the available input data.

- Objectives of prediction
- Anticipate inhabitant actions
- Detect unusual occurrences (anomalies)
- Predict the right course of actions

- Provide information for decision making

Different Prediction Methods

After the data are prepared, we can begin our search for the right prediction method. The goal is to build a prediction model that will predict the "outcome" of a new case.

- Mathematical (e. g., linear regression, statistical methods).
- Distance (e. g., instance-based learning, clustering).
- Logic (e. g., decision tables, decision trees, classification rules).
- Modern heuristic (e. g., neural networks, evolutionary algorithms, fuzzy logic).

Mathematical Methods

Linear regression:

The most popular explanatory method is **linear regression**. If the predicted outcome is numeric and all the variables in the prediction model are numeric, then linear regression is the classic choice.

In this method, we build a linear expression that uses the values of different variables to produce a predicted value for a "new" variable (i. e., a variable not used in the model).

Let us consider linear regression for predicting the auction price of a car. In this case, the "new" variable would be the predicted sale price. Note that many variables are not numeric, so we have to address this issue first.

It is clear that the non-numeric variables "make," "model," and "location" are of key importance, as they determine the basic price range (which is further influenced by the mileage, year, trim, etc.). By building a separate regression model for each make/model at each location, we can eliminate these three non-numeric variables.

Convert the remaining non-numeric variables into numeric variables we can take a list of the available colors, sort them from white to black according to some standard order (e. g., how they appear on a spectrum), and assign consecutive natural numbers. Assuming white would be 1 and black would be 30.Variables "mileage," "year, "and "damage level" are already numeric, so there is no need to covert these.

Because a linear regression model must answer (i. e., produce a value for) questions such as: "What's the price of a Toyota ("make") Camry ("model") at auction site Jacksonville, Florida ("location")?"

We need to develop a function:

Sale Price = a + (b × Mileage) + (c × Year) + (d × Color) + … that provides the predicted price for a new case (i. e., a used Toyota Camry) when supplied with the numeric values of the other variables ("mileage," "year," "color," etc.).

The main challenge here is to find the values for parameters *a, b, c, d,* etc. that give the prediction model the best possible performance (i. e., that minimize the predictive error). Since we have all the historic data from three million cases, we can extract all cases where "location" = Jacksonville, "make" =Toyota, and "model" = Camry. This subset of cases (say we identified 150 such cases) would constitute the data set available for training the prediction model (some of these cases would also be used for validation and testing.

To minimize the error on the training set, there are several standard procedures for determining the parameter values. Once these parameters are determined, the prediction model (for all Toyota Camry cars sold at the Jacksonville auction) is ready.

For every new case (again, by new case we mean a used Toyota Camry), we can determine the sale price for the Jacksonville location by inserting the appropriate values for "mileage," "year," "color," etc. into the sale price function.

Training process might not be that simple

First of all, some values might be missing (e. g., the mileage was not recorded). In such cases we can Remove the case from consideration and contact the appropriate auction site to recover the mileage value. Once this value is recovered, we can insert the case back into the system for processing. Although this would cause a delay in processing the car, it might prevent us from making a serious prediction error. Estimate the mileage on the basis of other variables. For example, if the car was "leased," it might be reasonable to assume that the average mileage allowance is 12,000 miles per year.

Second, because the prediction model has to provide more than just tomorrow's price (as it takes some time to transport the car to Jacksonville, and so we need a predicted price for next week and/or three weeks from today), the training process might be much more complex. The reason for this increased complexity is hidden in the fact that the prediction model's accuracy must be assessed for both shorter and longer time periods.

Third, from time to time the linear regression model would process a "rare" car, such as a Dodge Viper or Acura NSX. Note that we assumed a linear regression model for each make/model at each location. This assumption is fine, but the historical data set may only contain 100 Dodge Viper cars with zero occurrences at some locations!

How can we build a model for a location where the data set is empty? Well, as usual, there are several ways of dealing with this problem. One way would be to estimate the price on the basis of (1) prices of the same make/model at nearby locations, and (2) prices of similar models at the same location.

Distance Methods

It is based on the concept of "distance between cases." Any two cases in a data set can be compared for similarity, and this similarity measure (called "distance"). Using a distance measure within a data set would allow us to compare a new case with the most "similar" existing case.

Going back to our example of the Toyota Camry at the Jacksonville auction site, we may search our database of three million cases for the most similar Toyota Camry sold in Jacksonville and use its sale price as our prediction.

The essential aspect of this approach is creating a similarity measure between cases, because the probability of finding an identical case is very low. Hence, we have to base our decisions on similarities,

For instance, is a silver Toyota Camry with 33,000 miles "more similar" to a white Toyota Camry with 34,100 miles, or to a silver Toyota Camry with 36,000 miles?

One of the most popular distance-based prediction methods is k nearest neighbor, where k nearest neighbors (i. e., k most similar cases) of a new case are determined.

Clearly, if k = 1 (i. e., we find only one neighbor), the outcome of this single neighbor is the prediction for the new case. If k > 1, then a voting mechanism is used (classification problems) or the average value of the k answers is calculated (regression problems).

Note that calculating the distance is trivial when there is only one numeric variable, (e. g., 5.7 – 3.8 = 1.9). With several numerical variables, a Euclidean distance can be used, provided that the variables are normalized and of equal importance (otherwise a weighting must be applied).

The largest problem, however, is with nominal variables. Given our earlier question of whether "the difference in similarity between 'silver' and 'white' is the same as between 'red' and 'yellow'?" we can assume that different colors are just different (resulting in a distance of 1), or we can introduce a more sophisticated matrix that would assign a numeric measure for each color (e. g., so that the difference between "light blue" and "dark blue" is smaller than the difference between "blue" and "red").

Another issue to consider is that of missing values. A standard approach is to assume that the distance between an existing value and a missing value is as large as possible. Hence, for nominal values, the distance is assigned a normalized value of 1 (all distances are between 0 and 1), and for numeric variables the distance is assigned the largest possible normalized value between 0 and 1.

A distance-based method might be too time consuming for large data sets, because the whole data set must be searched to evaluate each new case. With larger values of parameter k, the computation time increases significantly. For efficiency reasons, it would be beneficial to reduce the number of stored cases. Also, some clustering methods can be used to group the

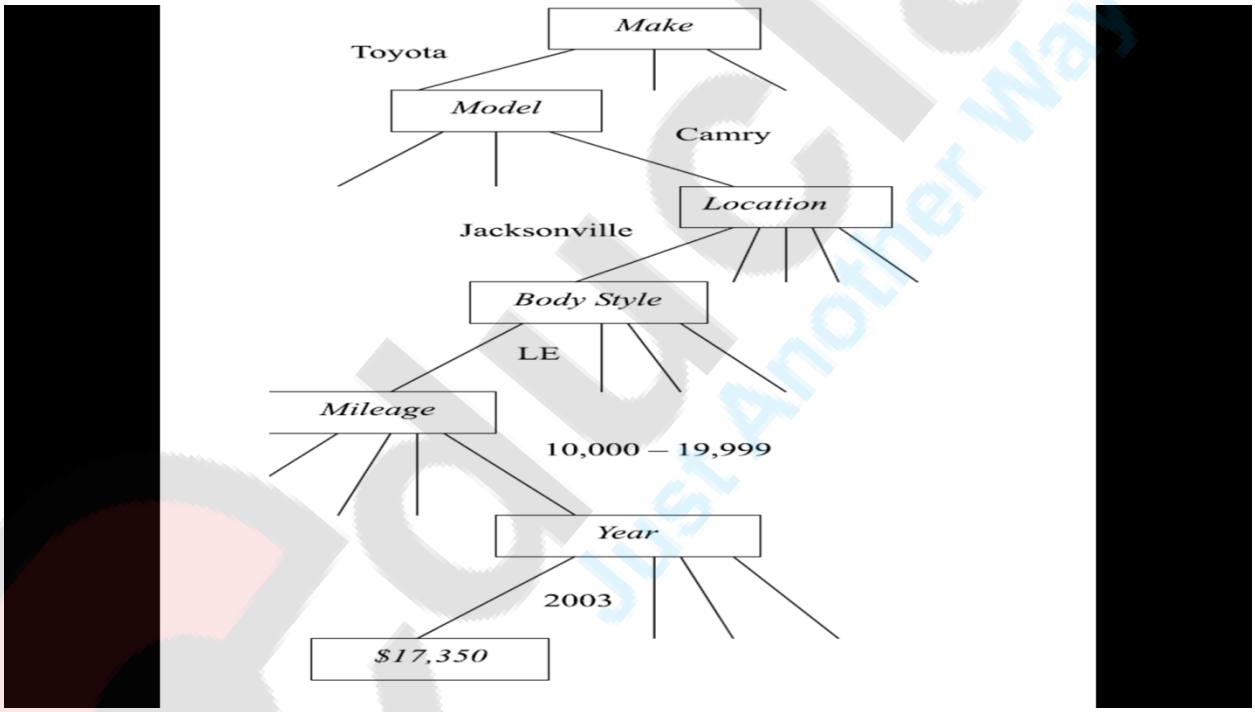
cases into meaningful categories. A new case would then be assigned to an existing category and the predicted value would be drawn from the cases present in that category

Logic Methods

A decision table (also known as a lookup table) is the simplest logic-based method for prediction, and there are many such tables published for estimating the price of a used car sold at auction.

The most widely used logic methods, is the decision tree. Because the structure of a decision tree is relatively easy to follow and understand (especially for smaller trees), its popularity is widespread.

To make a prediction for a new case, the root of a tree is examined, a test is performed and, depending on the result of the test, the case moves down the appropriate branch. The process continues until a terminal node (also known as a "leaf") is reached, and the value of this terminal node is the predicted outcome.



Another logic method is based on decision rules, which are "similar" to decision trees: after all, a decision tree can be interpreted as a collection of rules. For example, the single branch of the decision tree displayed earlier can be converted into the following rule:

if Make = Toyota & Model = Camry & Location = Jacksonville

& Body Style = LE & 10,000 ≤ Mileage ≤ 19,999 & Year = 2003, then Sale Price = $17,350

Association rules, which describe some regularity present in the data and can "predict" any variable (rather than just the class). For example, an associate rule may state that

if Make = Porsche & Model = Carrera, then Location in {Jacksonville, Tampa, Los Angeles, San Francisco, San Diego} as Porsche Carreras are sold only at auction sites in Florida and California.

Rules with exception, which extend a rule with exceptions, may refer to association rules, e. g.,

If Make = Porsche & Model = Carrera, then Location in {Jacksonville,Tampa, Los Angeles, San Francisco, San Diego}, except if Year ≤ 1997,then Location in {Austin, Houston, Dallas}

which states that older Porsche Carreras (produced in 1997 or earlier) are sent to auction sites in Texas; or to a classification rule, e. g.,

If Make = Toyota & Model = Camry & Location = Jacksonville & Body Style = LE & 10,000 ≤ Mileage ≤ 19,999 & Year = 2003, then Sale Price = $17,350, except if Color = Red, then Sale Price = $18,450 as red was a rare (but popular) color for Toyota Camry cars in 2003 and that increases the price.

Modern Heuristic Methods

The methods fall into the category of "modern heuristics are fuzzy systems, neural networks, genetic programming, and agent-based systems. These methods originated in different research communities, and their "mechanics" are very different to classic methods such as statistics and machine learning.

Many other considerations must be taken when selecting the "best" prediction method for an Adaptive Business Intelligence system. Although the prediction error is quite possibly the most important measure, it only provides one dimension of a model's quality.

Many other factors must be considered, such as:

1) Response time

2) Editing

3) Prediction justification

4) Model compactness

5) Tolerance for noise

1) Response time.

This is an essential consideration, as any Adaptive Business Intelligence system would have a defined response time. Fraud detection systems,for example, process millions of transactions per second, so the frequency of predictions (i. e., classifications of "fraudulent" or "legitimate") is very high. Other prediction methods, on the other hand, might be used on a weekly basis(e. g., inventory management) and so the response time is not that critical.

2) Editing.

Some prediction models are difficult to edit (e. g., neural networks),while others (e. g., rule-based systems) are easy. The ability to edit a model is an important consideration, as it might be necessary to add the knowledge of experts to the final model.

3) Prediction justification:

This is an often-overlooked aspect of evaluating the usefulness of a prediction model.

For some applications (e. g., credit scoring) it is very important to justify the prediction; in some cases, this might even be required by law (e. g., justification for rejecting a loan application).

4) Model compactness.

A prediction model should not be exceedingly large and complex, as that would make it difficult for humans to understand; also, it might take a longer amount of time to make predictions. A more compact prediction model is preferable over a sprawling prediction model assuming they both do an equally good job of predicting.

5) Tolerance for noise.

All prediction methods require some approach for handling missing values (e. g., the mileage of an off-lease car has not been recorded), but some methods do a better job of handling missing values than others.

Because of these many factors, it may be difficult to select "the best" prediction method for the problem at hand. Different prediction methods have different properties ,and so some of them may perform better or worse when trained on different data sets. Hence, it might be worthwhile to use a few methods to build a few models, and then use all the models to reach a consensus.

Evaluation of Models

Different prediction methods to build a variety of different prediction models, the key issue is which method should be applied to a particular problem. At first blush, this may seem easy. After all, after we complete and train a few models, we can test them on the data and measure the prediction error. The best model would then be selected for implementation.

First of all, the amount of available data might not be that large.

Secondly, the performance of a prediction model on the training data might be very different from the performance of the same model on an independent set of data. This might be due to over fitting, which is a common phenomenon. In short, a model tunes itself during the training stage to such an extent that all predictions on the training data set are perfect.

Thirdly, prediction models that provide different outcomes require different techniques for error measurement.

For instance, a prediction model may indicate whether a new case belongs to class A or B, or, if we have a larger number of classes, the probability that a new case belongs to each class. Alternately, a prediction model may predict a number (e. g., sale price) or a sequence of numbers (e. g.,sale price and sale date). In each of these examples, we have to carefully consider what we are predicting and apply the appropriate error measurement technique.

Cost of a potential error.

When classifying cases into two categories ("yes" or "no," "fraudulent" or "legitimate," etc.),there are two types of errors:

(a) false-positive, where the outcome is incorrectly predicted as "yes," when in fact it is "no," and

(b) false-negative, where the outcome is incorrectly predicted as "no," when in fact it is "yes"

Clearly, the cost of these errors is very different. By classifying a legitimate transaction as fraudulent (false-positive), there is a small cost to check the transaction. On the other hand, classifying a fraudulent transaction a legitimate (false-negative) usually carries a much higher cost,

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Type I error.

Type II error.

To predict a model's performance on new data divide the data set into 3 categories.

1) The training data set is used for building a prediction model.
2) The validation data set is used for tuning the parameters of the model (i. e., for optimizing the performance of the model).
3) The test data set is used to evaluate the performance of the model

UNIT 3

Definition of data warehouse:

A data warehouse is the foremost repository for the data available for developing business intelligence architectures and decision support systems. The term *data warehousing* indicates the whole set of interrelated activities involved in designing, implementing and using a data warehouse.

There are several reasons for implementing a data warehouse separately from the databases supporting OLTP applications in an enterprise. Among them, we recall here the most relevant.

Integration
In many instances, decision support systems must access information originating from several data sources, distributed across different parts of an organization or deriving from external sources. A data warehouse integrating multiple and often heterogeneous sources is then required to promote and facilitate the access to information. Data integration may be achieved by means of different techniques – for example, by using uniform encoding methods, converting to standard measurement units and achieving a semantic homogeneity of information.

Quality
The data transferred from operational systems into the data warehouse are examined and corrected in order to obtain reliable and error-free information, as much as possible. Needless to say, this increases the practical value of business intelligence systems developed starting from the data contained in a data warehouse.

Efficiency
Queries aimed at extracting information for a business intelligence analysis may turn out to be burdensome in terms of computing resources and processing time. As a consequence, if a 'killer' query were directed to the transactional systems it would risk severely compromising the efficiency required by enterprise resource planning applications, with negative consequences on the routine activities of a company. A better solution is then to direct complex queries for OLAP analyses to the data warehouse, physically separated from the operational systems.

Extendability
The data stored in transactional systems stretch over a limited time span in the past. Indeed, due to limitations on memory capacity, data relative to past periods are regularly removed from OLTP systems and permanently archived in off-line mass-storage devices, such as DVDs or magnetic tapes. On the other hand, business intelligence systems and prediction models need to access all available past data to be able to grasp trends and detect recurrent patterns. This is possible due to the ability of data warehouses to retain historical information.

Entity-oriented
The data contained in a data warehouse are primarily concerned with the main entities of interest for the analysis, such as products, customers, orders and sales. On the other hand, transactional systems are more oriented toward operational activities and are based on each single transaction recorded by enterprise resource planning applications. During a business intelligence analysis, orientation toward the entities allows the performance of a company to be more easily evaluated and any potential source of inefficiencies to be detected.

Integrated
The data originating from the different sources are integrated and homogenized as they are loaded into a data warehouse. For example, measurement units and encodings are harmonized and made consistent.

Time-variant
All data entered in a data warehouse are labeled with the time period to which they refer. We can fairly relate the data stored in a data warehouse to a sequence of nonvolatile snapshot pictures, taken at successive times and bearing the label of the reference period. As a consequence, the temporal dimension in any data warehouse is a critical element that plays a predominant role. In this way decision support applications may develop historical trend analysis.

Persistent
Once they have been loaded into a data warehouse, data are usually not modified further and are held permanently. This feature makes it easier toorganize read-only access by users and simplifies the updating process, avoiding concurrency which is of critical importance for operational systems.

Consolidated
Usually some data stored in a data warehouse are obtained as partial summaries of primary data belonging to the operational systems from which they originate. For example, a mobile phone company may store in a data warehouse the total cost of the calls placed by each customer in a week, subdivided by traffic routes and by type of service selected, instead of storing the individual calls recorded by the operational systems. The reason for such consolidation is twofold: on one hand, the reduction in the space required to store in the data warehouse the data accumulated over the years; on the other hand, consolidated information may be able to better meet the needs of business intelligence systems.

Denormalized
Unlike operational databases, the data stored in a data warehouse are not structured in normal form but can instead make provision for redundancies, to allow shorter response time to complex queries.

Granularity represents the highest level of detail expressed by the primary data contained in a data warehouse, also referred to as *atomic data*. Obviously, the granularity of a data warehouse cannot exceed that of the original data

Data warehouse architecture

The reference architecture of a data warehouse, shown in Figure 3.1, includes the following major functional components.

• The data warehouse itself, together with additional data marts, that contains the data and the functions that allow the data to be accessed, visualized and perhaps modified.

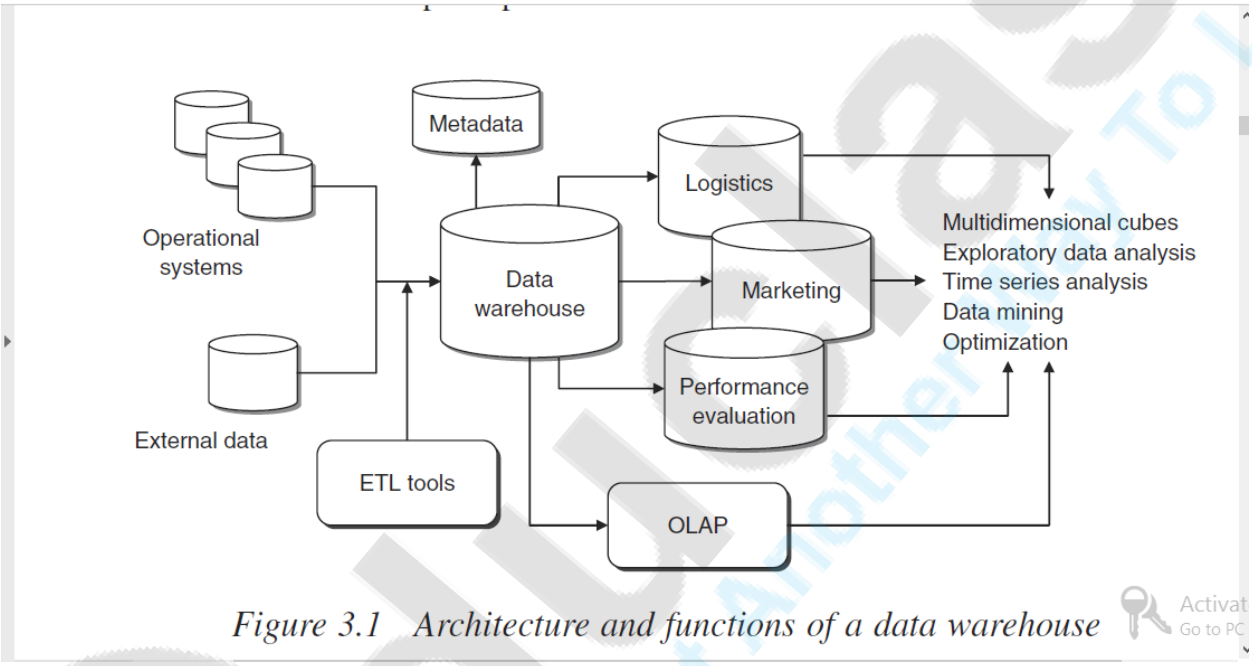


Figure 3.1 Architecture and functions of a data warehouse

• Data acquisition applications, also known as extract, transform and load (ETL) or back-end tools, which allow the data to be extracted, transformed and loaded into the data warehouse.

• Business intelligence and decision support applications, which represent the front-end and allow the knowledge workers to carry out the analyses and visualize the results.

The three-level distinction applies to the architecture shown in Figure 3.1 even from a technological perspective.

• The level of the data sources and the related ETL tools that are usually installed on one or more servers.

• The level of the data warehouse and any data mart, possibly available on one or more servers as well, and separated from those containing the data sources. This second level also includes the metadata documenting the origin and meaning of the records stored in the data warehouse.

• The level of the analyses that increase the value of the information contained in a data warehouse through query, reporting and possibly sophisticated decision support tools. The applications for business intelligence and decision support analysis are usually found on separate servers or directly on the client PC used by analysts and knowledge workers.

A data warehouse may be implemented according to different design approaches: top-down, bottom-up and mixed.

Top-down

The top-down methodology is based on the overall design of the data warehouse, and is therefore more systematic. However, it implies longer development times and higher risks of not being completed within schedule since the whole data warehouse is actually being developed.

Bottom-up

The bottom-up method is based on the use of prototypes and therefore system extensions are made according to a step-by-step scheme. This approach is usually quicker, provides more tangible results but lacks an overall vision of the entire system to be developed.

Mixed

The mixed methodology is based on the overall design of the data warehouse, but then proceeds with a prototyping approach, by sequentially implementing different parts of the entire system. This approach is highly practical and usually preferable, since it allows small and controlled steps to be taken while bearing in mind the whole picture.

The steps in the development of a data warehouse or a data mart can be summarized as follows.

• One or more processes within the organization to be represented in the data warehouse are identified, such as sales, logistics or accounting.
• The appropriate granularity to represent the selected processes is identified and the atomic level of the data is defined.
• The relevant measures to be expressed in the fact tables for multidimensional analysis are then chosen.
• Finally, the dimensions of the fact tables are determined.

ETL tools

ETL refers to the software tools that are devoted to performing in an automatic way three main functions: *extraction*, *transformation* and *loading* of data into the data warehouse.

Extraction

During the first phase, data are extracted from the available internal and external sources. A logical distinction can be made between the initial extractions, where the available data relative to all past periods are fed into the empty data warehouse, and the subsequent incremental extractions that update the data warehouse using new data that become available over time. The selection of data to be imported is based upon the data warehouse design, which in turn depends on the information needed by business intelligence analyses and decision support systems operating in a specific application domain.

Transformation

The goal of the cleaning and transformation phase is to improve the quality of the data extracted from the different sources, through the correction of inconsistencies, inaccuracies and missing values. Some of the major shortcomings that are removed during the data cleansing stage are:

- Inconsistencies between values recorded in different attributes having the same meaning;
- Data duplication;
- Missing data;
- Existence of inadmissible values.

During the cleaning phase, preset automatic rules are applied to correct most recurrent mistakes. In many instances, dictionaries with valid terms are used to substitute the supposedly incorrect terms, based upon the level of similarity.
Moreover, during the transformation phase, additional data conversions occur in order to guarantee homogeneity and integration with respect to the different data sources. Furthermore, data aggregation and consolidation are performed in order to obtain the summaries that will reduce the response time required by subsequent queries and analyses for which the data warehouse is intended.

Loading

Finally, after being extracted and transformed, data are loaded into the tables of the data warehouse to make them available to analysts and decision support applications.

2) Differentiate between OLTP and OLAP systems.

Feature	OLTP	OLAP
Characteristic	operational processing	informational processing
Orientation	transaction	analysis
User	clerk, DBA, database professional	knowledge worker (e.g., manager, executive, analyst)
Function	day-to-day operations	long-term informational requirements, decision support
DB design	ER based, application-oriented	star/snowflake, subject-oriented
Data	current; guaranteed up-to-date	historical; accuracy maintained over time
Summarization	primitive, highly detailed	summarized, consolidated
View	detailed, flat relational	summarized, multidimensional
Unit of work	short, simple transaction	complex query
Access	read/write	mostly read
Focus	data in	information out
Operations	index/hash on primary key	lots of scans
No. of records accessed	tens	millions
Number of users	thousands	hundreds
DB size	100 MB to GB	100 GB to TB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time

Table 3.1 Differences between OLTP and OLAP systems

Characteristic	OLTP	OLAP
volatility	dynamic data	static data
timeliness	current data only	current and historical data
time dimension	implicit and current	explicit and variant
granularity	detailed data	aggregated and consolidated data
updating	continuous and irregular	periodic and regular
activities	repetitive	unpredictable
flexibility	low	high
performance	high, few seconds per query	may be low for complex queries
users	employees	knowledge workers
functions	operational	analytical
purpose of use	transactions	complex queries and decision support
priority	high performance	high flexibility
metrics	transaction rate	effective response
size	megabytes to gigabytes	gigabytes to terabytes

Data marts

Data marts are systems that gather all the data required by a specific company department, such as marketing or logistics, for the purpose of performing business intelligence analyses and executing decision support applications specific to the function itself. Therefore, a data mart can be considered as a functional or departmental data warehouse of a smaller size and a more specific type than the overall company data warehouse.

A data mart therefore contains a subset of the data stored in the company data warehouse, which are usually integrated with other data that the company department responsible for the data mart owns and deems of interest. For example, a marketing data mart will contain data extracted from the central data warehouse, such as information on customers and sales transactions, but also additional data pertaining to the marketing function, such as the results of marketing campaigns run in the past.
Data warehouses and data marts thus share the same technological framework.
In order to implement business intelligence applications, some companies prefer to design and develop in an incremental way a series of integrated data marts rather than a central data warehouse, in order to reduce the implementation time and uncertainties connected with the project.

Metadata

In order to document the meaning of the data contained in a data warehouse, it is recommended to set up a specific information structure, known as *metadata*, i.e. data describing data. The metadata indicate for each attribute of a data warehouse the original

source of the data, their meaning and the transformations to which they have been subjected. The documentation provided by metadata should be constantly kept up to date, in order to reflect any modification in the data warehouse structure. The documentation should be directly accessible to the data warehouse users, ideally through a web browser, according to the access rights pertaining to the roles of each analyst.

In particular, metadata should perform the following informative tasks:
• A documentation of the data warehouse structure: layout, logical views, dimensions, hierarchies, derived data, localization of any data mart;
• A documentation of the data genealogy, obtained by tagging the data sources from which data were extracted and by describing any transformation performed on the data themselves;
• A list keeping the usage statistics of the data warehouse, by indicating how many accesses to a field or to a logical view have been performed;
• A documentation of the general meaning of the data warehouse with respect to the application domain, by providing the definition of the terms utilized, and fully describing data properties, data ownership and loading policies.

OLAP operations

Roll-up
A *roll-up* operation, also termed *drill-up*, consists of an aggregation of data in the cube, which can be obtained alternatively in the following two ways.
• Proceeding upwards to a higher level along a single dimension defined over a concepts hierarchy. For example, for the {location} dimension it is possible to move upwards from the {city} level to the {province} level and to consolidate the measures of interest through a *group-by* conditioned sum over all records whereby the city belongs to the same province.

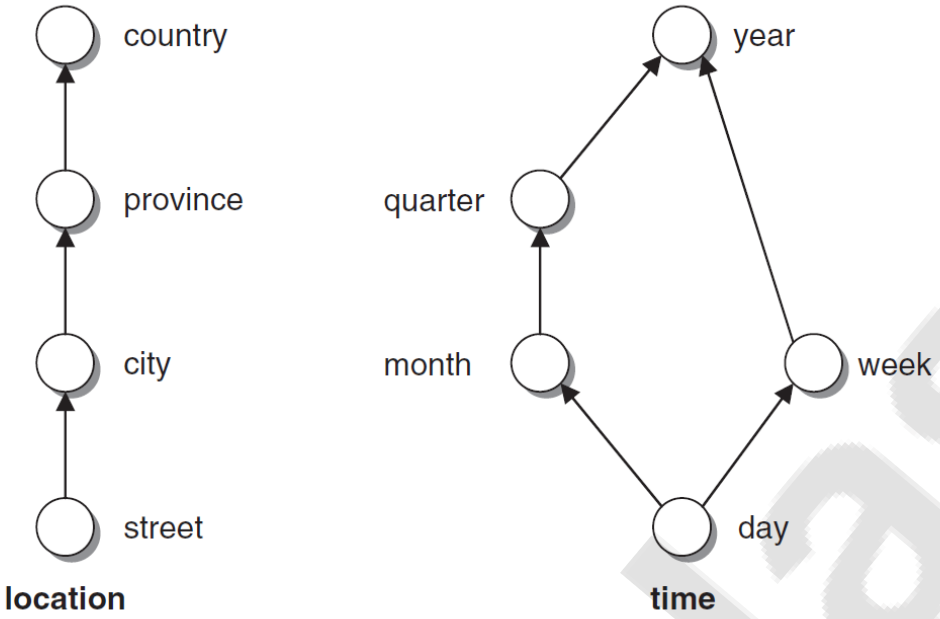


Figure 3.7 Hierarchies of concepts

• Reducing by one dimension. For example, the removal of the {time} dimension leads to consolidated measures through the sum over all time periods existing in the data cube.

Roll-down

A *roll-down* operation, also referred to as *drill-down*, is the opposite operation to roll-up. It allows navigation through a data cube from aggregated and consolidated information to more detailed information. The effect is to reverse the result achieved through a roll-up operation. A drill-down operation can therefore be carried out in two ways.

• Shifting down to a lower level along a single dimension hierarchy. For example, in the case of the {location} dimension, it is possible to shift from the {province} level to the {city} level and to disaggregate the measures of interest over all records whereby the city belongs to the same province.

• Adding one dimension. For example, the introduction of the {time} dimension leads to disaggregate the measures of interest over all time periods existing in a data cube.

Slice and dice

Through the *slice* operation the value of an attribute is selected and fixed along one dimension. For example, Table 3.3 has been obtained by fixing the region at the {Usa} value. The *dice* operation obtains a cube in a subspace by selecting several dimensions simultaneously.

Pivot

The *pivot* operation, also referred to as *rotation*, produces a rotation of the axes, swapping some dimensions to obtain a different view of a data cube.

OLAP Server Architectures: ROLAP versus MOLAP versus HOLAP

Logically, OLAP servers present business users with multidimensional data from data warehouses or data marts, without concerns regarding how or where the data are stored. However, the physical architecture and implementation of OLAP servers must consider data storage issues. Implementations of a warehouse server for OLAP processing include the following:

Relational OLAP (ROLAP) servers:
These are the intermediate servers that stand in between a relational back-end server and client front-end tools. They use a relational or extended-relational DBMS to store and manage warehouse data, and OLAP middleware to support missing pieces. ROLAP servers include optimization for each DBMS back end, implementation of aggregation navigation logic, and additional tools and services. ROLAP technology tends to have greater scalability than **MOLAP technology**
The DSS server of Microstrategy, for example, adopts the ROLAP approach.

Multidimensional OLAP (MOLAP) servers:
These servers support multidimensional data views through *array-based multidimensional storage engines*. They map multidimensional views directly to data cube array structures. The advantage of using a data cube is that it allows fast indexing to precomputed summarized data Many MOLAP servers adopt a two-level storage representation to handle dense and sparse data sets: Denser subcubes are identified and stored as array structures, whereas sparse subcubes employ compression technology for efficient storage utilization.

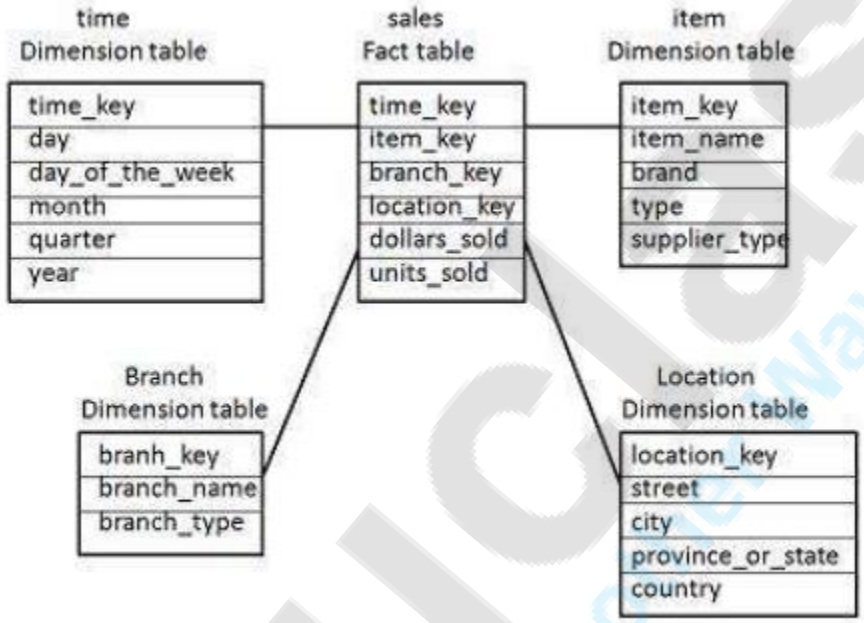
Hybrid OLAP (HOLAP) servers:
The hybrid OLAP approach combines ROLAP and MOLAP technology, benefiting from the greater scalability of ROLAP and the faster computation of MOLAP. For example, a HOLAP server may allow large volumes of detailed data to be stored in a relational database, while aggregations are kept in a separate MOLAP store. The Microsoft SQL Server 2000 supports a hybrid OLAP server.

Schema

Schema is a logical description of the entire database. It includes the name and description of records of all record types including all associated data-items and aggregates. Much like a database, a data warehouse also requires to maintain a schema. A database uses relational model, while a data warehouse uses Star, Snowflake, and Fact Constellation schema. In this chapter, we will discuss the schemas used in a data warehouse.

Star Schema

- Each dimension in a star schema is represented with only one-dimension table.
- This dimension table contains the set of attributes.
- The following diagram shows the sales data of a company with respect to the four dimensions, namely time, item, branch, and location.

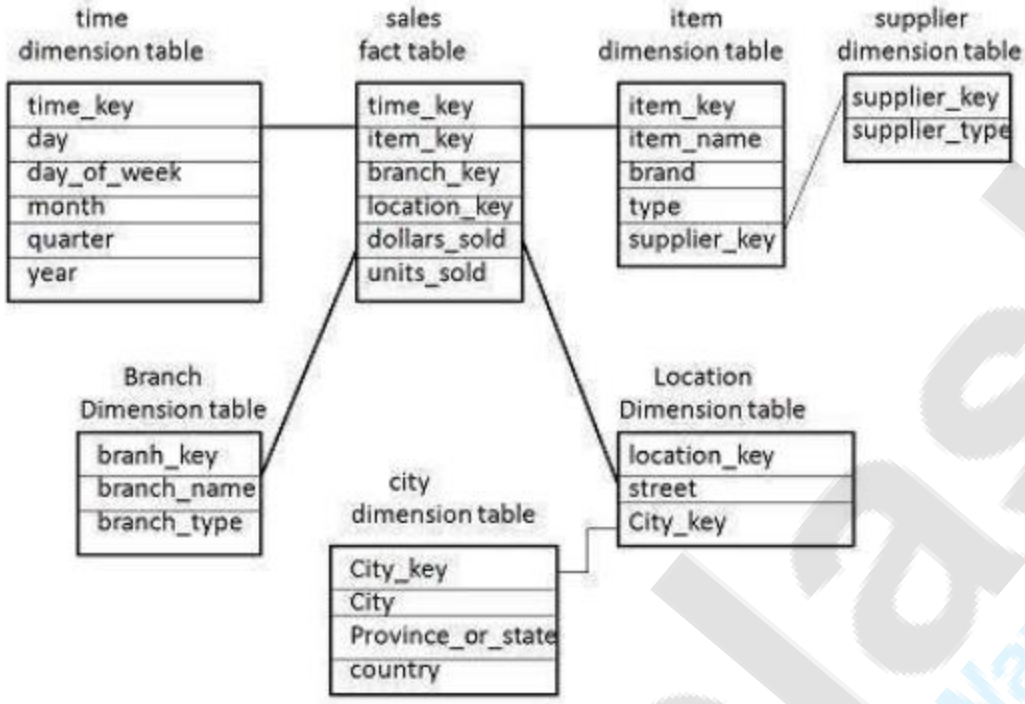


- There is a fact table at the center. It contains the keys to each of four dimensions.
- The fact table also contains the attributes, namely dollars sold and units sold.

Note –Each dimension has only one dimension table and each table holds a set of attributes. For example, the location dimension table contains the attribute set {location_key, street, city, province_or_state,country}. This constraint may cause data redundancy. For example, "Vancouver" and "Victoria" both the cities are in the Canadian province of British Columbia. The entries for such cities may cause data redundancy along the attributes province_or_state and country.

Snowflake Schema

- Some dimension tables in the Snowflake schema are normalized.
- The normalization splits up the data into additional tables.
- Unlike Star schema, the dimensions table in a snowflake schema are normalized. For example, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.

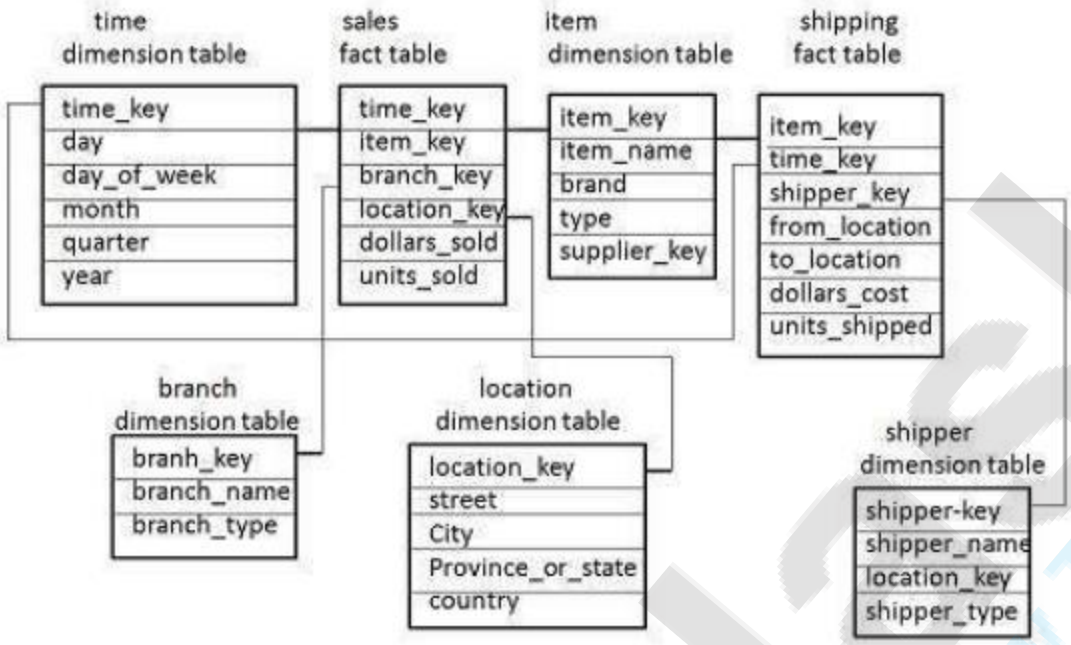


- Now the item dimension table contains the attributes item_key, item_name, type, brand, and supplier-key.
- The supplier key is linked to the supplier dimension table. The supplier dimension table contains the attributes supplier_key and supplier_type.

Note − Due to normalization in the Snowflake schema, the redundancy is reduced and therefore, it becomes easy to maintain and the save storage space.

Fact Constellation Schema

- A fact constellation has multiple fact tables. It is also known as galaxy schema.
- The following diagram shows two fact tables, namely sales and shipping.



- The sales fact table is same as that in the star schema.
- The shipping fact table has the five dimensions, namely item_key, time_key, shipper_key, from_location, to_location.
- The shipping fact table also contains two measures, namely dollars sold and units sold.
- It is also possible to share dimension tables between fact tables. For example, time, item, and location dimension tables are shared between the sales and shipping fact table.

UNIT -4

What Is Data Mining?

Data mining refers to extraction of information from large amount of data.

In today's world data mining is very important because huge amount of data is present in companies and different type of organization. It becomes impossible for humans to extract information from this large data, so machine learning technology are used in order to process data fast enough to extract information from it.

Data mining is used by companies in order to get customer preferences, determine price of their product and services and to analyze market.

Data mining is also known as knowledge discovery in Database (KDD).

Functionalities/Techniques:

1) Concept/Class Description: Characterization and Discrimination
2) Mining Frequent Patterns, Associations and correlations
3) Classification and Prediction
4) Cluster Analysis
5) Outlier Analysis
6) Evolution Analysis

1) Concept/Class Description: Characterization and Discrimination

Data Characterization: A data mining system should be able to produce a description summarizing the characteristics of customers.

Example: The characteristics of customers who spend more than $1000 a year at (some store called) All Electronics. The result can be a general profile such as age, employment status or credit ratings.

Data Discrimination: It is a comparison of the general features of targeting class data objects with the general features of objects from one or a set of contrasting classes. User can specify target and contrasting classes.

Example: The user may like to compare the general features of software products whose sales increased by 10% in the last year with those whose sales decreased by about 30% in the same duration.

2) Mining Frequent Patterns, Associations and correlations

Frequent patterns, are patterns that occur frequently in data. There are many kinds of frequent patterns, including itemsets, subsequences, and substructures.

Association analysis

Suppose, as a marketing manager, you would like to determine which items are frequently purchased together within the same transactions.

buys(X,"computer")=buys(X,"software") [support=1%,confidence=50%]

where X is a variable representing a customer.Confidence=50% means that if a customer buys a computer, there is a 50% chance that she will buy software as well.

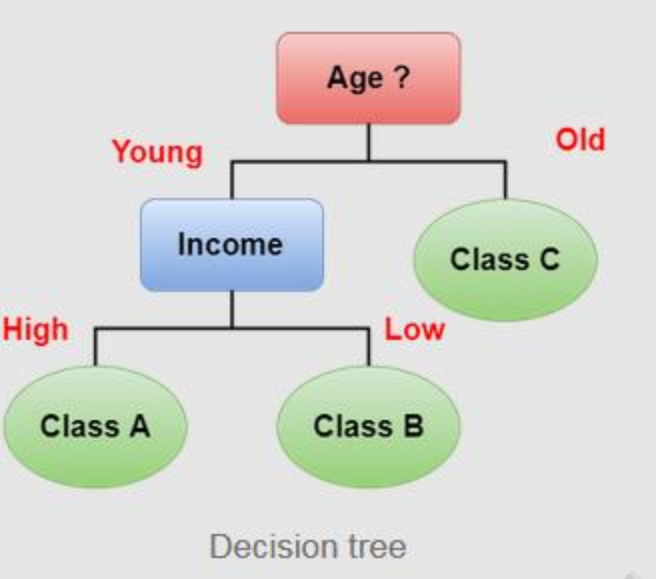
Support=1% means that 1% of all of the transactions under analysis showed that computer and software were purchased together.

3) Classification and Prediction

Classification is the process of finding a model that describes and distinguishes data classes for the purpose of being able to use the model to predict the class of objects whose class label is unknown.

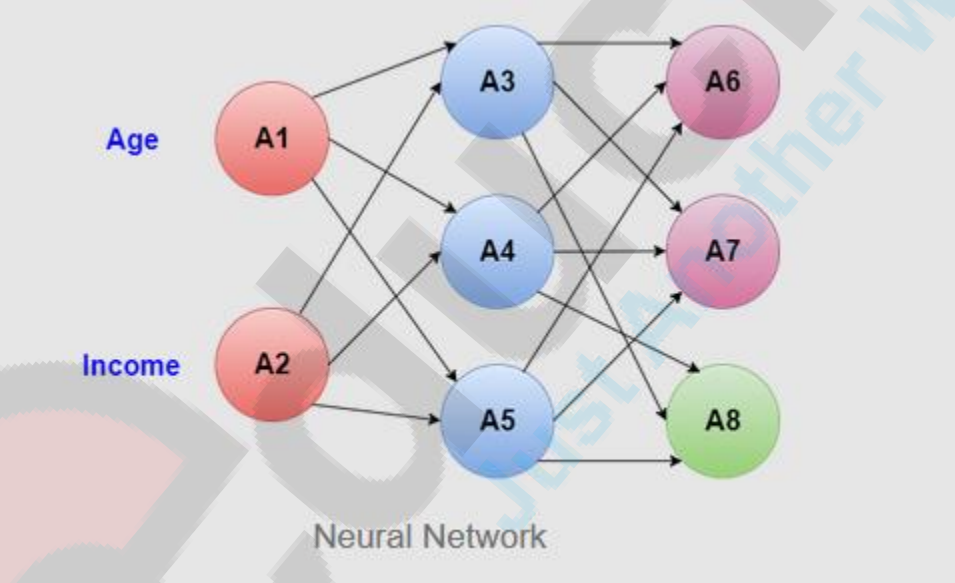
"How is the derived model presented?" The derived model may be represented in various forms, such as classification (IF-THEN) rules, decision trees, mathematical formulae, or neural networks.

A **decision tree** is a flow-chart-like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test, and tree leaves represent classes or class distributions.



Decision tree

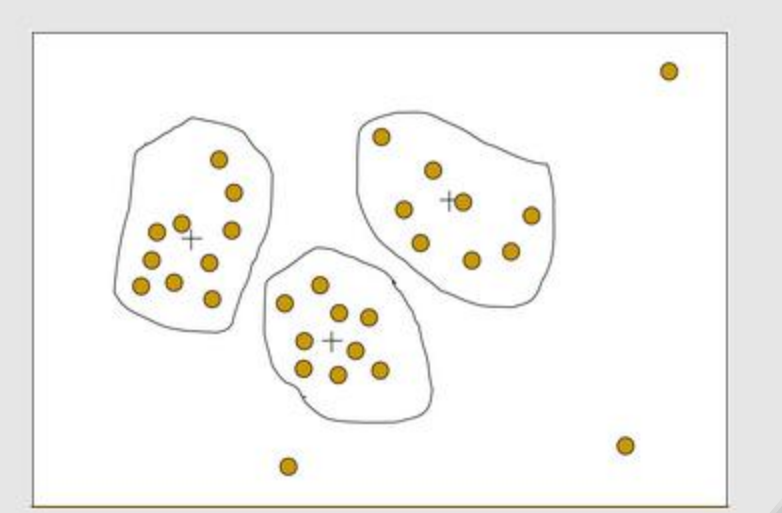
A **neural network**, when used for classification, is typically a collection of neuron-like processing units with weighted connections between the units.



Neural Network

4) Cluster Analysis

Clustering analyses data objects without consulting a known class label.
Example: Cluster analysis can be performed on All Electronics customer data in order to identify homogeneous subpopulations of customers. These clusters may represent individual target groups for marketing. The figure on next slide shows a 2-D plot of customers with respect to customer locations in a city.



Cluster Analysis

The objects are grouped based on the principle of maximizing the intraclass similarity and minimizing the interclass similarity. That is, clusters of objects are formed so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters.

5) Outlier Analysis

A database may contain data objects that do not comply with the general behavior or model of the data. These data objects are outliers.

Example: Use in finding Fraudulent usage of credit cards. Outlier Analysis may uncover Fraudulent usage of credit cards by detecting purchases of extremely large amounts for a given account number in comparison to regular charges incurred by the same account. Outlier values may also be detected with respect to the location and type of purchase or the purchase frequency.

6) Evolution Analysis

Data evolution analysis describes and models regularities or trends for objects whose behavior changes over time.

Example: Time-series data. If the stock market data (time-series) of the last several years available from the New York Stock exchange and one would like to invest in shares of high tech industrial companies. A data mining study of stock exchange data may identify stock evolution regularities for overall stocks and for the stocks of particular companies. Such regularities may help predict future trends in stock market prices, contributing to one's decision making regarding stock investments.

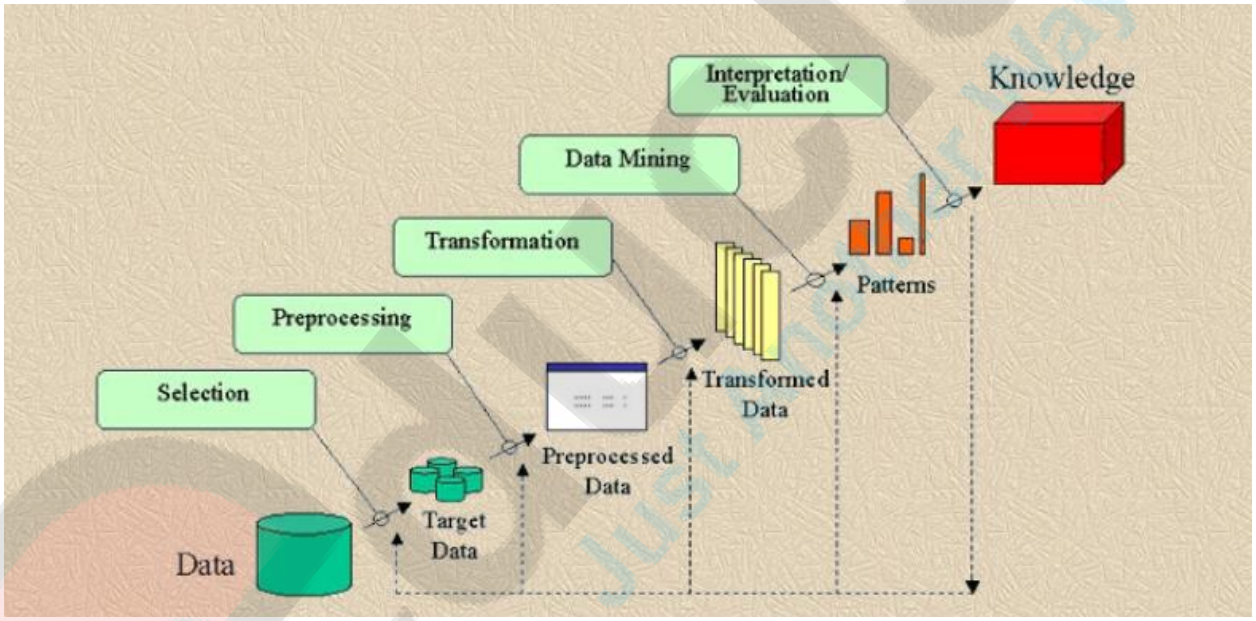
What is the KDD Process?

The term **Knowledge Discovery in Databases**, or KDD for short, refers to the broad process of finding knowledge in data, and emphasizes the "high-level" application of particular data mining methods. It is of interest to researchers in machine learning, pattern recognition, databases, statistics, artificial intelligence, knowledge acquisition for expert systems, and data visualization.

The unifying goal of the KDD process is to extract knowledge from data in the context of large databases.

It does this by using data mining methods (algorithms) to extract (identify) what is deemed knowledge, according to the specifications of measures and thresholds, using a database along with any required preprocessing, subsampling, and transformations of that database.

An Outline of the Steps of the KDD Process



The overall process of finding and interpreting patterns from data involves the repeated application of the following steps:

1. Developing an understanding of
   - the application domain
   - the relevant prior knowledge
   - the goals of the end-user
2. Creating a target data set: selecting a data set, or focusing on a subset of variables, or data samples, on which discovery is to be performed.

3. Data cleaning and preprocessing.
    - Removal of noise or outliers.
    - Collecting necessary information to model or account for noise.
    - Strategies for handling missing data fields.
    - Accounting for time sequence information and known changes.
4. Data reduction and projection.
    - Finding useful features to represent the data depending on the goal of the task.
    - Using dimensionality reduction or transformation methods to reduce the effective number of variables under consideration or to find invariant representations for the data.
5. Choosing the data mining task.
    - Deciding whether the goal of the KDD process is classification, regression, clustering, etc.
6. Choosing the data mining algorithm(s).
    - Selecting method(s) to be used for searching for patterns in the data.
    - Deciding which models and parameters may be appropriate.
    - Matching a particular data mining method with the overall criteria of the KDD process.
7. Data mining.
    - Searching for patterns of interest in a particular representational form or a set of such representations as classification rules or trees, regression, clustering, and so forth.
8. Interpreting mined patterns.
9. Consolidating discovered knowledge.

Data Cleaning in Data Mining

Quality of your data is critical in getting to final analysis.Any data which tend to be incomplete, noisy and inconsistent can effect your result.

Data cleaning in data mining is the process of detecting and removing corrupt or inaccurate records from a record set, table or database.

Some data cleaning methods:-

1. You can ignore the tuple.This is done when class label is missing.This method is not very effective, unless the tuple contains several attributes with missing values.
2. You can fill in the missing value manually.This approach is effective on small data set with some missing values.
3. You can replace all missing attribute values with global constant, such as a label like "Unknown" or minus infinity.

4. You can use the attribute mean to fill in the missing value.For example customer average income is 25000 then you can use this value to replace missing value for income.
5. Use the most probable value to fill in the missing value.

Noisy Data

Noise is a random error or variance in a measured variable. Noisy Data may be due to faulty data collection instruments, data entry problems and technology limitation.

How to Handle Noisy Data?

Binning:

Binning methods sorted data value by consulting its "neighbor- hood," that is, the values around it.The sorted values are distributed into a number of "buckets," or bins.

For example

Price = 4, 8, 15, 21, 21, 24, 25, 28, 34

Partition into (equal-frequency) bins:

Bin a: 4, 8, 15

Bin b: 21, 21, 24

Bin c: 25, 28, 34

In this example, the data for price are first sorted and then partitioned into equal-frequency bins of size 3.

Smoothing by bin means:

Bin a: 9, 9, 9

Bin b: 22, 22, 22

Bin c: 29, 29, 29

In smoothing by bin means, each value in a bin is replaced by the mean value of the bin.

Smoothing by bin boundaries:

Bin a: 4, 4, 15

Bin b: 21, 21, 24

Bin c: 25, 25, 34

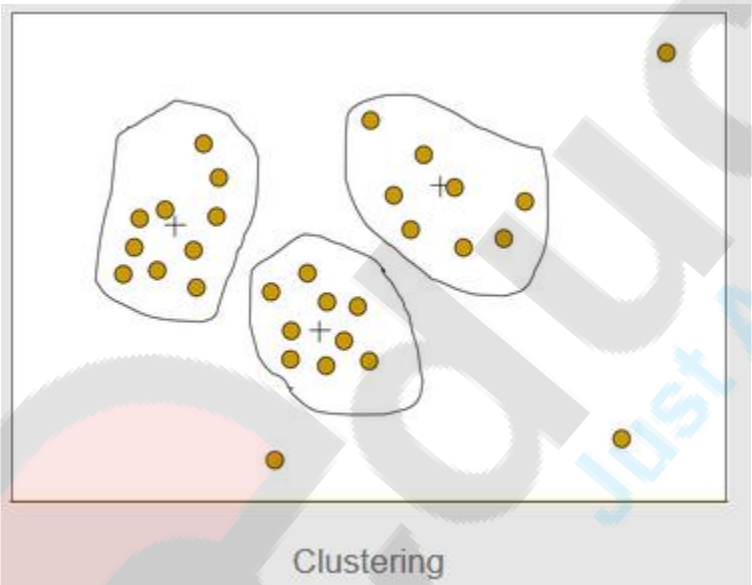
In smoothing by bin boundaries, each bin value is replaced by the closest boundary value.

Regression

Data can be smoothed by fitting the data into a regression functions.

Clustering:

Outliers may be detected by clustering,where similar values are organized into groups, or "clusters.Values that fall outside of the set of clusters may be considered outliers.



Incomplete (Missing) Data

Data is not always available

E.g., many tuples have no recorded value for several attributes, such as customer income in sales data

- Missing data may be due to
  - Equipment malfunction

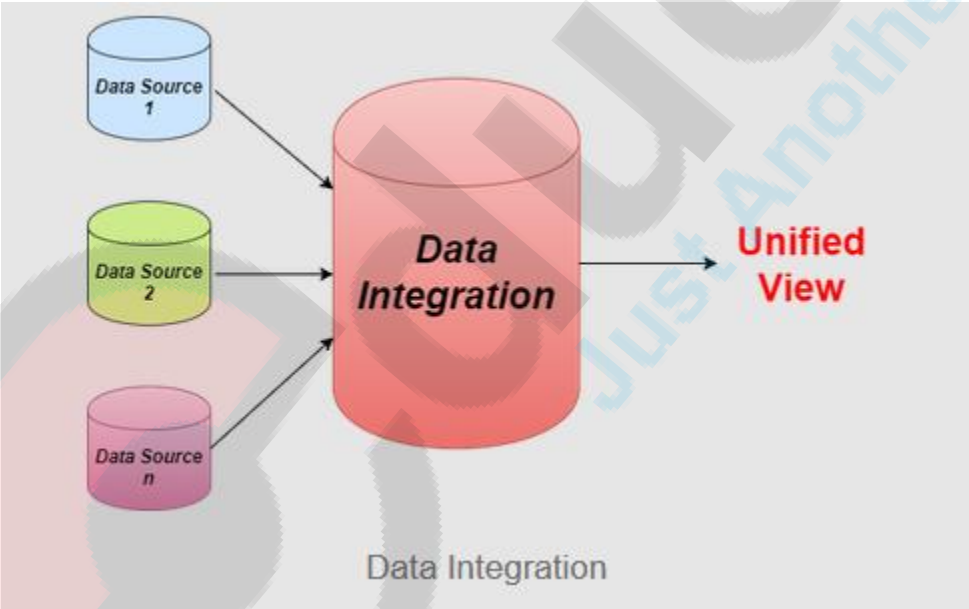
- Inconsistent with other recorded data and thus deleted
- Data not entered due to misunderstanding
- Certain data may not be considered important at the time of entry
- Not register history or changes of the data

How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing (when doing classification)—not effective when the % of missing values per attribute varies considerably
- Fill in the missing value manually: tedious + infeasible?
- Fill in it automatically with
  - A global constant : e.g., "unknown", a new class?!
  - The attribute mean
  - The attribute mean for all samples belonging to the same class: smarter
  - The most probable value: inference-based such as Bayesian formula or decision tree.

Data Integration

Data Integration is a data preprocessing technique that combines data from multiple sources and provides users a unified view of these data.



These sources may include multiple databases, data cubes, or flat files. One of the most well-known implementation of data integration is building an enterprise's data warehouse.

The benefit of a data warehouse enables a business to perform analyses based on the data in the data warehouse.

Redundant data occur often when integration of multiple databases

Object identification:

The same attribute or object may have different names in different databases

Derivable data:

One attribute may be a "derived" attribute in another table, e.g., annual revenue

Redundant attributes may be able to be detected by correlation analysis.Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality.

There are mainly 2 major approaches for data integration:-

1 Tight Coupling

In tight coupling data is combined from different sources into a single physical location through the process of ETL - Extraction, Transformation and Loading.

2 Loose Coupling

In loose coupling data only remains in the actual source databases. In this approach, an interface is provided that takes query from user and transforms it in a way the source database can understand and then sends the query directly to the source databases to obtain the result.

Data Transformation

- Smoothing: remove noise from data
- Aggregation: summarization, data cube construction
- Generalization: concept hierarchy climbing
- Normalization: scaled to fall within a small, specified range
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling
- Attribute/feature construction
  - New attributes constructed from the given ones

Data Transformation: Normalization

- Min-max normalization: to [new_min_A, new_max_A]

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

  - Ex. Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0]. Then \$73,000 is mapped to

$$\frac{73{,}600 - 12{,}000}{98{,}000 - 12{,}000}(1.0 - 0) + 0 = 0.716$$

- Z-score normalization (μ: mean, σ: standard deviation):

$$v' = \frac{v - \mu_A}{\sigma_A}$$

  - Ex. Let μ = 54,000, σ = 16,000. Then

$$\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$$

- Normalization by decimal scaling

$$v' = \frac{v}{10^j}$$

Where *j* is the smallest integer such that Max(|v'|) < 1

Data Reduction

- Why data reduction?
  - A database/data warehouse may store terabytes of data
  - Complex data analysis/mining may take a very long time to run on the complete data set

Data reduction techniques can be applied to obtain a reduced representation of the data set that is much smaller in volume but still contain critical information.

- Data reduction
  - Obtain a reduced representation of the data set that is much smaller in volume but yet produce the same (or almost the same) analytical results
- Data reduction strategies
  - Data cube aggregation:
  - Dimensionality reduction — e.g., remove unimportant attributes
  - Data Compression

- Numerosity reduction — e.g., fit data into models
- Discretization and concept hierarchy generation

Data Cube Aggregation

- The lowest level of a data cube (base cuboid)
  - The aggregated data for an individual entity of interest
  - E.g., a customer in a phone calling data warehouse
- Multiple levels of aggregation in data cubes
  - Further reduce the size of data to deal with
- Reference appropriate levels
  - Use the smallest representation which is enough to solve the task
- Queries regarding aggregated information should be answered using data cube, when possible

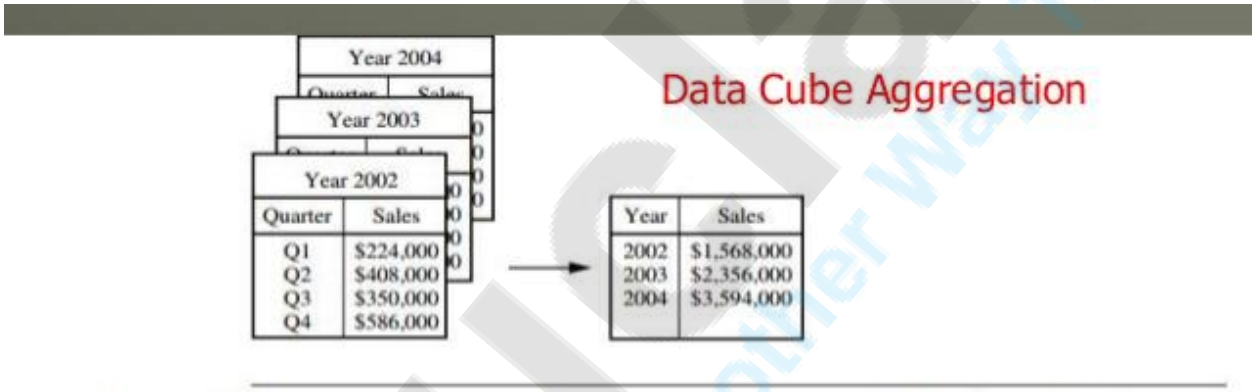
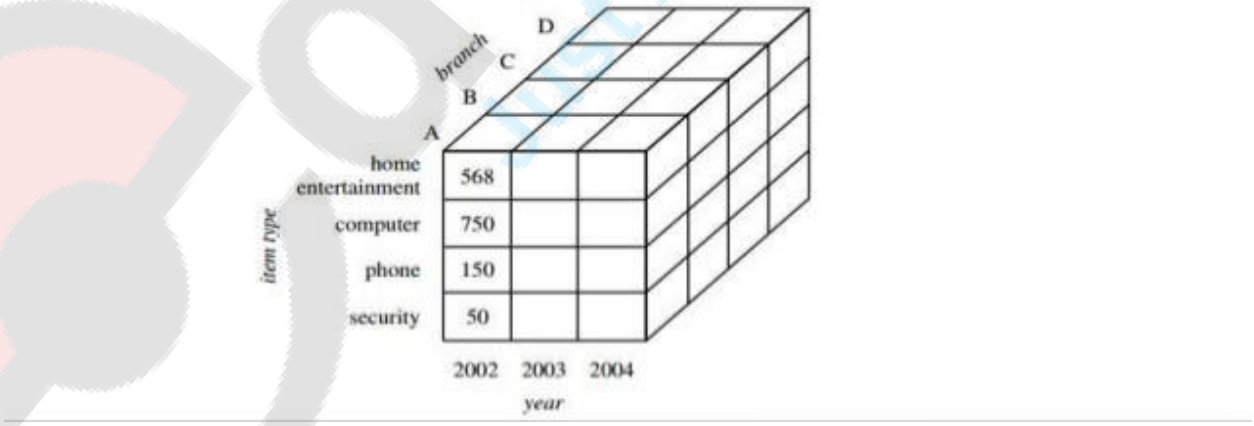


Figure 2.13 Sales data for a given branch of *AllElectronics* for the years 2002 to 2004. On the left, the sales are shown per quarter. On the right, the data are aggregated to provide the annual sales.

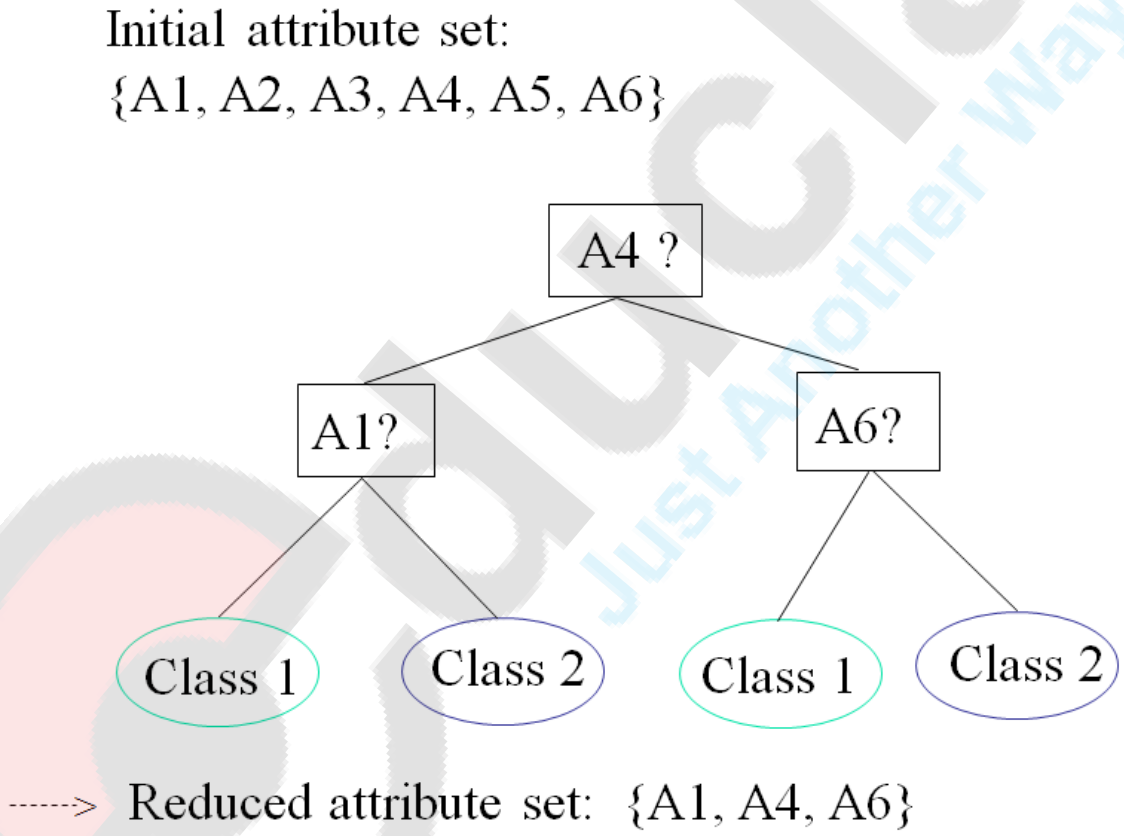


Dimensionality reduction

Attribute Subset Selection

- Feature selection (i.e., attribute subset selection):
  - Select a minimum set of features such that the probability distribution of different classes given the values for those features is as close as possible to the original distribution given the values of all features
  - reduce # of patterns in the patterns, easier to understand
- Heuristic methods (due to exponential # of choices):
  - Step-wise forward selection
  - Step-wise backward elimination
  - Combining forward selection and backward elimination
  - Decision-tree induction

Example of Decision Tree Induction



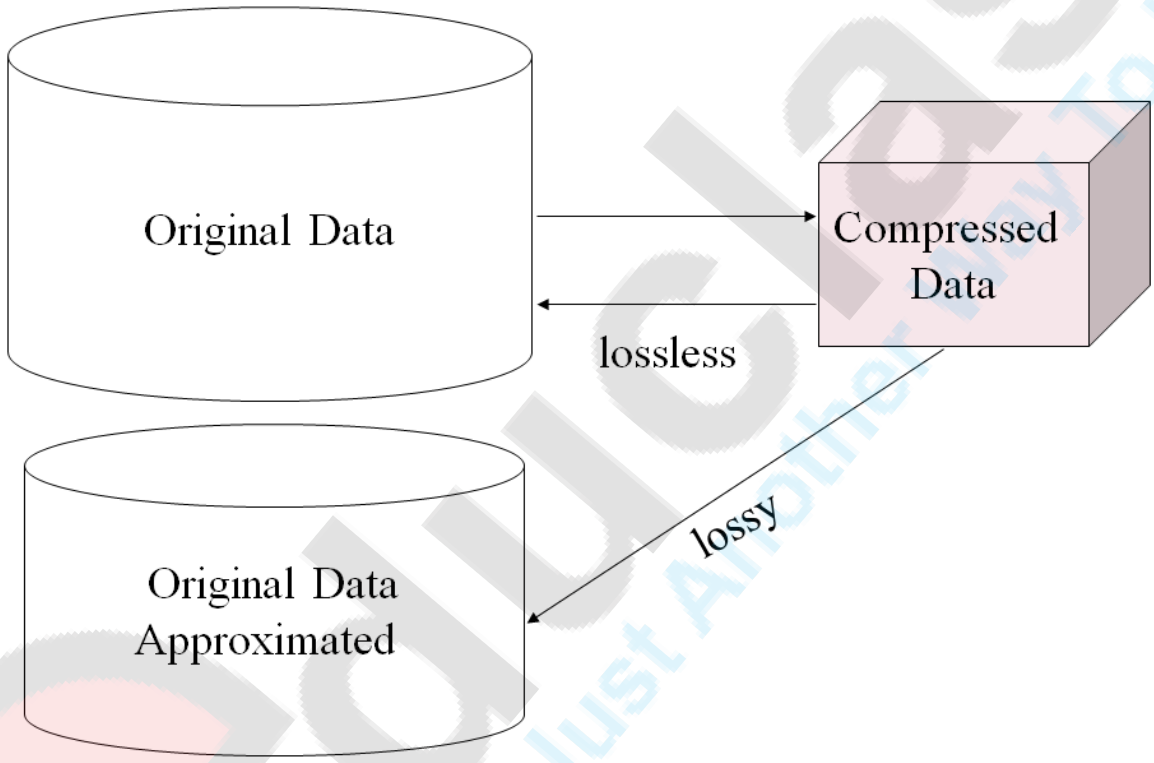
Heuristic Feature Selection Methods

- There are 2^d possible sub-features of d features
- Several heuristic feature selection methods:
  - Best single features under the feature independence assumption: choose by significance tests

- Best step-wise feature selection:
  - The best single-feature is picked first
  - Then next best feature condition to the first, ...
- Step-wise feature elimination:
  - Repeatedly eliminate the worst feature
- Best combined feature selection and elimination
- Optimal branch and bound:
  - Use feature elimination and backtracking

Data Compression

Encoding mechanisms are used to reduce the data set size.



Numerosity Reduction

- Parametric methods
  - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
- Non-parametric methods
  - Do not assume models
  - Major families: histograms, clustering, sampling

Histograms

- A popular data reduction technique
- Divide data into buckets and store average (sum) for each bucket
- Can be constructed optimally in one dimension using dynamic programming
- Related to quantization problems.

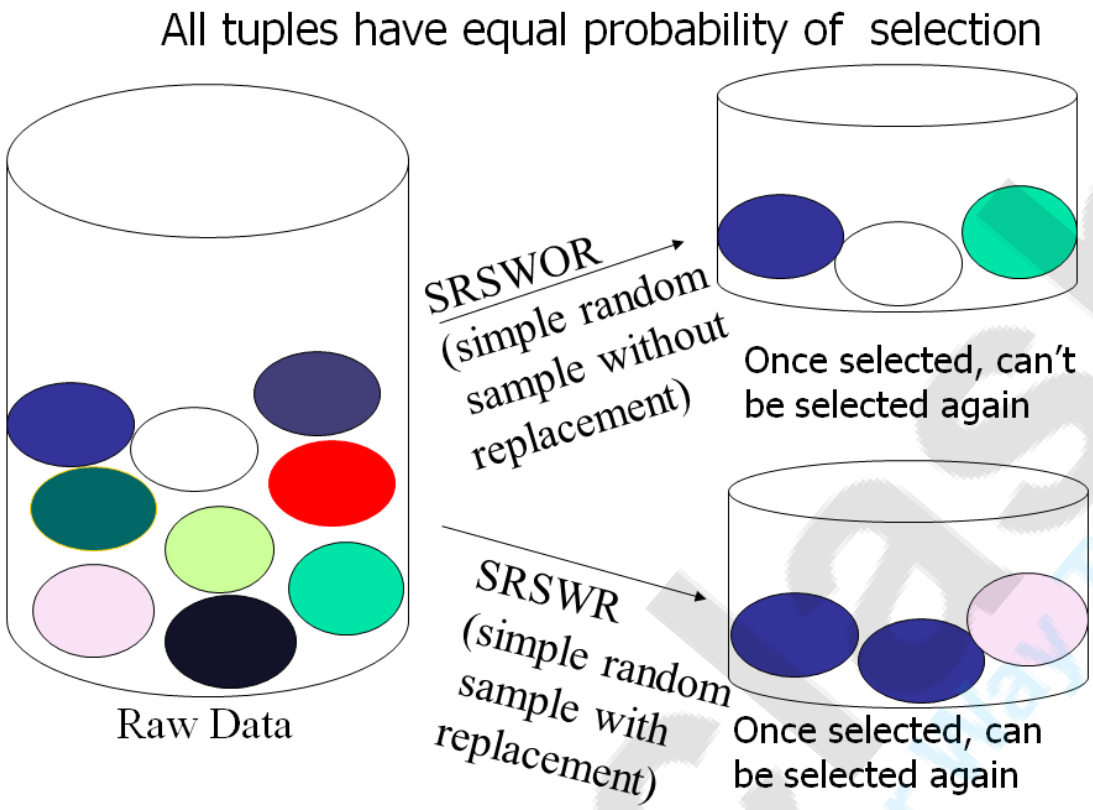
Clustering

- Partition data set into clusters, and one can store cluster representation only
- Can be very effective if data is clustered but not if data is "smeared"
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures

There are many choices of clustering definitions and clustering algorithms

Sampling

- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Choose a representative subset of the data
  - Simple random sampling may have very poor performance in the presence of skew
- Develop adaptive sampling methods
  - Stratified sampling:
    - Approximate the percentage of each class (or subpopulation of interest) in the overall database
    - Used in conjunction with skewed data
- Sampling may not reduce database I/Os (page at a time).



Data Discretization and Concept Hierarchy Generation

Data Discretization techniques can be used to divide the range of continuous attribute into intervals.Numerous continuous attribute values are replaced by small interval labels.

This leads to a concise, easy-to-use, knowledge-level representation of mining results.

Top-down discretization

If the process starts by first finding one or a few points (called split points or cut points) to split the entire attribute range, and then repeats this recursively on the resulting intervals, then it is called top-down discretization or splitting.

Bottom-up discretization

If the process starts by considering all of the continuous values as potential split-points, removes some by merging neighborhood values to form intervals, then it is called bottom-up discretization or merging.

Discretization can be performed rapidly on an attribute to provide a hierarchical partitioning of the attribute values, known as a **concept hierarchy**.

Concept hierarchies

Concept hierarchies can be used to reduce the data by collecting and replacing low-level concepts with higher-level concepts.

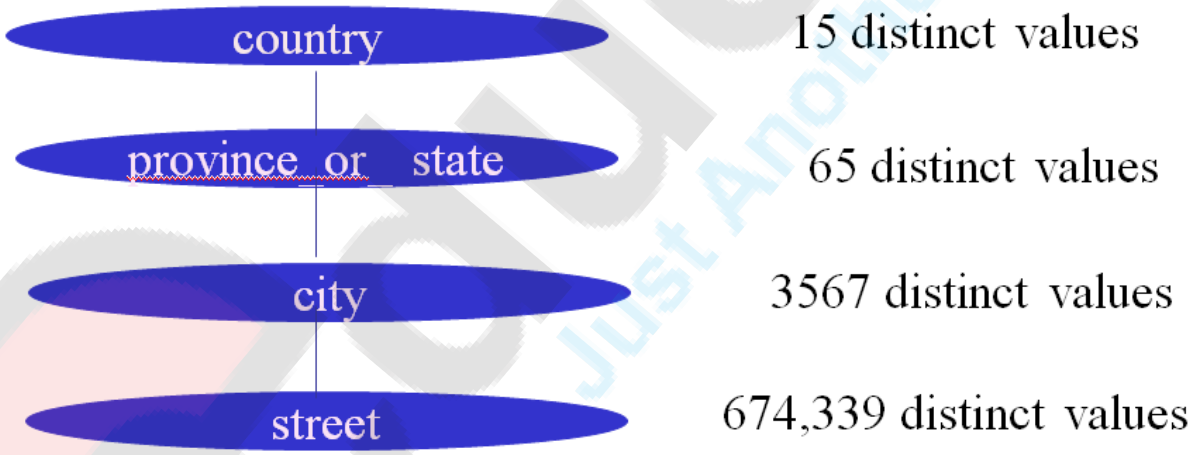
In the multidimensional model, data are organized into multiple dimensions, and each dimension contains multiple levels of abstraction defined by concept hierarchies. This organization provides users with the flexibility to view data from different perspectives.

Data mining on a reduced data set means fewer input/output operations and is more efficient than mining on a larger data set.

Because of these benefits, discretization techniques and concept hierarchies are typically applied before data mining, rather than during mining.

Specification of a set of attributes

Concept hierarchy can be automatically generated based on the number of distinct values per attribute in the given attribute set. The attribute with the most distinct values is placed at the lowest level of the hierarchy.



Discretization and Concept Hierarchy Generation for Numerical Data

Typical methods

1 Binning

Binning is a top-down splitting technique based on a specified number of bins.Binning is an unsupervised discretization technique.

2 Histogram Analysis

Because histogram analysis does not use class information so it is an unsupervised discretization technique.Histograms partition the values for an attribute into disjoint ranges called buckets.

3 Cluster Analysis

Cluster analysis is a popular data discretization method.A clustering algorithm can be applied to discrete a numerical attribute of A by partitioning the values of A into clusters or groups.

Each initial cluster or partition may be further decomposed into several subcultures, forming a lower level of the hierarchy.

Some other methods are

Entropy-Based Discretization

- Given a set of data tuples D, defined by set of attributes and class label attribute. If D is partitioned into two intervals D1 and D2 using boundary A, the entropy after partitioning is

$$E(D,A) = \frac{|D_1|}{|D} Ent(D_1) + \frac{|D_2|}{D} Ent(D_2)$$

  - D1 & D2 correspond to samples in D satisfying conditions A<split point & A>=split point

The boundary that minimizes the entropy function over all possible boundaries is selected as a binary discretization.

- The process is recursively applied to partitions obtained until some stopping criterion is met, e.g.,
- m
- Entropy(D1)= $-\sum pi \log_2(pi)$
- i=1
- Experiments show that it may reduce data size and improve classification accuracy.

-_____

- Entropy measures the amount of information in a random variable; it's the average length of the message needed to transmit an outcome of that variable using the optimal code
  - Uncertainty, Surprise, Information
  - *"High Entropy"* means X is from a uniform (boring) distribution
  - *"Low Entropy"* means X is from a varied (peaks and valleys) distribution

Playing Tennis

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Choosing an Attribute

- We want to make decisions based on one of the attributes
- There are four attributes to choose from:
  - Outlook
  - Temperature
  - Humidity
  - Wind

- What is Entropy of play?

  -5/14*log2(5/14) – 9/14*log2(9/14)
  = Entropy(5/14,9/14) = 0.9403

- Now based on Outlook, divided the set into three subsets, compute the entropy for each subset
- The expected conditional entropy is:
  5/14 * Entropy(3/5,2/5) +
  4/14 * Entropy(1,0) +
  5/14 * Entropy(3/5,2/5) = 0.6935
- So IG(Outlook) = 0.9403 – 0.6935 = 0.2468
- We seek an attribute that makes partitions as pure as possible

Information Gain in a Nutshell

$$InformationGain(A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \cdot Entropy(S_v)$$

$$Entropy = \sum_{d \in Decisions} -p(d) * \log(p(d))$$

typically yes/no

Temperature

- Now let us look at the attribute Temperature
- The expected conditional entropy is:
  4/14 * Entropy(2/4,2/4) +
  6/14 * Entropy(4/6,2/6) +
  4/14 * Entropy(3/4,1/4) = 0.9111
- So IG(Temperature) = 0.9403 – 0.9111 = 0.0292

Humidity

- Now let us look at attribute Humidity
- What is the expected conditional entropy?
- 7/14 * Entropy(4/7,3/7) +
  7/14 * Entropy(6/7,1/7) = 0.7885
- So IG(Humidity) = 0.9403 – 0.7885
  = 0.1518

Wind

- What is the information gain for wind?
- Expected conditional entropy:
  8/14 * Entropy(6/8,2/8) +
  6/14 * Entropy(3/6,3/6) = 0.8922
- IG(Wind) = 0.9403 – 0.8922 = 0.048

Information Gains

- Outlook 0.2468
- Temperature 0.0292
- Humidity 0.1518
- Wind 0.0481
- We choose Outlook since it has the highest information gain

UNIT -5

Association Rules in Data Mining

Association rules are if/then statements that are meant to find frequent patterns, correlation, and association data sets present in a relational database or other data repositories.

Example of Association Rule:-
Milk -> Bread{Support = 2%, Confidence = 60%}

A **support** of 2% for Association rule means that 2% of all the transactions show that milk and bread are purchased together(support indicates how frequently item appears in the database).

And 60% of **confidence** means 60% of all the customers who buy milk also bought bread.

Association rule is considered interesting if it satisfies both minimum support and minimum confidence threshold.

Applications of Association Rule:-

1. Market Basket data analysis.

2. Catalog design.

3. Cross marketing

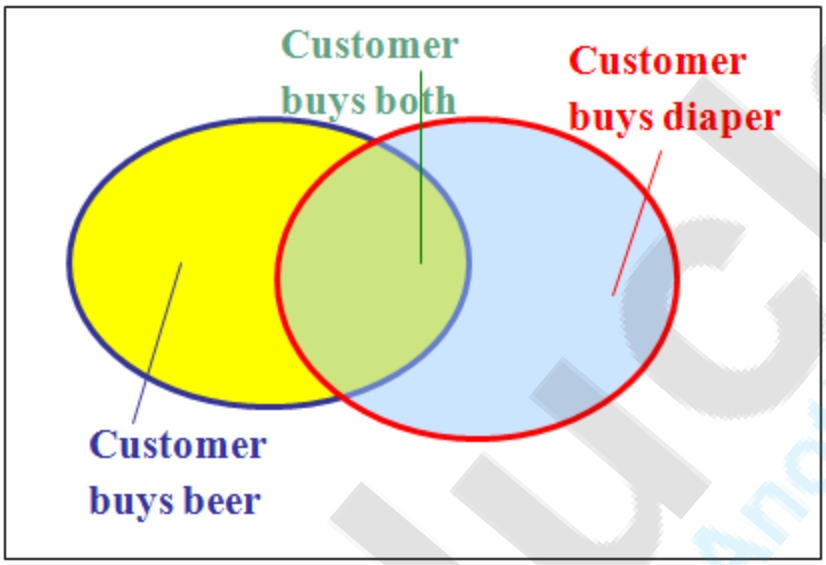
Frequent Patterns

itemset: A set of one or more items

k-itemset X = {x1, …, xk} (absolute) support, or, support count of X: Frequency or occurrence of an itemset X (relative) support, s, is the fraction of transactions that contains X (i.e., the probability that a transaction contains X)

An itemset X is frequent if X's support is no less than a minsup threshold.

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- Find all the rules $X \rightarrow Y$ with minimum support and confidence
  - support, *s*, probability that a transaction contains $X \cup Y$
  - confidence, *c*, conditional probability that a transaction having X also contains *Y*

Let minsup = 50%, minconf = 50%

Freq. Pat.: Beer:3, Nuts:3, Diaper:4, Eggs:3, {Beer, Diaper}:3

- Association rules: (many more!)
  - *Beer* → *Diaper* (60%, 100%)
  - *Diaper* → *Beer* (60%, 75%)

Market Basket Analysis

Market Basket Analysis is a modelling technique based upon the theory that if you buy a certain group of items, you are more (or less) likely to buy another group of items. For example, if you are in an English pub and you buy a pint of beer and don't buy a bar meal, you are more likely to buy crisps (US. chips) at the same time than somebody who didn't buy beer.

The set of items a customer buys is referred to as an **itemset**, and market basket analysis seeks to find relationships between purchases.

Typically the relationship will be in the form of a rule:

IF {beer, no bar meal} THEN {crisps}.

The probability that a customer will buy beer without a bar meal (i.e. that the antecedent is true) is referred to as the **support** for the rule. The conditional probability that a customer will purchase crisps is referred to as the **confidence**.

The algorithms for performing market basket analysis are fairly straightforward .The complexities mainly arise in exploiting taxonomies, avoiding combinatorial explosions (a supermarket may stock 10,000 or more line items), and dealing with the large amounts of transaction data that may be available.

A major difficulty is that a large number of the rules found may be trivial for anyone familiar with the business. Although the volume of data has been reduced, we are still asking the user to find a needle in a haystack. Requiring rules to have a high minimum support level and a high confidence level risks missing any exploitable result we might have found. One partial solution to this problem is *differential market basket analysis*,

How is it used?

In retailing, *most purchases are bought on impulse*. Market basket analysis gives clues as to what a customer might have bought *if the idea had occurred to them* .

As a first step, therefore, market basket analysis can be used in deciding the location and promotion of goods inside a store. If, as has been observed, purchasers of Barbie dolls have are more likely to buy candy, then high-margin candy can be placed near to the Barbie doll display. Customers who would have bought candy with their Barbie dolls *had they thought of it* will now be suitably tempted.

But this is only the first level of analysis. **Differential market basket analysis** can find interesting results and can also eliminate the problem of a potentially high volume of trivial results.

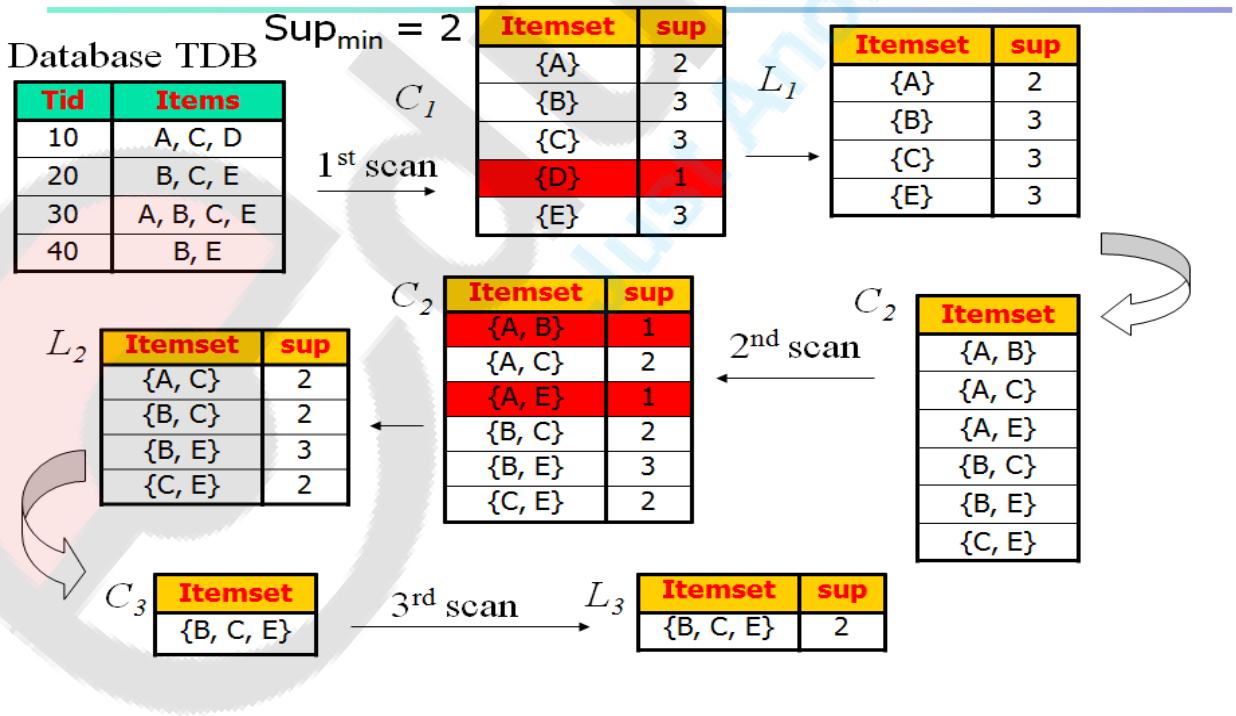
In differential analysis, we compare results between different stores, between customers in different demographic groups, between different days of the week, different seasons of the year, etc.

If we observe that a rule holds in one store, but not in any other (or does not hold in one store, but holds in all others), then we know that there is something interesting about that store. Perhaps its clientele are different, or perhaps it has organized its displays in a novel and more lucrative way. Investigating such differences may yield useful insights which will improve company sales.

Apriori: A Candidate Generation & Test Approach

- Apriori pruning principle: If there is any itemset which is infrequent, its superset should not be generated/testedMethod:
  - Initially, scan DB once to get frequent 1-itemset
  - Generate length (k+1) candidate itemsets from length k frequent itemsets
  - Test the candidates against DB
  - Terminate when no frequent or candidate set can be generated

The Apriori Algorithm—An Example



The Apriori Algorithm (Pseudo-Code)

C_k: Candidate itemset of size k

L_k : frequent itemset of size k

L_1 = {frequent items};

for (k = 1; L_k !=∅; k++) **do begin**

C_{k+1} = candidates generated from L_k;

for each transaction t in database do

increment the count of all candidates in C_{k+1} that are contained in t

L_{k+1} = candidates in C_{k+1} with min_support

end

return $\cup_k L_k$;

Example:

TID	Itemsets
T100	1 3 4
T200	2 3 5
T300	1 2 3 5
T400	2 5

Count, Support, Confidence:

Count(13)=2

|D| = 4

Support(13)=0.5

Support(3→2)=0.5

Confidence(3→2)=0.67

Implementation of Apriori

Step 1: self-joining L_k

Step 2: pruning

Example of Candidate-generation

L_3={*abc, abd, acd, ace, bcd*}

Self-joining: L_3*L_3

abcd* from *abc* and *abd

acde* from *acd* and *ace

Pruning:

***acde* is removed because *ade* is not in L_3**

C_4 = {*abcd*}

UNIT-6

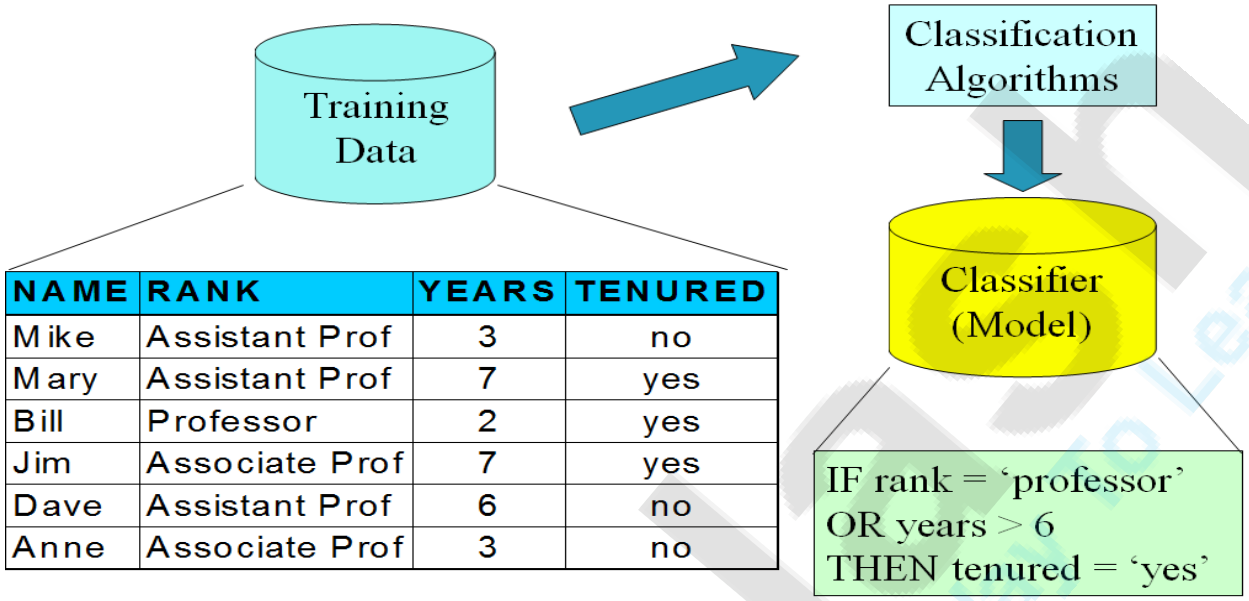
- Classification
  - predicts categorical class labels (discrete or nominal)
  - classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data

Classification—A Two-Step Process

1) Model construction: describing a set of predetermined classes
   - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
   - The set of tuples used for model construction is training set
   - The model is represented as classification rules, decision trees, or mathematical formulae
2) Model usage: for classifying future or unknown objects
   - Estimate accuracy of the model
     - The known label of test sample is compared with the classified result from the model
     - Accuracy rate is the percentage of test set samples that are correctly classified by the model
     - Test set is independent of training set (otherwise overfitting)
   - If the accuracy is acceptable, use the model to classify new data

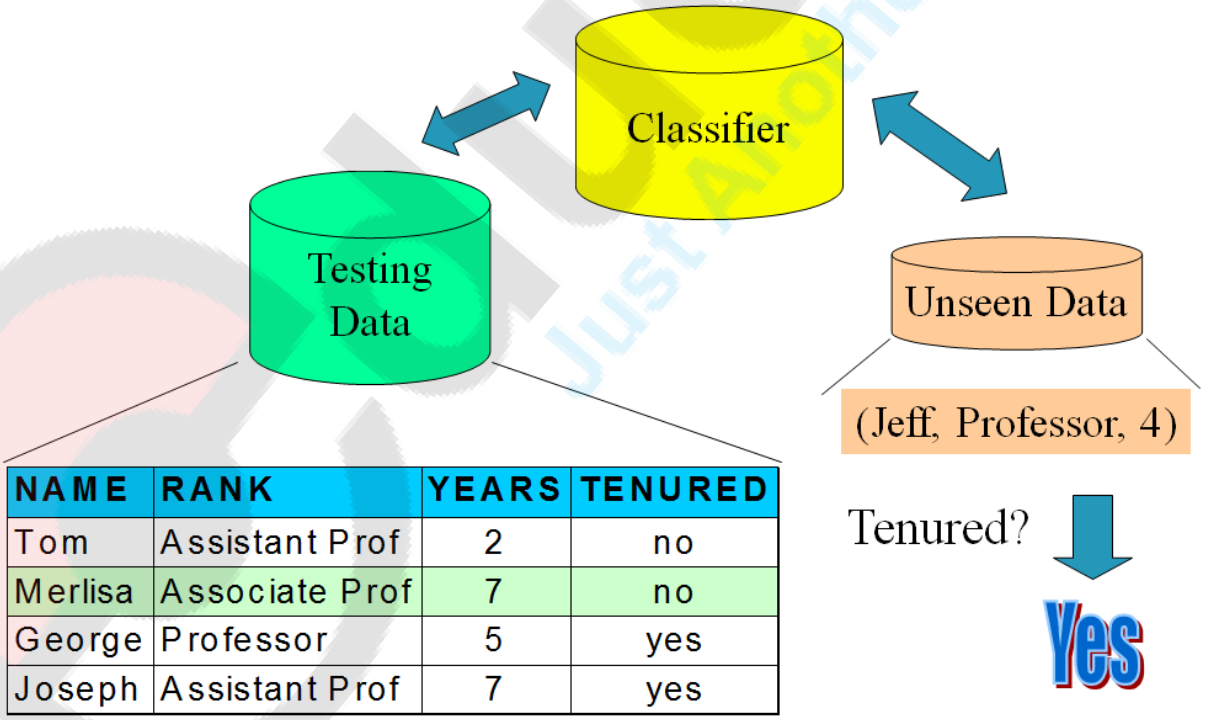
Note: If *the test set* is used to select models, it is called validation (test) set.

Process (1): Model Construction



NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Process (2): Using the Model in Prediction



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

Supervised vs. Unsupervised Learning

- Supervised learning (classification)
  - Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
  - New data is classified based on the training set
- Unsupervised learning (clustering)
  - The class labels of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm)
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they are discretized in advance)
  - Examples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)
- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left

ID3 algorithm

The ID3 algorithm begins with the original set as the root node. On each iteration of the algorithm, it iterates through every unused attribute of the set and calculates the entropy (or information gain) of that attribute. It then selects the attribute which has the smallest entropy (or largest information gain) value. The set is then split by the selected attribute (e.g. age is less than 50, age is between 50 and 100, age is greater than 100) to produce subsets of the data. The algorithm continues to recurse on each subset, considering only attributes never selected before.
Recursion on a subset may stop in one of these cases:

- every element in the subset belongs to the same class (+ or -), then the node is turned into a leaf and labelled with the class of the examples
- there are no more attributes to be selected, but the examples still do not belong to the same class (some are + and some are -), then the node is turned into a leaf and labelled with the most common class of the examples in the subset
- there are no examples in the subset, this happens when no example in the parent set was found to be matching a specific value of the selected attribute, for example if there was no example with age = 100. Then a leaf is created, and labelled with the most common class of the examples in the parent set.

Throughout the algorithm, the decision tree is constructed with each non-terminal node representing the selected attribute on which the data was split, and terminal nodes representing the class label of the final subset of this branch.

Properties

ID3 does not guarantee an optimal solution; it can get stuck in local optima. It uses a greedy approach by selecting the best attribute to split the dataset on each iteration.One improvement that can be made on the algorithm can be to use backtracking during the search for the optimal decision tree.

ID3 can overfits to the training data. To avoid overfitting, smaller decision trees should be preferred over larger ones. This algorithm usually produces small trees, but it does not always produce the smallest possible tree.

ID3 is harder to use on continuous data. If the values of any given attribute is continuous, then there are many more places to split the data on this attribute, and searching for the best value to split by can be time consuming.

Attribute Selection Measure: Information Gain

- Select the attribute with the highest information gain
- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_{i,D}|/|D|$
- Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

- Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

Attribute Selection: Information Gain

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = 0.694$$

$\frac{5}{14}I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes'es and 3 no's. Hence

age	income	student	credit rating	buys computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Similarly,

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

14

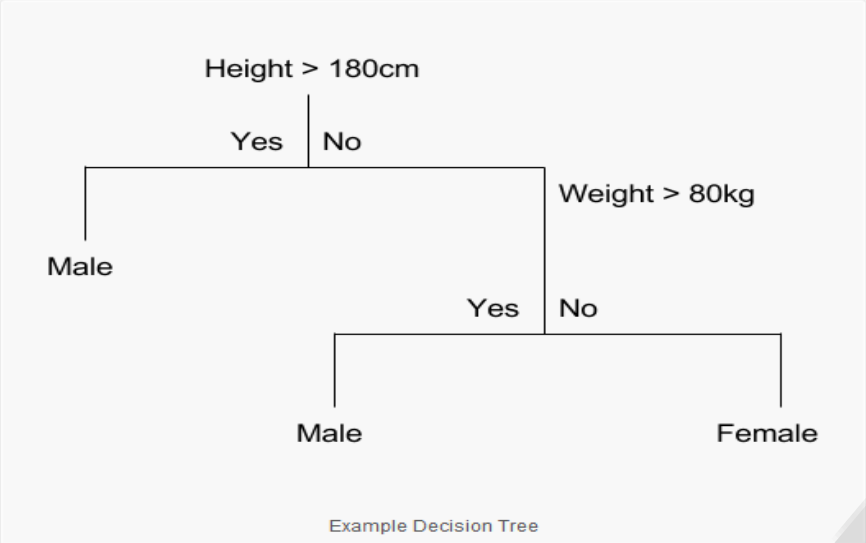
CART Model Representation

The representation for the CART model is a binary tree.

This is your binary tree from algorithms and data structures, nothing too fancy. Each root node represents a single input variable (x) and a split point on that variable (assuming the variable is numeric).

The leaf nodes of the tree contain an output variable (y) which is used to make a prediction.

Given a dataset with two inputs (x) of height in centimeters and weight in kilograms the output of sex as male or female, below is a crude example of a binary decision tree (completely fictitious for demonstration purposes only).



Example Decision Tree

The tree can be stored to file as a graph or a set of rules. For example, below is the above decision tree as a set of rules.

If Height > 180 cm Then Male

If Height <= 180 cm AND Weight > 80 kg Then Male

If Height <= 180 cm AND Weight <= 80 kg Then Female

Make Predictions With CART Models

With the binary tree representation of the CART model described above, making predictions is relatively straightforward.

Given a new input, the tree is traversed by evaluating the specific input started at the root node of the tree.

A learned binary tree is actually a partitioning of the input space. You can think of each input variable as a dimension on a p-dimensional space. The decision tree split this up into rectangles (when p=2 input variables) or some kind of hyper-rectangles with more inputs.

New data is filtered through the tree and lands in one of the rectangles and the output value for that rectangle is the prediction made by the model. This gives you some feeling for the type of decisions that a CART model is capable of making, e.g. boxy decision boundaries.

Learn a CART Model From Data

Creating a CART model involves selecting input variables and split points on those variables until a suitable tree is constructed. The selection of which input variable to use and the specific

split or cut-point is chosen using a greedy algorithm to minimize a cost function. Tree construction ends using a predefined stopping criterion, such as a minimum number of training instances assigned to each leaf node of the tree.

Greedy Splitting

Creating a binary decision tree is actually a process of dividing up the input space. A greedy approach is used to divide the space called recursive binary splitting. This is a numerical procedure where all the values are lined up and different split points are tried and tested using a cost function. The split with the best cost (lowest cost because we minimize cost) is selected.

All input variables and all possible split points are evaluated and chosen in a greedy manner (e.g. the very best split point is chosen each time).

For regression predictive modeling problems the cost function that is minimized to choose split points is the sum squared error across all training samples that fall within the rectangle:

$$\text{sum}(y - \text{prediction})^2$$

Where y is the output for the training sample and prediction is the predicted output for the rectangle.

For classification the Gini index function is used which provides an indication of how "pure" the leaf nodes are (how mixed the training data assigned to each node is).

$$G = \text{sum}(pk * (1 - pk))$$

Where G is the Gini index over all classes, pk are the proportion of training instances with class k in the rectangle of interest. A node that has all classes of the same type (perfect class purity) will have G=0, where as a G that has a 50-50 split of classes for a binary classification problem (worst purity) will have a G=0.5.

For a binary classification problem, this can be re-written as:

$$G = 2 * p1 * p2$$
or
$$G = 1 - (p1^2 + p2^2)$$

The Gini index calculation for each node is weighted by the total number of instances in the parent node. The Gini score for a chosen split point in a binary classification problem is therefore calculated as follows:

$$G = ((1 - (g1\_1^2 + g1\_2^2)) * (ng1/n)) + ((1 - (g2\_1^2 + g2\_2^2)) * (ng2/n))$$

Where G is the Gini index for the split point, g1_1 is the proportion of instances in group 1 for class 1, g1_2 for class 2, g2_1 for group 2 and class 1, g2_2 group 2 class 2, ng1 and ng2 are the total number of instances in group 1 and 2 and n are the total number of instances we are trying to group from the parent node.

Stopping Criterion

The recursive binary splitting procedure described above needs to know when to stop splitting as it works its way down the tree with the training data.

The most common stopping procedure is to use a minimum count on the number of training instances assigned to each leaf node. If the count is less than some minimum then the split is not accepted and the node is taken as a final leaf node.

The count of training members is tuned to the dataset, e.g. 5 or 10. It defines how specific to the training data the tree will be. Too specific (e.g. a count of 1) and the tree will overfit the training data and likely have poor performance on the test set.

Pruning the Tree

The stopping criterion is important as it strongly influences the performance of your tree. You can use pruning after learning your tree to further lift performance.

The complexity of a decision tree is defined as the number of splits in the tree. Simpler trees are preferred. They are easy to understand (you can print them out and show them to subject matter experts), and they are less likely to overfit your data.

The fastest and simplest pruning method is to work through each leaf node in the tree and evaluate the effect of removing it using a hold-out test set. Leaf nodes are removed only if it results in a drop in the overall cost function on the entire test set. You stop removing nodes when no further improvements can be made.

More sophisticated pruning methods can be used such as cost complexity pruning (also called weakest link pruning) where a learning parameter (alpha) is used to weigh whether nodes can be removed based on the size of the sub-tree.

Learn a CART Model from Data

Creating a CART model involves selecting input variables and split points on those variables until a suitable tree is constructed.

The selection of which input variable to use and the specific split or cut-point is chosen using a greedy algorithm to minimize a cost function. Tree construction ends using a predefined

stopping criterion, such as a minimum number of training instances assigned to each leaf node of the tree.

Greedy Splitting

Creating a binary decision tree is actually a process of dividing up the input space. A greedy approach is used to divide the space called recursive binary splitting.

This is a numerical procedure where all the values are lined up and different split points are tried and tested using a cost function. The split with the best cost (lowest cost because we minimize cost) is selected.

All input variables and all possible split points are evaluated and chosen in a greedy manner (e.g. the very best split point is chosen each time).

For regression predictive modeling problems the cost function that is minimized to choose split points is the sum squared error across all training samples that fall within the rectangle:

$$\text{sum}(y - \text{prediction})^2$$

Where y is the output for the training sample and prediction is the predicted output for the rectangle.

For classification the Gini index function is used which provides an indication of how "pure" the leaf nodes are (how mixed the training data assigned to each node is).

$$G = \text{sum}(pk * (1 - pk))$$

Where G is the Gini index over all classes, pk are the proportion of training instances with class k in the rectangle of interest. A node that has all classes of the same type (perfect class purity) will have G=0, where as a G that has a 50-50 split of classes for a binary classification problem (worst purity) will have a G=0.5.

For a binary classification problem, this can be re-written as:

$$G = 2 * p1 * p2$$
or
$$G = 1 - (p1^2 + p2^2)$$

The Gini index calculation for each node is weighted by the total number of instances in the parent node. The Gini score for a chosen split point in a binary classification problem is therefore calculated as follows:

$$G = ((1 - (g1\_1^2 + g1\_2^2)) * (ng1/n)) + ((1 - (g2\_1^2 + g2\_2^2)) * (ng2/n))$$

Where G is the Gini index for the split point, g1_1 is the proportion of instances in group 1 for class 1, g1_2 for class 2, g2_1 for group 2 and class 1, g2_2 group 2 class 2, ng1 and ng2 are the total number of instances in group 1 and 2 and n are the total number of instances we are trying to group from the parent node.

Stopping Criterion

The recursive binary splitting procedure described above needs to know when to stop splitting as it works its way down the tree with the training data.

The most common stopping procedure is to use a minimum count on the number of training instances assigned to each leaf node. If the count is less than some minimum then the split is not accepted and the node is taken as a final leaf node.

The count of training members is tuned to the dataset, e.g. 5 or 10. It defines how specific to the training data the tree will be. Too specific (e.g. a count of 1) and the tree will overfit the training data and likely have poor performance on the test set.

Pruning the Tree

The stopping criterion is important as it strongly influences the performance of your tree. You can use pruning after learning your tree to further lift performance.

The complexity of a decision tree is defined as the number of splits in the tree. Simpler trees are preferred. They are easy to understand (you can print them out and show them to subject matter experts), and they are less likely to overfit your data.

The fastest and simplest pruning method is to work through each leaf node in the tree and evaluate the effect of removing it using a hold-out test set. Leaf nodes are removed only if it results in a drop in the overall cost function on the entire test set. You stop removing nodes when no further improvements can be made.

More sophisticated pruning methods can be used such as cost complexity pruning (also called weakest link pruning) where a learning parameter (alpha) is used to weigh whether nodes can be removed based on the size of the sub-tree.

Bayesian Classification

A statistical classifier: performs probabilistic prediction, i.e., predicts class membership probabilities

Foundation: Based on Bayes' Theorem.

Performance: A simple Bayesian classifier, naïve Bayesian classifier, has comparable performance with decision tree and selected neural network classifiers

Incremental: Each training example can incrementally increase/decrease the probability that a hypothesis is correct — prior knowledge can be combined with observed data

Standard: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured.

Bayes' Theorem: Basics

- Total probability Theorem: $P(B) = \sum_{i=1}^{M} P(B|A_i)P(A_i)$
- Bayes' Theorem: $P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})} = P(\mathbf{X}|H) \times P(H) / P(\mathbf{X})$
  - Let **X** be a data sample ("*evidence*"): class label is unknown
  - Let H be a *hypothesis* that X belongs to class C
  - Classification is to determine P(H|**X**), (i.e., *posteriori probability):* the probability that the hypothesis holds given the observed data sample **X**
  - P(H) (*prior probability*): the initial probability
    - E.g., **X** will buy computer, regardless of age, income, …
  - P(**X**): probability that sample data is observed
  - P(**X**|H) (likelihood): the probability of observing the sample **X**, given that the hypothesis holds
    - E.g., Given that **X** will buy computer, the prob. that X is 31..40, medium income

Prediction Based on Bayes' Theorem

- Given training data **X**, *posteriori probability of a hypothesis* H, P(H|**X**), follows the Bayes' theorem

$$P(H \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid H)P(H)}{P(\mathbf{X})} = P(\mathbf{X} \mid H) \times P(H) / P(\mathbf{X})$$

- Informally, this can be viewed as

  posteriori = likelihood x prior/evidence

- Predicts **X** belongs to C_i iff the probability $P(C_i|\mathbf{X})$ is the highest among all the $P(C_k|X)$ for all the *k* classes
- Practical difficulty: It requires initial knowledge of many probabilities, involving significant computational cost

Classification Is to Derive the Maximum Posteriori

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n-D attribute vector **X** = (x_1, x_2, …, x_n)
- Suppose there are *m* classes C_1, C_2, …, C_m.
- Classification is to derive the maximum posteriori, i.e., the maximal $P(C_i|\mathbf{X})$
- This can be derived from Bayes' theorem

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

- Since P(X) is constant for all classes, only

$$P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$$

needs to be maximized

Naïve Bayes Classifier

- A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):

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

- This greatly reduces the computation cost: Only counts the class distribution
- If A_k is categorical, $P(x_k|C_i)$ is the # of tuples in C_i having value x_k for A_k divided by $|C_{i,D}|$ (# of tuples of C_i in D)
- If A_k is continous-valued, $P(x_k|C_i)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ

and $P(x_k|C_i)$ is

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

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

Naïve Bayes Classifier: Training Dataset

Class:
C1:buys_computer = 'yes'
C2:buys_computer = 'no'

Data to be classified:
X = (age <=30,
Income = medium,
Student = yes
Credit_rating = Fair)

age	income	student	credit_rating	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Naïve Bayes Classifier: An Example

- $P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357
- Compute $P(X|C_i)$ for each class
  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4
- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

$P(X|C_i)*P(C_i)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")

Avoiding the Zero-Probability Problem

- Naïve Bayesian prediction requires each conditional prob. be **non-zero**. Otherwise, the predicted prob. will be zero

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

- Ex. Suppose a dataset with 1000 tuples, income=low (0), income= medium (990), and income = high (10)
- Use **Laplacian correction** (or Laplacian estimator)
  - *Adding 1 to each case*

    Prob(income = low) = 1/1003

    Prob(income = medium) = 991/1003

    Prob(income = high) = 11/1003
  - The "corrected" prob. estimates are close to their "uncorrected" counterparts

Advantages and disadvantages of Naïve Bayes Classifier

- Advantages
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., hospitals: patients: Profile: age, family history, etc. Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayes Classifier

Model Evaluation and Selection

- Evaluation metrics: How can we measure accuracy? Other metrics to consider?
- Use validation test set of class-labeled tuples instead of training set when assessing accuracy
- Methods for estimating a classifier's accuracy:
  - Holdout method, random subsampling
  - Cross-validation
  - Bootstrap
- Comparing classifiers:
  - Confidence intervals
  - Cost-benefit analysis and ROC Curves

Classifier Evaluation Metrics: Confusion Matrix

Confusion Matrix:

Actual class\Predicted class	C_1	$\neg C_1$
C_1	**True Positives (TP)**	**False Negatives (FN)**
$\neg C_1$	**False Positives (FP)**	**True Negatives (TN)**

Example of Confusion Matrix:

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

- Given *m* classes, an entry, $CM_{i,j}$ in a **confusion matrix** indicates # of tuples in class *i* that were labeled by the classifier as class *j*
- May have extra rows/columns to provide totals

Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

- **Classifier Accuracy,** or recognition rate: percentage of test set tuples that are correctly classified

  Accuracy = (TP + TN)/All
- **Error rate:** *1 – accuracy*, or

  Error rate = (FP + FN)/All

- **Class Imbalance Problem**:
  - One class may be *rare*, e.g. fraud, or HIV-positive
  - Significant *majority of the negative class* and minority of the positive class
  - **Sensitivity**: True Positive recognition rate
    - **Sensitivity = TP/P**
  - **Specificity**: True Negative recognition rate
    - **Specificity = TN/N**

Classifier Evaluation Metrics:

Precision and Recall, and F-measures

- **Precision**: exactness – what % of tuples that the classifier labeled as positive are actually positive

$$precision = \frac{TP}{TP + FP}$$

- **Recall:** completeness – what % of positive tuples did the classifier label as positive?

$$recall = \frac{TP}{TP + FN}$$

- Perfect score is 1.0
- Inverse relationship between precision & recall
- ***F* measure (*F_1* or *F*-score)**: harmonic mean of precision and recall,

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

- ***F_β*:** weighted measure of precision and recall
  - assigns ß times as much weight to recall as to precision

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$

Classifier Evaluation Metrics: Example

Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	**90**	**210**	300	30.00 (*sensitivity*
cancer = no	**140**	**9560**	9700	98.56 (*specificity)*
Total	230	9770	10000	96.40 (*accuracy*)

- *Precision* = 90/230 = 39.13% *Recall* = 90/300 = 30.00%

Evaluating Classifier Accuracy:
Holdout & Cross-Validation Methods

- **Holdout method**
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation
  - Random sampling: a variation of holdout
    - Repeat holdout k times, accuracy = avg. of the accuracies obtained
- **Cross-validation** (*k*-fold, where k = 10 is most popular)
  - Randomly partition the data into *k mutually exclusive* subsets, each approximately equal size
  - At *i*-th iteration, use D_i as test set and others as training set
  - Leave-one-out: *k* folds where *k* = # of tuples, for small sized data
  - ***Stratified cross-validation***: folds are stratified so that class dist. in each fold is approx. the same as that in the initial data

What Is Prediction?

- (Numerical) prediction is similar to classification
  - construct a model
  - use model to predict continuous or ordered value for a given input
- Prediction is different from classification
  - Classification refers to predict categorical class label
  - Prediction models continuous-valued functions
- Major method for prediction: regression
  - model the relationship between one or more *independent* or **predictor** variables and a *dependent* or **response** variable
- Regression analysis
  - Linear and multiple regression
  - Non-linear regression
  - Other regression methods: generalized linear model, Poisson regression, log-linear models, regression trees

Linear Regression

Regression is a data mining technique used to predict a range of numeric values (also called *continuous values*), given a particular dataset. For example, regression might be used to predict the cost of a product or service, given other variables.

Regression is used across multiple industries for business and marketing planning, financial forecasting, environmental modeling and analysis of trends.

Regression vs. Classification

Regression and classification are data mining techniques used to solve similar problems, but they are frequently confused.

Both are used in prediction analysis, but regression is used to predict a numeric or continuous value while classification assigns data into discrete categories.

For example, regression would be used to predict a home's value based on its location, square feet, price when last sold, the price of similar homes, and other factors. Classification would be in order if you wanted to instead organize houses into categories, such as walkability, lot size or crime rates.

Types of Regression Techniques

The simplest and oldest form of regression is linear regression used to estimate a relationship between two variables. This technique uses the mathematical formula of a straight line (y = mx + b). In plain terms, this simply means that, given a graph with a Y and an X-axis, the relationship between X and Y is a straight line with few outliers. For example, we might assume that, given an increase in population, food production would increase at the same rate — this requires a strong, linear relationship between the two figures.

To visualize this, consider a graph in which the Y-axis tracks population increase, and the X-axis tracks food production. As the Y value increases, the X value would increase at the same rate, making the relationship between them a straight line.

Advanced techniques, such as multiple regression, predict a relationship between multiple variables- for example, is there a correlation between income, education and where one chooses to live?

The addition of more variables considerably increases the complexity of the prediction. There are several types of multiple regression techniques including standard, hierarchical, setwise and stepwise, each with its own application.

At this point, it's important to understand what we are trying to predict (the dependent or *predicted* variable) and the data we are using to make the prediction (the independent or *predictor* variables). In our example, we want to predict the location where one chooses to live (the *predicted* variable) given income and education (both *predictor* variables).

- Linear regression: involves a response variable y and a single predictor variable x

  $y = w_0 + w_1 x$

  where w_0 (y-intercept) and w_1 (slope) are regression coefficients
- Method of least squares: estimates the best-fitting straight line

$$w_1 = \frac{\sum_{i=1}^{|D|}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|}(x_i - \bar{x})^2} \qquad w_0 = \bar{y} - w_1 \bar{x}$$

- Multiple linear regression: involves more than one predictor variable
  - Training data is of the form $(\mathbf{X_1}, y_1)$, $(\mathbf{X_2}, y_2)$,…, $(\mathbf{X_{|D|}}, y_{|D|})$
  - Ex. For 2-D data, we may have: $y = w_0 + w_1 x_1 + w_2 x_2$
  - Solvable by extension of least square method or using SAS, S-Plus
  - Many nonlinear functions can be transformed into the above

Nonlinear Regression

- Some nonlinear models can be modeled by a polynomial function
- A polynomial regression model can be transformed into linear regression model. For example,

  $y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$

  convertible to linear with new variables: $x_2 = x^2$, $x_3 = x^3$

  $y = w_0 + w_1 x + w_2 x_2 + w_3 x_3$
- Other functions, such as power function, can also be transformed to linear model
- Some models are intractable nonlinear (e.g., sum of exponential terms)
  - possible to obtain least square estimates through extensive calculation on more complex formulae

UNIT-7

What is Clustering?

Clustering is the process of making a group of abstract objects into classes of similar objects.

Points to Remember

- A cluster of data objects can be treated as one group.
- While doing cluster analysis, we first partition the set of data into groups based on data similarity and then assign the labels to the groups.
- The main advantage of clustering over classification is that, it is adaptable to changes and helps single out useful features that distinguish different groups.

Applications of Cluster Analysis

- Clustering analysis is broadly used in many applications such as market research, pattern recognition, data analysis, and image processing.
- Clustering can also help marketers discover distinct groups in their customer base. And they can characterize their customer groups based on the purchasing patterns.
- In the field of biology, it can be used to derive plant and animal taxonomies, categorize genes with similar functionalities and gain insight into structures inherent to populations.
- Clustering also helps in identification of areas of similar land use in an earth observation database. It also helps in the identification of groups of houses in a city according to house type, value, and geographic location.
- Clustering also helps in classifying documents on the web for information discovery.
- Clustering is also used in outlier detection applications such as detection of credit card fraud.
- As a data mining function, cluster analysis serves as a tool to gain insight into the distribution of data to observe characteristics of each cluster.

Clustering Methods

Clustering methods can be classified into the following categories −

- Partitioning Method
- Hierarchical Method
- Model-Based Method

Partitioning Method

Suppose we are given a database of 'n' objects and the partitioning method constructs 'k' partition of data. Each partition will represent a cluster and k ≤ n. It means that it will classify the data into k groups, which satisfy the following requirements −

- Each group contains at least one object.
- Each object must belong to exactly one group.

Points to remember –

- For a given number of partitions (say k), the partitioning method will create an initial partitioning.
- Then it uses the iterative relocation technique to improve the partitioning by moving objects from one group to other.

The *K-Means* Clustering Method

Given *k*, the *k-means* algorithm is implemented in four steps:

- Partition objects into *k* nonempty subsets
- Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., *mean point*, of the cluster)
- Assign each object to the cluster with the nearest seed point
- Go back to Step 2, stop when the assignment does not change

The simplest and most commonly used algorithm, employing a squared error criterion is the K-means algorithm. This algorithm partitions the data into K clusters (C1, C2, . . . ,CK) ,represented by their centers or means. The cen-ter of each cluster is calculated as the mean of all the instances belonging to that cluster.

K-means algorithm

The algo-rithm starts with an initial set of cluster centers, chosen at random or according to some heuristic procedure. In each iteration, each instance is assigned to its nearest cluster center according to the Euclidean distance between the two. Then the cluster centers are re-calculated. The center of each cluster is calculated as the mean of all the instances belonging to that cluster:

$$\mu_k = \frac{1}{N_k} \sum_{q=1}^{N_k} x_q$$

where N_k is the number of instances belonging to cluster k and μ_k is the mean of the cluster k.

A number of convergence conditions are possible. For example, the search may stop when the partitioning error is not reduced by the relocation of the centers. This indicates that the present partition is locally optimal. Other stopping criteria can be used also such as exceeding a pre-defined number of iterations.

Input: S (instance set), K (number of cluster)
Output: clusters
1: Initialize K cluster centers.
2: **while** termination condition is not satisfied **do**
3: Assign instances to the closest cluster center.
4: Update cluster centers based on the assignment.
5: **end while**

Hierarchical Methods

These methods construct the clusters by recursively partitioning the instances in either a top-down or bottom-up fashion. These methods can be sub-divided as following:

Agglomerative hierarchical clustering - Each object initially represents a cluster of its own. Then clusters are successively merged until the desired cluster structure is obtained.
Divisive hierarchical clustering — All objects initially belong to one cluster. Then the cluster is divided into sub-clusters, which are succes- sively divided into their own sub-clusters. This process continues until the desired cluster structure is obtained.
The result of the hierarchical methods is a dendrogram, representing the nested grouping of objects and similarity levels at which groupings change. A clustering of the data objects is obtained by cutting the dendrogram at the desired similarity level.

The merging or division of clusters is performed according to some similarity measure, chosen so as to optimize some criterion (such as a sum of squares).
The hierarchical clustering methods could be further divided according to the manner that the similarity measure is calculated.

Single-link clustering (also called the connectedness, the minimum method or the nearest neighbor method):- Methods that consider the distance between two clusters to be equal to the shortest distance from any member of one cluster to any member of the other cluster. If the data consist of similarities, the similarity between a pair of clusters is considered to be equal to the greatest similarity from any member of one cluster to any member of the other cluster.

Complete-link clustering (also called the diameter, the maximum method or the furthest neighbor method) - methods that consider the distance between two clusters to be equal to the longest distance from any member of one cluster to any member of the other cluster.

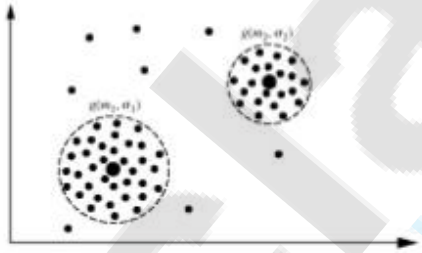
Average-link clustering (also called minimum variance method) - methods that consider the distance between two clusters to be equal to the average distance from any member of one cluster to any member of the other cluster.

Model-Based Clustering Methods

- Attempt to optimize the fit between the data and some mathematical model
- Assumption: Data are generated by a mixture of underlying probability distributions
- Techniques
  - Expectation-Maximization
  - Conceptual Clustering
  - Neural Networks Approach

Expectation Maximization

- Each cluster is represented mathematically by a parametric probability distribution
  - Component distribution
  - Data is a mixture of these distributions
  - Mixture density model
  - Problem: To estimate parameters of probability distributions



Expectation Maximization

- Iterative Refinement Algorithm – used to find parameter estimates
- Extension of k-means
  - Assigns an object to a cluster according to a weight representing probability of membership
- Initial estimate of parameters
- Iteratively reassigns scores

Expectation Maximization

- Initial guess for parameters; randomly select k objects to represent cluster means or centers
- Iteratively refine parameters / clusters
  - Expectation Step
    - Assign each object x to cluster C_k with probability

$$P(\boldsymbol{x}_i \in C_k) = p(C_k|\boldsymbol{x}_i) = \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)},$$

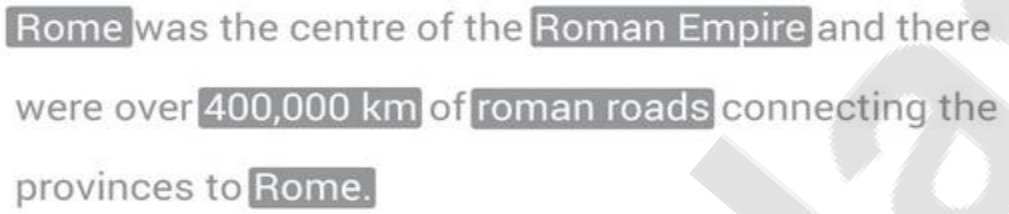
    where $p(\boldsymbol{x}_i|C_k) = N(m_k, E_k(\boldsymbol{x}_i))$
  - Maximization Step
    - Re-estimate model parameters $m_k = \frac{1}{n}\sum_{i=1}^{n}\frac{\boldsymbol{x}_i P(\boldsymbol{x}_i \in C_k)}{\sum_j P(\boldsymbol{x}_i \in C_j)}.$
- Simple and easy to implement
- Complexity depends on features, objects and iterations

UNIT -8

What is Text Analysis?

Text analysis is about parsing texts in order to extract machine-readable facts from them. The purpose of text analysis is to create sets of structured data out of heaps of unstructured, heterogeneous documents.

The process can be thought of as slicing and dicing documents into easy-to-manage and integrate data pieces.



After sentences are split, the important concepts and entities (i.e.the proper nouns) are identified through dictionary word lists.

For example, through text analysis the text in the sentence ***Rome was the centre of the Roman Empire and there were over 400 000 km of Roman roads connecting the provinces to Rome*** is divided into small chunks, which are further classified. This is done by algorithms that first parse the textual content and then extract salient facts about pre-specified types of events, people, things, entities or relationships.

Often the purpose of text analysis is semantic annotation, which overarching goal is to allow easy-to-automate operations related to textual sources.

Text databases consist of huge collection of documents. They collect these information from several sources such as news articles, books, digital libraries, e-mail messages, web pages, etc. Due to increase in the amount of information, the text databases are growing rapidly. In many of the text databases, the data is semi-structured.

For example, a document may contain a few structured fields, such as title, author, publishing_date, etc. But along with the structure data, the document also contains unstructured text components, such as abstract and contents. Without knowing what could be in the documents, it is difficult to formulate effective queries for analyzing and extracting useful information from the data. Users require tools to compare the documents and rank their importance and relevance. Therefore, text mining has become popular and an essential theme in data mining.

Information Retrieval

Information retrieval deals with the retrieval of information from a large number of text-based documents. Some of the database systems are not usually present in information retrieval systems because both handle different kinds of data. Examples of information retrieval system include −

- Online Library catalogue system
- Online Document Management Systems
- Web Search Systems etc.

Note −The main problem in an information retrieval system is to locate relevant documents in a document collection based on a user's query. This kind of user's query consists of some keywords describing an information need.

In such search problems, the user takes an initiative to pull relevant information out from a collection. This is appropriate when the user has ad-hoc information need, i.e., a short-term need. But if the user has a long-term information need, then the retrieval system can also take an initiative to push any newly arrived information item to the user.

This kind of access to information is called Information Filtering. And the corresponding systems are known as Filtering Systems or Recommender Systems.

Basic Measures for Text Retrieval

We need to check the accuracy of a system when it retrieves a number of documents on the basis of user's input. Let the set of documents relevant to a query be denoted as {Relevant} and the set of retrieved document as {Retrieved}. The set of documents that are relevant and retrieved can be denoted as {Relevant} ∩ {Retrieved}. This can be shown in the form of a Venn diagram as follows −

There are three fundamental measures for assessing the quality of text retrieval −

- Precision
- Recall
- F-score

Precision

Precision is the percentage of retrieved documents that are in fact relevant to the query. Precision can be defined as −

Precision= |{Relevant} ∩ {Retrieved}| / |{Retrieved}|

Recall

Recall is the percentage of documents that are relevant to the query and were in fact retrieved. Recall is defined as −

Recall = |{Relevant} ∩ {Retrieved}| / |{Relevant}|

F-score

F-score is the commonly used trade-off. The information retrieval system often needs to trade-off for precision or vice versa. F-score is defined as harmonic mean of recall or precision as follows −

F-score = recall x precision / (recall + precision) / 2

Information Retrieval Techniques

- Index Terms (Attribute) Selection:
  - Stop list
  - Word stem
  - Index terms weighting methods
- Terms × Documents Frequency Matrices
- Information Retrieval Models:
  - Boolean Model
  - Vector Model
  - Probabilistic Model

Boolean Model

- Consider that index terms are either present or absent in a document
- As a result, the index term weights are assumed to be all binaries
- A query is composed of index terms linked by three connectives: not, and, and or
  - e.g.: car *and* repair, plane *or* airplane
- The Boolean model predicts that each document is either relevant or non-relevant based on the match of a document to the query

Keyword-Based Retrieval

- A document is represented by a string, which can be identified by a set of keywords
- Queries may use expressions of keywords
  - E.g., car *and* repair shop, tea *or* coffee, DBMS *but not* Oracle
  - Queries and retrieval should consider synonyms, e.g., repair and maintenance
- Major difficulties of the model
  - Synonymy: A keyword *T* does not appear anywhere in the document, even though the document is closely related to *T*, e.g., data mining
  - Polysemy: The same keyword may mean different things in different contexts, e.g., mining

Similarity-Based Retrieval in Text Data

- Finds similar documents based on a set of common keywords
- Answer should be based on the degree of relevance based on the nearness of the keywords, relative frequency of the keywords, etc.
- Basic techniques
- Stop list
  - Set of words that are deemed "irrelevant", even though they may appear frequently
  - E.g., *a, the, of, for, to, with*, etc.
  - Stop lists may vary when document set varies
- Word stem
  - Several words are small syntactic variants of each other since they share a common word stem
  - E.g., *drug, drugs, drugged*
- A term frequency table
  - Each entry *frequent_table(i, j)* = # of occurrences of the word t_i in document d_i
  - Usually, the *ratio* instead of the absolute number of occurrences is used
- Similarity metrics: measure the closeness of a document to a query (a set of keywords)
  - Relative term occurrences
  - Cosine distance:

$$sim(v_1, v_2) = \frac{v_1 \cdot v_2}{| v_1 \| v_2 |}$$

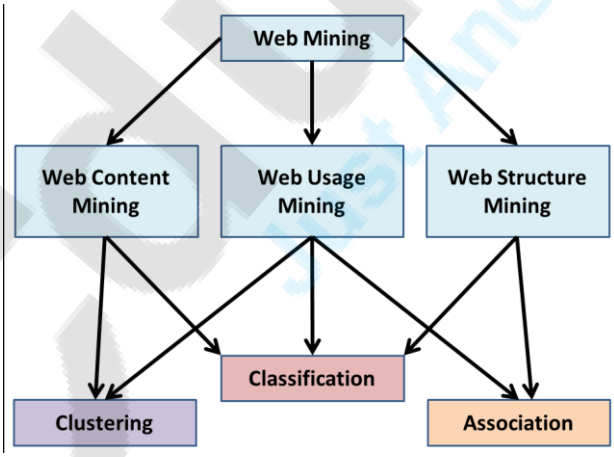
Web mining

Web mining is the application of data mining techniques to discover patterns from the World Wide Web. As the name proposes, this is information gathered by mining the web. It makes utilization of automated apparatuses to reveal and extricate data from servers and web2 reports, and it permits organizations to get to both organized and unstructured information from browser activities, server logs, website and link structure, page content and different sources.

The goal of Web structure mining is to generate structural summary about the Web site and Web page. Technically, Web content mining mainly focuses on the structure of inner-document, while Web structure mining tries to discover the link structure of the hyperlinks at the inter-document level. Based on the topology of the hyperlinks, Web structure mining will categorize the Web pages and generate the information, such as the similarity and relationship between different Web sites.

Web structure mining can also have another direction -- discovering the structure of Web document itself. This type of structure mining can be used to reveal the structure (schema) of Web pages, this would be good for navigation purpose and make it possible to compare/integrate Web page schemes. This type of structure mining will facilitate introducing database techniques for accessing information in Web pages by providing a reference schema.

Web mining can be divided into three different types – **Web usage mining**, **Web content mining** and **Web structure mining**.



Web usage mining

Web Usage Mining is the application of data mining techniques to discover interesting usage patterns from Web data in order to understand and better serve the needs of Web-based applications. Usage data captures the identity or origin of Web users along with their browsing behavior at a Web site.

Web usage mining itself can be classified further depending on the kind of usage data considered:

- **Web Server Data**: The user logs are collected by the Web server. Typical data includes IP address, page reference and access time.
- **Application Server Data**: Commercial application servers have significant features to enable e-commerce applications to be built on top of them with little effort. A key feature is the ability to track various kinds of business events and log them in application server logs.
- **Application Level Data**: New kinds of events can be defined in an application, and logging can be turned on for them thus generating histories of these specially defined events. It must be noted, however, that many end applications require a combination of one or more of the techniques applied in the categories above.

Studies related to work are concerned with two areas: constraint-based data mining algorithms applied in Web Usage Mining and developed software tools (systems). Costa and Seco demonstrated that web log mining can be used to extract semantic information (hyponymy relationships in particular) about the user and a given community

Web structure mining

Web structure mining uses graph theory to analyze the node and connection structure of a web site. According to the type of web structural data, web structure mining can be divided into two kinds:

1. Extracting patterns from hyperlinks in the web: a hyperlink is a structural component that connects the web page to a different location.
2. Mining the document structure: analysis of the tree-like structure of page structures to describe HTML or XML tag usage.

Web structure mining terminology:

- Web graph: directed graph representing web.
- Node: web page in graph.
- Edge: hyperlinks.
- In degree: number of links pointing to particular node.
- Out degree: number of links generated from particular node.

Techniques of web structure mining:

- PageRank: this algorithm is used by Google to rank search results. The name of this algorithm is given by Google-founder Larry Page. The rank of a page is decided by the number of links pointing to the target node.

Web content mining

Web content mining is the mining, extraction and integration of useful data, information and knowledge from Web page content. The heterogeneity and the lack of structure that permits much of the ever-expanding information sources on the World Wide Web, such as hypertext documents, makes automated discovery, organization, and search and indexing tools of the Internet and the World Wide Web such as Lycos, Alta Vista, WebCrawler, Aliweb, MetaCrawler, and others provide some comfort to users, but they do not generally provide structural information nor categorize, filter, or interpret documents. These factors have prompted researchers to develop more intelligent tools for information retrieval, such as intelligent web agents, as well as to extend database and data mining techniques to provide a higher level of organization for semi-structured data available on the web. The agent-based approach to web mining involves the development of sophisticated AI systems that can act autonomously or semi-autonomously on behalf of a particular user, to discover and organize web-based information.

Web content mining is differentiated from two different points of view: Information Retrieval View and Database View.Summarized the research works done for unstructured data and semi-structured data from information retrieval view. It shows that most of the researches use bag of words, which is based on the statistics about single words in isolation, to represent unstructured text and take single word found in the training corpus as features. For the semi-structured data, all the works utilize the HTML structures inside the documents and some utilized the hyperlink structure between the documents for document representation. As for the database view, in order to have the better information management and querying on the web, the mining always tries to infer the structure of the web site to transform a web site to become a database.

There are several ways to represent documents; vector space model is typically used. The documents constitute the whole vector space. This representation does not realize the importance of words in a document. To resolve this, **tf-idf** (Term Frequency Times Inverse Document Frequency) is introduced.

By multi-scanning the document, we can implement feature selection. Under the condition that the category result is rarely affected, the extraction of feature subset is needed. The general algorithm is to construct an evaluating function to evaluate the features. As feature set, information gain, cross entropy, mutual information, and odds ratio are usually used. The classifier and pattern analysis methods of text data mining are very similar to traditional data mining techniques. The usual evaluative merits are classification accuracy, precision and recall and information score.

Web mining is an important component of content pipeline for web portals. It is used in data confirmation and validity verification, data integrity and building taxonomies, content management, content generation and opinion mining.