Q.2 a. With the help of suitable figure explain the architecture of a typical data mining system.

Answer: Page no 7-8 of text book

b. List and describe the five primitives for specifying a data mining task

Answer: Page no 31 of text book

Q.3 a. Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.
(i) What is the mean of the data? What is the median?
(ii) What is the mode of the data? Comment on the data's modality (i.e., bimodal, trimodal etc.)
(iii) What is the midrange of the data?
(iv) Can you find (roughly) the first quartile (Q1) and the third quartile (Q3) of the data?
(v) Give the five-number summary of the data.

Answer: Page no 98 of text book

b. Why is data cleansing and data transformation functions considered to be a vital task in the integration process. Explain these functions and also mention where these functions take place.

Answer:
Data cleansing, data cleaning or data scrubbing is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database. Used mainly in databases, the term refers to identifying incomplete, incorrect, inaccurate, irrelevant, etc. parts of the data and then replacing, modifying, or deleting this dirty data.

After cleansing, a data set will be consistent with other similar data sets in the system. The inconsistencies detected or removed may have been originally caused by user entry errors, by corruption in transmission or storage, or by different data dictionary definitions of similar entities in different stores.

Data cleansing differs from data validation in that validation almost invariably means data is rejected from the system at entry and is performed at entry time, rather than on batches of data.

The actual process of data cleansing may involve removing typographical errors or validating and correcting values against a known list of entities. The validation may be strict (such as rejecting any address that does not have a valid postal code) or fuzzy (such as correcting records that partially match existing, known records).

Some data cleansing solutions will clean data by cross checking with a validated data set. Also data enhancement, where data is made more complete by adding related information, is a common data cleansing practice. For example, appending addresses with phone numbers related to that address.

Data cleansing may also involve activities like, harmonization of data, and standardization of data. For example, harmonization of short codes (St, rd etc.) to actual words (street, road). Standardization of data is a means changing of reference data set to a new standard, ex, use of standard codes.
High-quality data needs to pass a set of quality criteria. Those include:

- **Validity**: The degree to which the measures conform to defined business rules or constraints (see also Validity (statistics). When modern database technology is used to design data-capture systems, validity is fairly easy to ensure: invalid data arises mainly in legacy contexts (where constraints were not implemented in software) or where inappropriate data-capture technology was used (e.g., spreadsheets, where it is very hard to limit what a user chooses to enter into a cell).Data constraints fall into the following categories:
    - *Data-Type Constraints* – e.g., values in a particular column must be of a particular datatype, e.g., Boolean, numeric (integer or real), date, etc.
    - *Range Constraints:* typically, numbers or dates should fall within a certain range. That is, they have minimum and/or maximum permissible values.
    - *Mandatory Constraints:* Certain columns cannot be empty.
    - *Unique Constraints:* A field, or a combination of fields, must be unique across a dataset. For example, no two persons can have the same social security number.
    - *Set-Membership constraints*: The values for a column come from a set of discrete values or codes. For example, a person's gender may be Female, Male or Unknown (not recorded).
    - *Foreign-key constraints*: This is the more general case of set membership. The set of values in a column is defined in a column of another table that contains unique values. For example, in a US taxpayer database, the "state" column is required to belong to one of the US's defined states or territories: the set of permissible states/territories is recorded in a separate States table. The term foreign key is borrowed from relational database terminology: follow the hyperlink for more details.
    - Regular expression patterns: Occasionally, text fields will have to be validated this way. For example, phone numbers may be required to have the pattern (999) 999-9999.
    - Cross-field validation: Certain conditions that utilize multiple fields must hold. For example, in laboratory medicine, the sum of the components of the differential white blood cell count must be equal to 100 (since they are all percentages). In a hospital database, a patient's date of discharge from hospital cannot be earlier than the date of admission.

- **Accuracy**: The degree of conformity of a measure to a standard or a true value - see also Accuracy and precision. Accuracy is very hard to achieve through data-cleansing in the general case, because it requires accessing an external source of data that contains the true value: such "gold standard" data is often unavailable. Accuracy has been achieved in some cleansing contexts, notably customer contact data, by using external databases that match up zip codes to geographical locations (city and state), and also help verify that street addresses within these zip codes actually exist.
- **Completeness**: The degree to which all required measures are known (see also Completeness). Incompleteness is almost impossible to fix with data cleansing methodology: one cannot infer facts that were not captured when the data in question was initially recorded. (In some contexts, e.g., interview data, it may be possible to fix incompleteness by going back to the original source of data, i,e., re-interviewing the subject, but even this does not guarantee success because of problems of recall - e.g., in an interview to gather data on food consumption, no one is likely to remember exactly what one ate six months ago. In the case of systems that insist certain columns should not be empty, one may work around the problem by designating a value that indicates "unknown" or "missing", but supplying of default values does not imply that the data has been made complete.
- **Consistency**: The degree to which a set of measures are equivalent in across systems (see also Consistency). Inconsistency occurs when two data items in the data set contradict each other: e.g., a customer is recorded in two different systems as having two different current addresses, and only one of them can be correct. Fixing inconsistency is not always possible: it requires a variety of strategies - e.g., deciding which data were recorded more recently, which data source is likely to be most reliable (the latter knowledge may be specific to a given organization), or simply trying to find the truth by testing both data items (e.g., calling up the customer).
- **Uniformity**: The degree to which a set data measures are specified using the same units of measure in all systems (see also Unit of measure). In datasets pooled from different locales, weight may be recorded either in pounds or kilos, and must be converted to a single measure using an arithmetic transformation.

The term **Integrity** encompasses accuracy, consistency and some aspects of validation (see also Data integrity) but is rarely used by itself in data-cleansing contexts because it is insufficiently specific. (For example, "referential integrity" is a term used to refer to the enforcement of foreign-key constraints above.)
The process of data cleansing

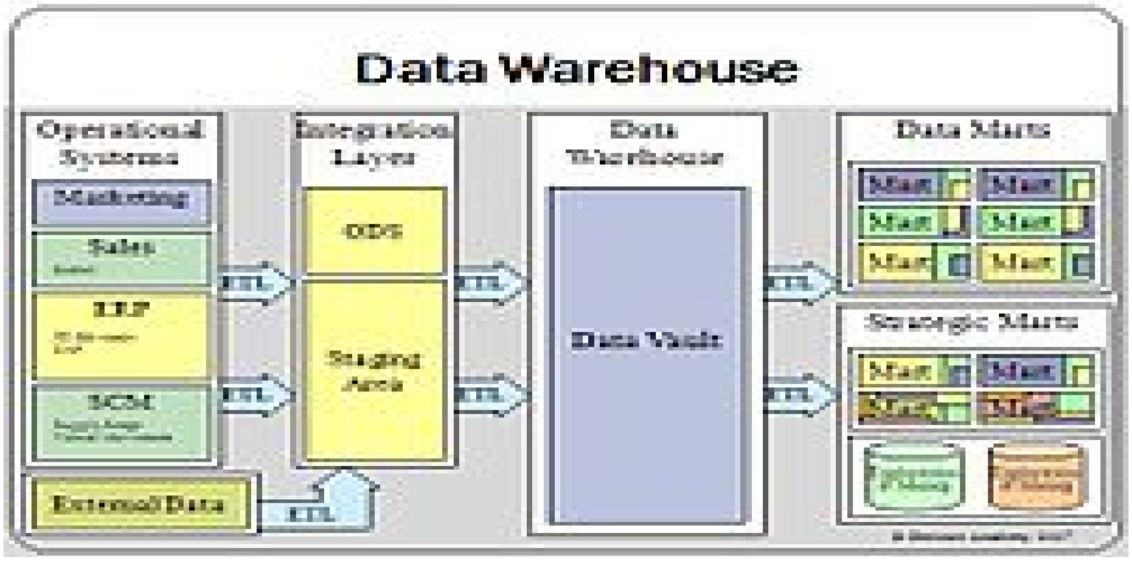
- Data auditing: The data is audited with the use of statistical and database methods to detect anomalies and contradictions: this eventually gives an indication of the characteristics of the anomalies and their locations. Several commercial software packages will let you specify constraints of various kinds (using a grammar that conforms to that of a standard programming language, e.g., JavaScript of Visual Basic) and then generate code that checks the data for violation of these constraints. This process is referred to below in the bullets "workflow specification" and "workflow execution." For users who lack access to high-end cleansing software, Microcomputer database packages such as Microsoft Access or FileMaker Pro will also let you perform such checks, on a constraint-by-constraint basis, interactively with little or no programming required in many cases.

- **Workflow specification**: The detection and removal of anomalies is performed by a sequence of operations on the data known as the workflow. It is specified after the process of auditing the data and is crucial in achieving the end product of high-quality data. In order to achieve a proper workflow, the causes of the anomalies and errors in the data have to be closely considered.
- **Workflow execution**: In this stage, the workflow is executed after its specification is complete and its correctness is verified. The implementation of the workflow should be efficient, even on large sets of data, which inevitably poses a trade-off because the execution of a data-cleansing operation can be computationally expensive.
- **Post-processing and controlling**: After executing the cleansing workflow, the results are inspected to verify correctness. Data that could not be corrected during execution of the workflow is manually corrected, if possible. The result is a new cycle in the data-cleansing process where the data is audited again to allow the specification of an additional workflow to further cleanse the data by automatic processing.

Q.4 a. Explain the architecture of a data warehouse. Also explain the single-tier and three tier architectures of a data warehouse.

Answer:
A simple architecture is the single-layer architecture. There is no physical data warehouse or data mart between the operation data and the analytic tools. The middleware in this type of system should be considered a virtual data warehouse, which consists of a software layer and not a data based layer. The single-layer model is light weight as it minimises redundancies and thereby the amount of data stored. It has, however, no separation between analytical and operational processing. The analysis are based directly on the operational data.

The three-layer architecture consists of the source layer (containing multiple source systems), the reconciled layer and the data warehouse layer (containing both data warehouses and data marts). The reconciled layer sits between the source data and data warehouse. It is populated with data from the source systems through an ETL process and the data stored in it is published further through another ETL process. In the reconciled layer the data has been cleaned up once and integrated to a common standardised form from multiple different source systems. The ETL process that feeds the data warehouse then only gets already integrated data that has less need for transformation. This architecture is especially useful for the very large, enterprise-wide systems. A disadvantage of this architecture is the extra data storage space used through the extra redundant reconciled layer. It also makes the analytical tools a little further away from being real-time.



b. **You are the senior analyst responsible for selecting the tools in your data warehouse. Make a list of the tools you will provide for use by the developers and the end users of your data warehouse. Describe the features of these tools.**

Answer:

DATA MINING TOOLS

Organizations that wish to use data mining tools can purchase mining programs designed for existing software and hardware platforms, which can be integrated into new products and systems as they are brought online, or they can build their own custom mining solution. For instance, feeding the output of a data mining exercise into another computer system, such as a neural network, is quite common and can give the mined data more value. This is because the data mining tool gathers the data, while the second program (e.g., the neural network) makes decisions based on the data collected.

Different types of data mining tools are available in the marketplace, each with their own strengths and weaknesses. Internal auditors need to be aware of the different kinds of data mining tools available and recommend the purchase of a tool that matches the organization's current detective needs. This should be considered as early as possible in the project's lifecycle, perhaps even in the feasibility study.

Most data mining tools can be classified into one of three categories: traditional data mining tools, dashboards, and text-mining tools. Below is a description of each.

- ***Traditional Data Mining Tools.*** Traditional data mining programs help companies establish data patterns and trends by using a number of complex algorithms and techniques. Some of these tools are installed on the desktop to monitor the data and highlight trends and others capture information residing outside a database. The majority are available in both Windows and UNIX versions, although some specialize in one operating system only. In addition, while some may concentrate on one database type, most will be able to handle any data using online analytical processing or a similar technology.

- ***Dashboards.*** Installed in computers to monitor information in a database, dashboards reflect data changes and updates onscreen — often in the form of a chart or table — enabling the user to see how the business is performing. Historical data also can be referenced, enabling the user to see where things have changed (e.g., increase in sales from the same period last year). This functionality makes dashboards easy to use and particularly appealing to managers who wish to have an overview of the company's performance.
- ***Text-mining Tools.*** The third type of data mining tool sometimes is called a text-mining tool because of its ability to mine data from different kinds of text — from Microsoft Word and Acrobat PDF documents to simple text files, for example. These tools scan content and convert the selected data into a format that is compatible with the tool's database, thus providing users with an easy and convenient way of accessing data without the need to open different applications. Scanned content can be unstructured (i.e., information is scattered almost randomly across the document, including e-mails, Internet pages, audio and video data) or structured (i.e., the data's form and purpose is known, such as content found in a database). Capturing these inputs can provide organizations with a wealth of information that can be mined to discover trends, concepts, and attitudes.

In addition, internal auditors can use spreadsheets to undertake simple data mining exercises or to produce summary tables. Some of the desktop, notebook, and server computers that run operating systems such as Windows, Linux, and Macintosh can be imported directly into Microsoft Excel. Using pivotal tables in the spreadsheet, auditors can review complex data in a simplified format and drill down where necessary to find the underlining assumptions or information.

When evaluating data mining strategies, companies may decide to acquire several tools for specific purposes, rather than purchasing one tool that meets all needs. Although acquiring several tools is not a mainstream approach, a company may choose to do so if, for example, it installs a dashboard to keep managers informed on business matters, a full data-mining suite to capture and build data for its marketing and sales arms, and an interrogation tool so auditors can identify fraud activity.

DATA MINING TECHNIQUES AND THEIR APPLICATION

In addition to using a particular data mining tool, internal auditors can choose from a variety of data mining techniques. The most commonly used techniques include artificial neural networks, decision trees, and the nearest-neighbor method. Each of these techniques analyzes data in different ways:

- ***Artificial neural networks*** are non-linear, predictive models that learn through training. Although they are powerful predictive modeling techniques, some of the power comes at the expense of ease of use and deployment. One area where auditors can easily use them is when reviewing records to identify fraud and fraud-like actions. Because of their complexity, they are better employed in situations where they can be used and reused, such as reviewing credit card transactions every month to check for anomalies.

- ***Decision trees*** are tree-shaped structures that represent decision sets. These decisions generate rules, which then are used to classify data. Decision trees are the favored technique for building understandable models. Auditors can use them to assess, for example, whether the organization is using an appropriate cost-effective marketing strategy that is based on the assigned value of the customer, such as profit.
- ***The nearest-neighbor method*** classifies dataset records based on similar data in a historical dataset. Auditors can use this approach to define a document that is interesting to them and ask the system to search for similar items.

Each of these approaches brings different advantages and disadvantages that need to be considered prior to their use. Neural networks, which are difficult to implement, require all input and resultant output to be expressed numerically, thus needing some sort of interpretation depending on the nature of the data-mining exercise. The decision tree technique is the most commonly used methodology, because it is simple and straightforward to implement. Finally, the nearest-neighbor method relies more on linking similar items and, therefore, works better for extrapolation rather than predictive enquiries.

A good way to apply advanced data mining techniques is to have a flexible and interactive data mining tool that is fully integrated with a database or data warehouse. Using a tool that operates outside of the database or data warehouse is not as efficient. Using such a tool will involve extra steps to extract, import, and analyze the data. When a data mining tool is integrated with the data warehouse, it simplifies the application and implementation of mining results. Furthermore, as the warehouse grows with new decisions and results, the organization can mine best practices continually and apply them to future decisions.

Regardless of the technique used, the real value behind data mining is modeling — the process of building a model based on user-specified criteria from already captured data. Once a model is built, it can be used in similar situations where an answer is not known. For example, an organization looking to acquire new customers can create a model of its ideal customer that is based on existing data captured from people who previously purchased the product. The model then is used to query data on prospective customers to see if they match the profile. Modeling also can be used in audit departments to predict the number of auditors required to undertake an audit plan based on previous attempts and similar work.

c) Depending on the requirements of the organization, this process varies widely. Some data warehouses may overwrite existing information with cumulative information, frequently updating extract data is done on daily, weekly or monthly basis. Other DW (or even other parts of the same DW) may add new data in a historical form, for example, hourly. To understand this, consider a DW that is required to maintain sales records of the last year. Then, the DW will overwrite any data that is older than a year with newer data. However, the entry of data for any one year window will be made in a historical manner. The timing and scope to replace or append are strategic design choices

dependent on the time available and the business needs. More complex systems can maintain a history and audit trail of all changes to the data loaded in the DW.
As the load phase interacts with a database, the constraints defined in the database schema — as well as in triggers activated upon data load — apply (for example, uniqueness, referential integrity, mandatory fields), which also contribute to the overall data quality performance of the ETL process.

- For example, a financial institution might have information on a customer in several departments and each department might have that customer's information listed in a different way. The membership department might list the customer by name, whereas the accounting department might list the customer by number. ETL can bundle all this data and consolidate it into a uniform presentation, such as for storing in a database or data warehouse.

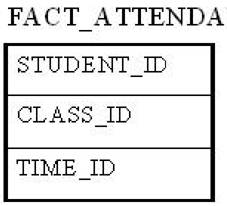
- Another way that companies use ETL is to move information to another application permanently. For instance, the new application might use another database vendor and most likely a very different database schema. ETL can be used to transform the data into a format suitable for the new application to use.

- An example of this would be an Expense and Cost Recovery System (ECRS) such as used by accountancies, consultancies and lawyers. The data usually ends up in the time and billing system, although some businesses may also utilize the raw data for employee productivity reports to Human Resources (personnel dept.) or equipment usage reports to Facilities Management

Q.5 a. Write a short note on fact less fact table. Draw a star schema representing a factless fact table of a patient visiting a hospital.

Answer:

A fact less fact table is a fact table that does not have any measures. It is essentially an intersection of dimensions. On the surface, a fact less fact table does not make sense, since a fact table is, after all, about facts. However, there are situations where having this kind of relationship makes sense in data warehousing.
For example, think about a record of student attendance in classes. In this case, the fact table would consist of 3 dimensions: the student dimension, the time dimension, and the class dimension. This fact less fact table would look like the following:



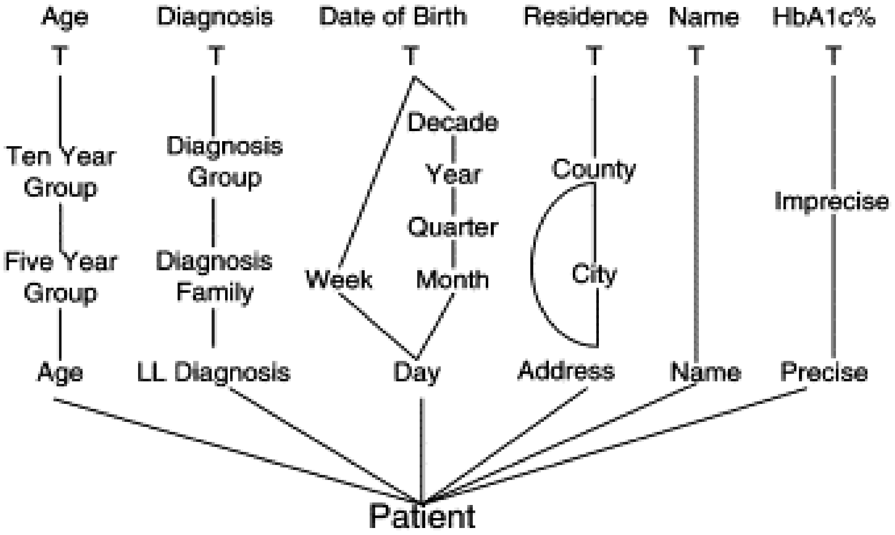
FACT_ATTENDANCE

STUDENT_ID
CLASS_ID
TIME_ID

The only measure that you can possibly attach to each combination is "1" to show the presence of that particular combination. However, adding a fact that always shows 1 is redundant because we can simply use the COUNT function in SQL to answer the same questions.
Fact less fact tables offer the most flexibility in data warehouse design. For example, one can easily answer the following questions with this fact less fact table:

- How many students attended a particular class on a particular day?
- How many classes on average does a student attend on a given day?

Without using a factless fact table, we will need two separate fact tables to answer the above two questions. With the above factless fact table, it becomes the only fact table that's needed.



b. Give a description on the methods used in data cube implementation. Explain with examples.

Answer:
Users of decision support systems often see data in the form of ***data cubes***. The cube is used to represent data along some measure of interest. Although called a "cube", it can be 2-dimensional, 3-dimensional, or higher-dimensional. Each dimension represents some attribute in the database and the cells in the data cube represent the measure of interest. For example, they could contain a count for the number of times that attribute combination occurs in the database, or the minimum, maximum, sum or average value of some attribute. Queries are performed on the cube to retrieve decision support information.

Example: We have a database that contains transaction information relating company sales of a part to a customer at a store location. The data cube formed from this database is a 3-dimensional representation, with each cell (*p*,*c*,*s*) of the cube representing a combination of values from *part*, *customer* and *store-location*. A sample data cube for this combination is shown in Figure 1. The contents of each cell is the count of the number of times that specific combination of values occurs together in the database. Cells that appear blank in fact have a value of zero. The cube can then be used to retrieve information within the database about, for example, which store should be given a certain part to sell in order to make the greatest sales.

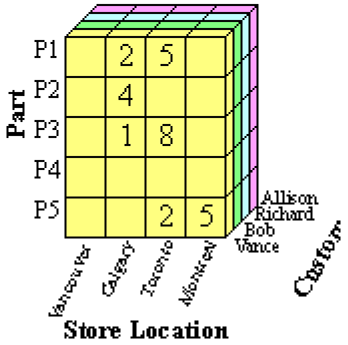


Figure 1(a): Front View of Sample Data Cube

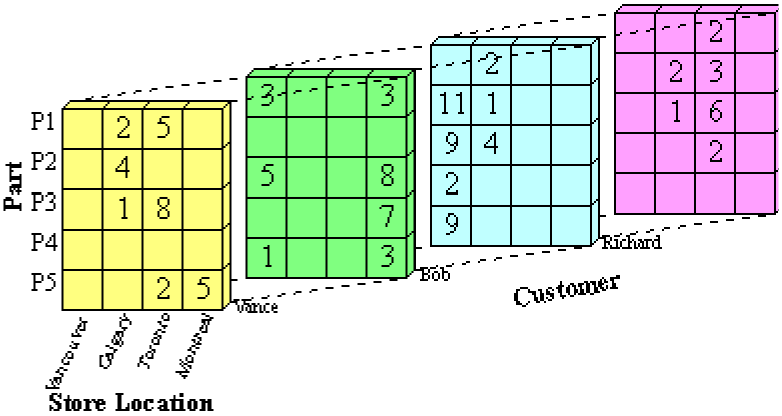


Figure 1(b): Entire View of Sample Data Cube

Computed versus Stored Data Cubes

The goal is to retrieve the decision support information from the data cube in the most efficient way possible. Three possible solutions are:

1. Pre-compute all cells in the cube
2. Pre-compute no cells
3. Pre-compute some of the cells

If the whole cube is pre-computed, then queries run on the cube will be very fast. The disadvantage is that the pre-computed cube requires a lot of memory. The size of a cube for *n* attributes $A_1,...,A_n$ with cardinalities $|A_1|,...,|A_n|$ is $\pi|A_i|$. This size increases exponentially with the number of attributes and linearly with the cardinalities of those attributes.

To minimize memory requirements, we can pre-compute none of the cells in the cube. The disadvantage here is that queries on the cube will run more slowly because the cube will need to be rebuilt for each query.

As a compromise between these two, we can pre-compute only those cells in the cube which will most likely be used for decision support queries. The trade-off between memory space and computing time is called the ***space-time trade-off***, and it often exists in data mining and computer science in general.

Representation

***m*-DimensionalArray:**

A data cube built from *m* attributes can be stored as an *m*-dimensional array. Each element of the array contains the measure value, such as count. The array itself can be represented as a 1-dimensional array. For example, a 2-dimensional array of size *x* x *y* can be stored as a 1-dimensional array of size *x***y*, where element (*i*,*j*) in the 2-D array is stored in location (*y***i*+*j*) in the 1-D array. The disadvantage of storing the cube directly as an array is that most data cubes are sparse, so the array will contain many empty elements (zero values).

List of Ordered Sets:

To save storage space we can store the cube as a sparse array or a list of ordered sets. If we store all cells in the data cube from Figure 1, then the resulting datacube will contain ($card_{Part}$ $*card_{StoreLocation}*card_{Customer}$) combinations, which is 5 * 4 * 4 = 80 combinations. If we eliminate cells in the cube that contain zero, such as {P1, Vancouver, Allison}, only 27 combinations remain, as seen in Table 1.

Table 1 shows an ***ordered set representation*** of the data cube. Each attribute value combination is paired with its corresponding count. This representation can be easily stored in a database table to facilitate queries on the data cube.

Combination	**Count**
{P1, Calgary, Vance}	2
{P2, Calgary, Vance}	4
{P3, Calgary, Vance}	1
{P1, Toronto, Vance}	5
{P3, Toronto, Vance}	8
{P5, Toronto, Vance}	2
{P5, Montreal, Vance}	5
{P1, Vancouver, Bob}	3
{P3, Vancouver, Bob}	5
{P5, Vancouver, Bob}	1
{P1, Montreal, Bob}	3
{P3, Montreal, Bob}	8
{P4, Montreal, Bob}	7
{P5, Montreal, Bob}	3
{P2, Vancouver, Richard}	11
{P3, Vancouver, Richard}	9
{P4, Vancouver, Richard}	2
{P5, Vancouver, Richard}	9
{P1, Calgary, Richard}	2
{P2, Calgary, Richard}	1
{P3, Calgary, Richard}	4
{P2, Calgary, Allison}	2
{P3, Calgary, Allison}	1
{P1, Toronto, Allison}	2
{P2, Toronto, Allison}	3
{P3, Toronto, Allison}	6
{P4, Toronto, Allison}	2

Table 1: Ordered Set Representation of a Data Cube

Representation of Totals

Another aspect of data cube representation which can be considered is the representation of totals. A simple data cube does not contain totals. The storage of totals increases the size of the data cube but can also decrease the time to make total-based queries. A simple way to represent totals is to add an additional layer on *n* sides of the *n*-dimensional datacube. This can be easily visualized with the 3-dimensional data cube introduced in Figure 1. Figure 2 shows the original cube with an additional layer on each of three sides to store total values. The totals represent the sum of all values in one horizontal row, vertical row (column) or depth row of the data cube.

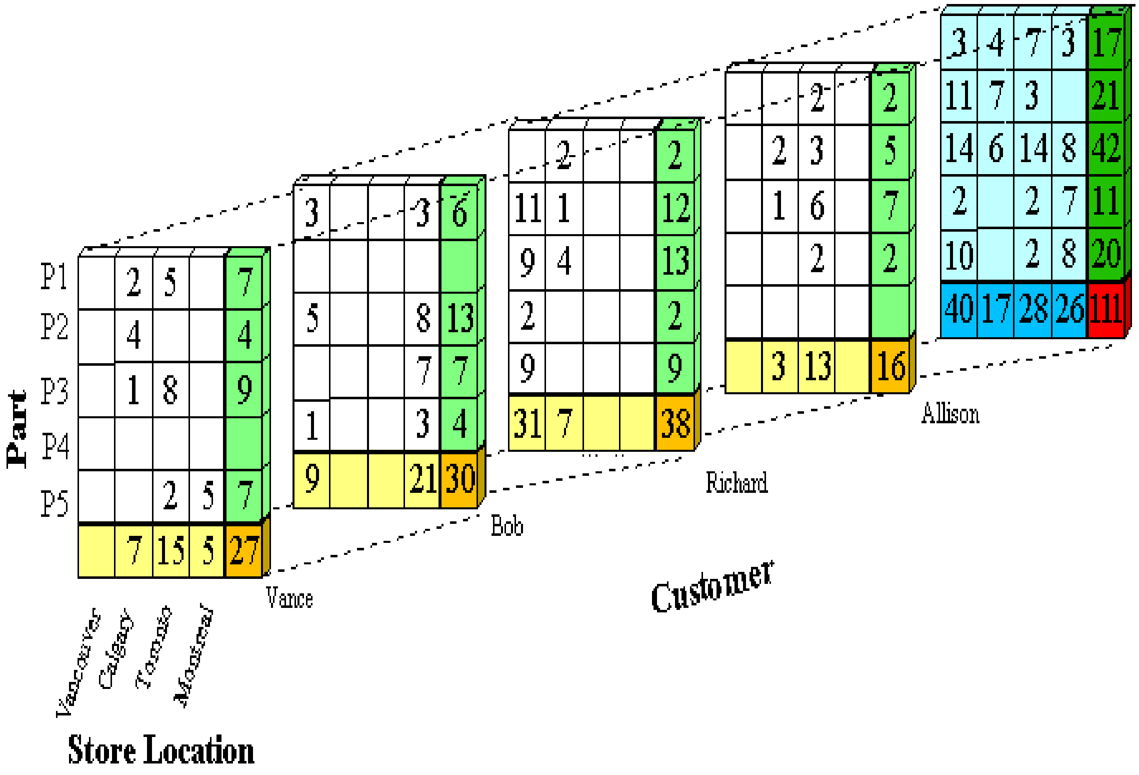


Figure 2: Cube with Totals

The color coding used in Figure 2 is as follows:

- White: Original values
- Light yellow: Total for one customer and one store location
- Light green: Total for one customer and one part
- Light blue: Total for one part and one store location
- Dark yellow: Total for one customer
- Dark green: Total for one part
- Dark blue: Total for one store location
- Red: Total number of transactions in all

To store these totals in ordered set representation the value ***ANY*** can be used. For example, there are 15 transactions where Vance buys a part in Toronto. The ordered set representation of this is ({ANY, Toronto, Vance},15), because it could be any part. The ordered set representation of all of Vance's transactions is ({ANY, ANY, Vance}, 27), that is all transactions at all store locations for Vance. The total number of transactions in the whole cube is found in the red cell and is 111. This is represented as ({ANY, ANY, ANY}, 111).

c. Give reasons why feeding data into the OLAP system directly from the source system is not preferred. Explain.

Answer:
Business Users needed to build queries that summarized the data and fed management reports. Such queries, were extremely slow because they usually summarize large amounts of data, while sharing the database engine with every day operations, **Reason1**
which in turn **adversely affected the performance of operational systems. Reason**
Delay in strategic planning of the enterprise.

The solution was, therefore, to separate the data used for reporting and decision making from the operational systems. Hence, data warehouses were designed and built to house this kind of data so that it can be used later in the strategic planning of the enterprise. In OLTP machine you want to keep the data normalized because you want the transaction or the machine to be fast. In OLAP machine you want to keep the data denormalized because of the complex Querying by the DSS tools.

Q.6 a. Explain the working of decision tree algorithm with the help of an example. Write down the advantages and limitations of the technique.

Answer:

```
TreeGrowing (S,A,y)
Where:
S - Training Set
A - Input Feature Set
y - Target Feature

Create a new tree T with a single root node.
IF One of the Stopping Criteria is fulfilled THEN
Mark the root node in T as a leaf with the most
common value of y in S as a label.

ELSE
Find a discrete function f(A) of the input
attributes values such that splitting S
according to f(A)’s outcomes (v1,...,vn) gains
the best splitting metric.

IF best splitting metric > treshold THEN
Label t with f(A)
FOR each outcome vi of f(A):
Set Subtreei= TreeGrowing (¾f(A)=viS,A,y).
Connect the root node of tT to Subtreei with
```

```
an edge that is labelled as vi
END FOR

ELSE
Mark the root node in T as a leaf with the most
common value of y in S as a label.
END IF
END IF
RETURN T
TreePruning (S,T,y)
Where:
S - Training Set
y - Target Feature
T - The tree to be pruned

DO
Select a node t in T such that pruning it
maximally improve some evaluation criteria
IF t6=Ø THEN T=pruned(T,t)
UNTIL t=Ø
RETURN                                                     T
```

A decision tree is a classifier expressed as a recursive partition of the instance space. The decision tree consists of nodes that form a rooted tree, meaning it is a directed tree with a node called “root” that has no incoming edges. All other nodes have exactly one incoming edge. A node with outgoing edges is called an internal or test node. All other nodes are called leaves (also known as terminal or decision nodes). In a decision tree, each internal node splits the instance space into two or more sub-spaces according to a certain discrete function of the input attributes values. In the simplest and most frequent case, each test considers a single attribute, such that the instance space is partitioned according to the attribute’s value. In the case of numeric attributes, the condition refers to a range. Each leaf is assigned to one class representing the most appropriate target value. Alternatively, the leaf may hold a probability vector indicating the probability of the target attribute having a certain value. Instances are classified by navigating them from the root of the tree down to a leaf, according to the outcome of the tests along the path.

b. Discuss the different criteria based on which frequent pattern mining can be classified

Answer: Page no 232 of text book

Q7 a. Prediction is frequently referred to as the forecasting of missing numerical values. Justify the statement with an example.

Answer:

In case of predication we do not use class label attributes because the attribute for which values are being predicted is continuous-valued (ordered) rather than categorical (discrete-valued and unordered). The attribute are generally called as the predicted attribute. Assume that, in our example, we instead wanted to predict the credit limit amount that would be "safe" for the card request applicant. Then the data mining task becomes prediction, rather than classification. The categorical attribute, Card decision, is replaced by the continuous-valued variable credit limit amount as the predicted attribute, and build a predictor for our task. Prediction can also be viewed as a mapping or function, y= f (X), where X is the input (e.g., a tuple describing a card applicant), and the output y is a continuous or ordered value (such as the predicted amount that the can be set as credit limit of the applicant); That means, we wish to learn a mapping or function that models the relationship between X and y.

Prediction and classification also differ in the methods that are used to build their respective models. As with classification, the training set used to build a predictor should not be used to assess its accuracy. An independent test set should be used instead. The accuracy of a predictor is estimated by computing an error based on the difference between the predicted value and the actual known value of y for each of the test tuples, X. There are various predictor error measures

Preparing Data for Classification and Prediction

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

Data cleaning:

The preprocessing of data so as to remove or reduce and the treatment of missing values

Relevance analysis:

Many of the attributes in the data may be redundant. Correlation analysis can be used to identify whether any two given attributes are statistically related. Relevance analysis, in the form of correlation analysis and attribute subset selection, can be used to detect attributes that do not contribute to the classification or prediction task. Including such attributes may otherwise slow down, and possibly mislead, the learning step.

Data transformation and reduction:

The data may be transformed by normalization. Normalization involves scaling all values for a given attribute so that they fall within a small specified range. The data can also be transformed by generalizing it to higher-level concepts. Concept hierarchies may be used for this purpose. This is particularly useful for continuous valued attributes. Because generalization compresses the original training data, fewer input/output operations may be involved during learning. Data can also be reduced by applying many other methods, ranging from wavelet transformation and principle components analysis to discretization techniques, such as binning, histogram analysis, and clustering.

b. Give the underlying principle of neural networks. State its advantages and shortcomings. What are the applications of this technique in today's scenario?

Answer:
An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurones. This is true of ANNs as well.

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse. This expert can then be used to provide projections given new situations of interest and answer "what if" questions.
Other advantages include:

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
2. Self-Organisation: An ANN can create its own organisation or representation of the information it receives during learning time.
3. Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage

Assume that we want a network to recognise hand-written digits. We might use an array of, say, 256 sensors, each recording the presence or absence of ink in a small area of a single digit. The network would therefore need 256 input units (one for each sensor), 10 output units (one for each kind of digit) and a number of hidden units.
For each kind of digit recorded by the sensors, the network should produce high activity in the appropriate output unit and low activity in the other output units.
To train the network, we present an image of a digit and compare the actual activity of the 10 output units with the desired activity. We then calculate the error, which is defined as the square of the difference between the actual and the desired activities. Next we change the weight of each connection so as to reduce the error.We repeat this training process for many different images of each different images of each kind of digit until the network classifies every image correctly.
To implement this procedure we need to calculate the error derivative for the weight (EW) in order to change the weight by an amount that is proportional to the rate at which the error changes as the weight is changed. One way to calculate the EW is to perturb a

weight slightly and observe how the error changes. But that method is inefficient because it requires a separate perturbation for each of the many weights.
Another way to calculate the EW is to use the Back-propagation algorithm which is described below, and has become nowadays one of the most important tools for training neural networks. It was developed independently by two teams, one (Fogelman-Soulie, Gallinari and Le Cun) in France, the other (Rumelhart, Hinton and Williams) in U.S.

Neural networks are being used:

in investment analysis:

to attempt to predict the movement of stocks currencies etc., from previous data. There, they are replacing earlier simpler linear models.

in signature analysis:

as a mechanism for comparing signatures made (e.g. in a bank) with those stored. This is one of the first large-scale applications of neural networks in the USA, and is also one of the first to use a neural network chip.

in process control:

there are clearly applications to be made here: most processes cannot be determined as computable algorithms. Newcastle University Chemical Engineering Department is working with industrial partners (such as Zeneca and BP) in this area.

in monitoring:

networks have been used to monitor

- the state of aircraft engines. By monitoring vibration levels and sound, early warning of engine problems can be given.
- British Rail have also been testing a similar application monitoring diesel engines.

in marketing:

networks have been used to improve marketing mailshots. One technique is to run a test mailshot, and look at the pattern of returns from this. The idea is to find a predictive mapping from the data known about the clients to how they have responded. This mapping is then used to direct further mailshots.

c. With the help of example, explain post-pruning.

Answer:
verfitting is a significant practical difficulty for decision tree models and many other predictive models. Over fitting happens when the learning algorithm continues to develop hypotheses that reduce training set error at the cost of an
increased test set error. There are several approaches to avoiding over fitting in building decision trees.

- **Pre-pruning** that stop growing the tree earlier, before it perfectly classifies the training set.
- **Post-pruning** that allows the tree to perfectly classify the training set, and then post prune the tree.

Practically, the second approach of post-pruning overfit trees is more successful because it is not easy to precisely estimate when to stop growing the tree. The important step of tree pruning is to define a criterion be used to determine the correct final tree size using one of the following methods:

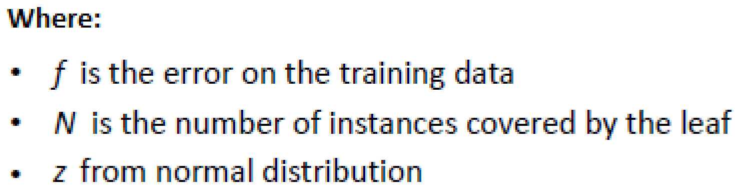
1. Use a distinct dataset from the training set (called validation set), to evaluate the effect of post-pruning nodes from the tree.
2. Build the tree by using the training set, then apply a statistical test to estimate whether pruning or expanding a particular node is likely to produce an improvement beyond the training set.
    - Error estimation
    - Significance testing (e.g., Chi-square test)
3. Minimum Description Length principle: Use an explicit measure of the complexity for encoding the training set and the decision tree, stopping growth of the tree when this encoding size (size (tree) + size(misclassifications(tree)) is minimized.

The first method is the most common approach. In this approach, the available data are separated into two sets of examples: a training set, which is used to build the decision tree, and a validation set, which is used to evaluate the impact of pruning the tree. The second method is also a common approach. Here, we explain the error estimation and Chi^2 test.

Post-pruning using Error estimation

Error estimate for a sub-tree is weighted sum of error estimates for all its leaves. The error estimate (*e*) for a node is:

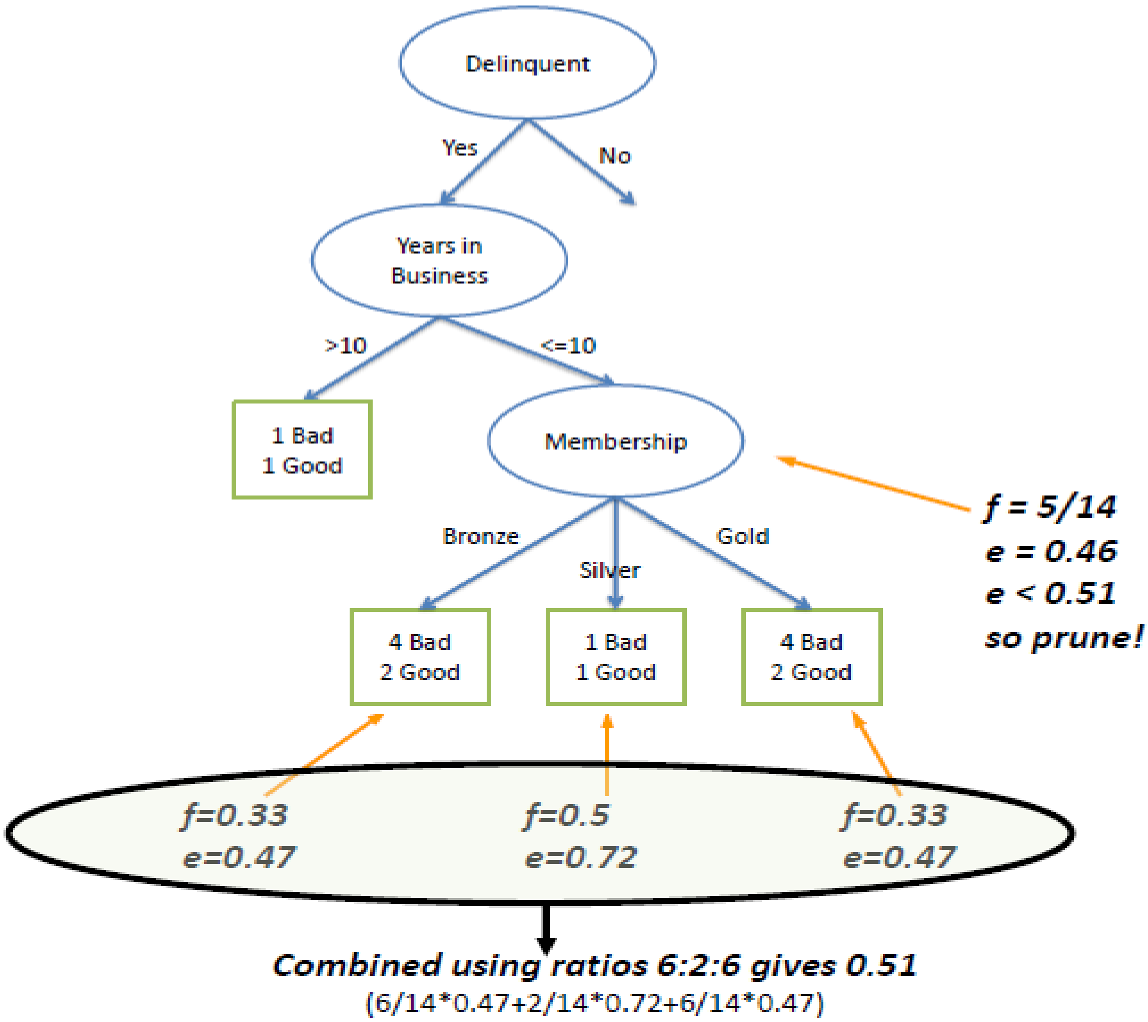
$$e=\left(f+\frac{z^2}{2N}+z\sqrt{\frac{f}{N}-\frac{f^2}{N}+\frac{z^2}{4N^2}}\right)\bigg/\left(1+\frac{z^2}{N}\right)$$



Where:

- f is the error on the training data
- N is the number of instances covered by the leaf
- z from normal distribution

In the following example we set Z to 0.69 which is equal to a confidence level of 75%.



The error rate at the parent node is 0.46 and since the error rate for its children (0.51) increases with the split, we do not want to keep the children. **Post-pruning using Chi^2 test** In Chi^2 test we construct the corresponding frequency table and calculate the Chi^2 value and its probability.

	Bronze	Silver	Gold
Bad	4	1	4
Good	2	1	2

Chi^2 = 0.21 Probability = 0.90 degree of freedom=2

If we require that the probability has to be less than a limit (e.g., 0.05), therefore we decide not to split the node.

Q8. a. What is wave cluster? Why is wavelet transformation useful for clustering?

Answer: Page no 427 of text book

b. Classify and explain hierarchical clustering techniques. Also differentiate between the approaches.

Answer:

How They Work

Given a set of N items to be clustered, and an N*N distance (or similarity) matrix, the basic process of hierarchical clustering is this:

1. Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain.
2. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.
3. Compute distances (similarities) between the new cluster and each of the old clusters.
4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size N. (*)

Step 3 can be done in different ways, which is what distinguishes *single-linkage* from *complete-linkage* and *average-linkage* clustering.

In *single-linkage* clustering (also called the *connectedness* or *minimum* method), we consider the distance between one cluster and another cluster 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, we consider the similarity between one cluster and another cluster to be equal to the greatest similarity from any member of one cluster to any member of the other cluster.

In *complete-linkage* clustering (also called the *diameter* or *maximum* method), we consider the distance between one cluster and another cluster to be equal to the greatest distance from any member of one cluster to any member of the other cluster.

In *average-linkage* clustering, we consider the distance between one cluster and another cluster to be equal to the average distance from any member of one cluster to any member of the other cluster.

A variation on average-link clustering is the UCLUS method of R. D'Andrade (1978) which uses the median distance, which is much more outlier-proof than the average distance.

This kind of hierarchical clustering is called *agglomerative* because it merges clusters iteratively. There is also a *divisive* hierarchical clustering which does the reverse by starting with all objects in one cluster and subdividing them into smaller pieces. Divisive methods are not generally available, and rarely have been applied.

(*) Of course there is no point in having all the N items grouped in a single cluster but, once you have got the complete hierarchical tree, if you want k clusters you just have to cut the k-1 longest links.

Single-Linkage Clustering: The Algorithm

Let’s now take a deeper look at how Johnson’s algorithm works in the case of single-linkage clustering.

The algorithm is an agglomerative scheme that erases rows and columns in the proximity matrix as old clusters are merged into new ones.
The N*N proximity matrix is D = [d(i,j)]. The clusterings are assigned sequence numbers 0,1,......, (n-1) and L(k) is the level of the kth clustering. A cluster with sequence number m is denoted (m) and the proximity between clusters (r) and (s) is denoted d [(r),(s)].
The algorithm is composed of the following steps:

1. *Begin with the disjoint clustering having level L(0) = 0 and sequence number m = 0.*
2. *Find the least dissimilar pair of clusters in the current clustering, say pair (r), (s), according to*

   $d[(r),(s)] = \min d[(i),(j)]$

   where the minimum is over all pairs of clusters in the current clustering.
3. *Increment the sequence number : m = m +1. Merge clusters (r) and (s) into a single cluster to form the next clustering m. Set the level of this clustering to*

   $L(m) = d[(r),(s)]$
4. *Update the proximity matrix, D, by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster, denoted (r,s) and old cluster (k) is defined in this way:*

   $d[(k), (r,s)] = \min d[(k),(r)], d[(k),(s)]$
5. *If all objects are in one cluster, stop. Else, go to step 2.*

Problems

The main weaknesses of agglomerative clustering methods are:

- they do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects;
- they can never undo what was done previously.

Q9 a. How can data mining be used for customer retention, target marketing and CRM? Explain with examples.

Answer:

The various data mining techniques that support CRM applications are

Segmentation

The discovery process of data mining aids in identifying new customers, differentiating the customers and finding best ways to interact with the customers. Data mining techniques like segmentation and classification help in achieving these goals. Organizations can gain insights on customer profiles through segmentation and in turn serve them better by offering them the right type of products and services (Anonymous). Through segmentation, custom-ers could be divided into various groups based on their preferences. These segments of customers could be the bases for target marketing strategies, where one segment is targeted for marketing one type of product.

Target marketing might increase the probability that the customer actually responds to the marketing cam-paign, since the marketing is based on the knowledge of their preferences. Customer profiles are created and market-ing campaigns are personalized for each segment of customers, thereby increasing the probability that they would respond to the campaign .

Or example, the data mining process at Fingerhut Corporation revealed that for customers from a particular zip code, the percentage of expenditure on gold per order was higher than any other product ordered from a particu-lar catalog. This triggered an interest to further analyze the demographics associated with the Zip code. This analysis revealed that the majority of the population residing within the range of that zip code was Hispanic. Following this information Fingerhut revised their catalogs by including more gold jewelry in the catalogs sent out to Hispanic cus-tomers.

AXA Financial uses data mining to gain a better perspective on customer behavior in devising CRM strate-gies. Analysis at AXA financial already includes measuring customer profitability using associations and calculating earnings in tune with the generally accepted accounting principles. Using data mining techniques enables the organ-ization to segment their customers into profitability clusters. This in turn would help AXA develop various models like cost effective service levels, target marketing campaigns, customer scoring, retention, and also devise cross-selling and up-selling strategies.

Association

Another data mining technique that can be used to identify new customers is association. The knowledge about the related interests of the customers could provide good grounds for cross selling products to the different segments of customers. The data about a customer consistently revealing preferences to a certain mix of goods or services, provides information about which associated products the customer is more likely to buy or which types of campaigns he or she is more likely to respond to (Lubel 1998). The classic case of market basket analysis is an ex-ample for identifying associations. Cross selling is one main application that takes cues from association.

In the context of web data mining, the concept of cross selling has gained more importance in the form of personalized ad campaigns based on available data. For example, an online store, which analyzes the shopping baskets of their customers can better personalize their ad campaigns and increase sales (Bugher 2000). On the other hand, an online store can anticipate what its customers would need and suggest other products.

While association allows organization to cross-sell, it also helps in deciding warehouse layouts, which is a very popular application of data mining with the grocery and retail store chains (Lubel 1998). Apart from aiding in store layout, association also helps in identifying valuable relationships among the data as exemplified by a grocery chain in UK. The analysis of their customer database revealed that a particular product was being consumed by the top-spending 25% of its customers. This led to a decision to retain the product on the shelves instead of disconti-nuing it (Sethi).

Predictive Modeling

The predictive modeling capability of data mining helps to attract and retain profitable customers. Using techniques like response models, the customer behavior could be predicted with a fair degree of certainty. Based on the available customer data, response models could be built to predict the response of future customers or the res-ponses of current customers to any new product or service offered (Anonymous 2000). This technique has many ap-plications in the credit, insurance and catalog industries. Even in the case where prior data is not available, it is possible to create profiles of responding customers by studying the population (Cozine 2000). Based on predictions from the response model, a person with a certain mix of characteristics could be classified as either a worthwhile customer or not. If the customer's profile suggests that he or she is a profitable customer, strategies could be devised to attract or to retain him/her. This process of identifying profitable customers would save the organization an enormous amount of expenditures and minimize risks of having bad customers.

Response modeling also aids in more efficient target marketing and saves the finances of an organization (Sethi). Data mining techniques enable the marketing staff to make informed decisions about marketing campaigns without having to wait for the expert statisticians' advice. For example, Liverpool Victoria is a large insurance pro-vider in UK and partnered with Quadstone to provide CRM solutions. Quadstone developed CRM models based on their customer data and made them accessible to the marketers. Now Liverpool Victoria's Marketers have a tool that gave them control of customer profiling, segmentation, and modeling process (Davis 2001). As a result of this effort, Liverpool Victoria is now able to improve retention rates and reduce attrition rates. The database marketers are now empowered to make most of the business decisions on their own without the help of professional statisticians. They are able to verify their business instincts about customer behavior- now more scientifically with the tool available from Quadstone.

C&A is a European fashion retailer, which traditionally outsourced its database marketing operations. When a decision was made to implement CRM strategies in the organization, there was a need for a tool that would integrate the available customer data into the CRM strategy. Once again, with the help of Quadstone's CRM suit, the marketing analysts are now empowered to conduct analyses in-house and C&A has already shown an improvement in the mailing response rates by about 6% (Anonymous 1999). C&A also gained by being able to response rates.

Deviation Detection

Deviation detection is probably one of the most important uses of data mining to CRM. Forensic analysis helps one to identify deviations from the norm. For example, it plays a valuable role in identifying true delinquen-cies and then devising strategies to classify the customer behavior as a delinquent one or as an exception to the be-havior pattern. It also has implications for detecting fraud.

With respect to CRM, forensic analysis might reveal an unusual pattern about a particular customer and af-ter conducting a profitability analysis (Anonymous) the organization could determine if retaining the customer is worthwhile. If the customer turns out to be worthwhile, customized solutions could be offered to that customer and motivate him/her to stay with the organization. On the other hand, if the customer does not turn out to be profitable, then the organization could make an informed decision of not employing any retention strategies to retain the unpro-fitable customer. Fraud detection plays a major role in the health care and credit industries (Sethi), (Anonymous 2000). Deviations detected through data mining could be followed up resulting in huge savings for companies.

Churn detection

In the telecommunication industry churn is a common problem [Berger, 1999 #2], (Lubel 1998), (Anonym-ous 2000). Churn refers to the process where customers switch to the services of a competitor. Usually in these types of industries, customer acquisition is more expensive than customer retention. If an organization is losing many cus-tomers to its competitors, it can analyze the profiles of the lost customers and predict which of the present customers are likely to follow suit. This will enable the organization to devise customer retention strategies and loyalty pro-grams to retain profitable customers. Also, once the churn is identified, it is necessary to perform a profitability analysis to determine if the customer needs better target their high responding customers by making marketing offers tailored to their needs. Hence, C&A used the predictive modeling techniques to improve their direct mailing to be retained. This would save the organization from investing in retain-ing a not so profitable customer.

b.Why is data mining and warehousing applied in biomedical area? Explain.

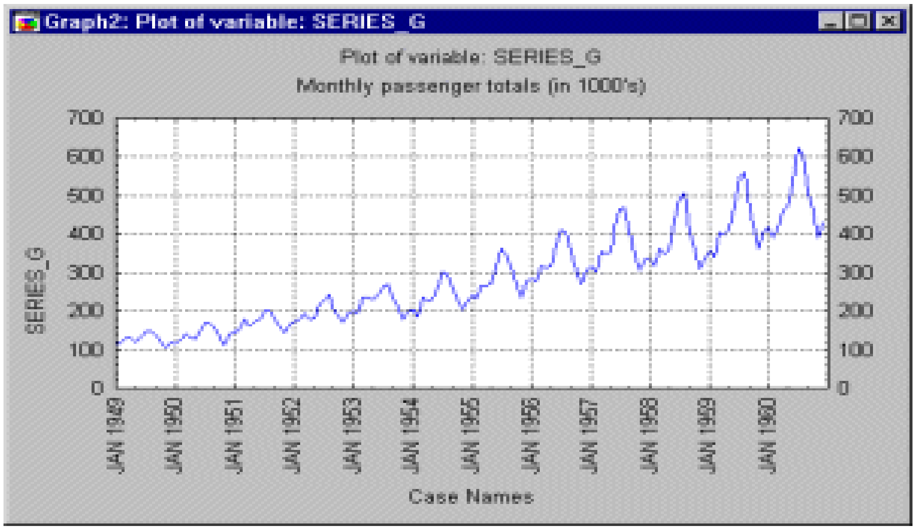
Answer:

Data mining is the process of finding the patterns, associations or relationships among data using different analytical techniques involving the creation of a model and the concluded result will become useful information or knowledge. The advancement of the new medical deceives and the database management systems create a huge number of data-bases in the biomedicine world. Establishing a methodology for knowledge discovery and management of the large amounts of heterogeneous data has become a major priority of research. This paper introduces some basic data mining techniques, unsupervised learning and supervising learning, and reviews the application of data mining in biomedicine. Applications of the multimedia mining, including text, image, video and web mining are discussed. The key issues faced by the computing professional, medical doctors and clinicians are highlighted. We also state some foreseeable future developments in the field. Although extracting useful information from raw biomedical data is a challenging task, data mining is still a good area of scientific study and remains a promising and rich field for research.

c. Give a brief note on time series analysis and its relevance in data mining.

Answer:

Most time series patterns can be described in terms of two basic classes of components: trend and seasonality. The former represents a general systematic linear or (most often) nonlinear component that changes over time and does not repeat or at least does not repeat within the time range captured by our data (e.g., a plateau followed by a period of exponential growth). The latter may have a formally similar nature (e.g., a plateau followed by a period of exponential growth), however, it repeats itself in systematic intervals over time. Those two general classes of time series components may coexist in real-life data. For example, sales of a company can rapidly grow over years but they still follow consistent seasonal patterns (e.g., as much as 25% of yearly sales each year are made in December, whereas only 4% in August).



This general pattern is well illustrated in a "classic" *Series G* data set (Box and Jenkins, 1976, p. 531) representing monthly international airline passenger totals (measured in thousands) in twelve consecutive years from 1949 to 1960 (see example data file *G.sta* and graph above). If you plot the successive observations (months) of airline passenger totals, a clear, almost linear trend emerges, indicating that the airline industry enjoyed a steady growth over the years (approximately 4 times more passengers traveled in 1960 than in 1949). At the same time, the monthly figures will follow an almost identical pattern each year (e.g., more people travel during holidays than during any other time of the year). This example data file also illustrates a very common general type of pattern in time series data, where the amplitude of the seasonal changes increases with the overall trend (i.e., the variance is correlated with the mean over the segments of the series). This pattern which is called *multiplicative seasonality* indicates that the relative amplitude of seasonal changes is constant over time, thus it is related to the trend.

Trend Analysis

There are no proven "automatic" techniques to identify trend components in the time series data; however, as long as the trend is monotonous (consistently increasing or decreasing) that part of data analysis is typically not very difficult. If the time series data contain considerable error, then the first step in the process of trend identification is smoothing.

Smoothing. Smoothing always involves some form of local averaging of data such that the nonsystematic components of individual observations cancel each other out. The most common technique is *moving average* smoothing which replaces each element of the series by either the simple or weighted average of *n* surrounding elements, where *n* is the width of the smoothing "window" (see Box & Jenkins, 1976; Velleman & Hoaglin, 1981). Medians can be used instead of means. The main advantage of median as compared to moving average smoothing is that its results are less biased by outliers (within the smoothing window). Thus, if there are outliers in the data (e.g., due to measurement errors), median smoothing typically produces smoother or at least more "reliable" curves than moving average based on the same window width. The main disadvantage of median smoothing is that in the absence of clear outliers it may produce more "jagged" curves than moving average and it does not allow for weighting.

In the relatively less common cases (in time series data), when the measurement error is very large, the *distance weighted least squares smoothing* or *negative exponentially weighted smoothing* techniques can be used. All those methods will filter out the noise and convert the data into a smooth curve that is relatively unbiased by outliers (see the respective sections on each of those methods for more details). Series with relatively few and systematically distributed points can be smoothed with *bicubic splines*.

Fitting a function. Many monotonous time series data can be adequately approximated by a linear function; if there is a clear monotonous nonlinear component, the data first need to be transformed to remove the nonlinearity. Usually a logarithmic, exponential, or (less often) polynomial function can be used.

Text Book

Data Mining, Concepts & Techniques Jiawei Han & Micheline Kamber,
Elsevier, II Edition, 2006