Q2. (a) List and describe the five primitives for specifying a data mining task.

Ans: Data Mining Task Primitives

Each user will have a **data mining task** in mind, that is, some form of data analysis that he or she would like to have performed. A data mining task can be specified in the form of a **data mining query**, which is input to the data mining system. A data mining query is defined in terms of **data mining task primitives**. These primitives allow the user to *interactively* communicate with the data mining system during discovery in order to direct the mining process, or examine the findings from different angles or depths. The data mining primitives specify the following, as illustrated in Figure 1.13.

- The set of *task-relevant data* to be mined: This specifies the portions of the database or the set of data in which the user is interested. This includes the database attributes or data warehouse dimensions of interest (referred to as the *relevant attributes or dimensions*).
- The *kind of knowledge* to be mined: This specifies the *data mining functions* to be performed, such as characterization, discrimination, association or correlation analysis, classification, prediction, clustering, outlier analysis, or evolution analysis.
- The *background knowledge* to be used in the discovery process: This knowledge about the domain to be mined is useful for guiding the knowledge discovery process and for evaluating the patterns found. *Concept hierarchies* are a popular form of background knowledge, which allow data to be mined at multiple levels of abstraction. An example of a concept hierarchy for the attribute (or dimension) *age* is shown in Figure 1.14. User beliefs regarding relationships in the data are another form of background knowledge.
- The *interestingness measures and thresholds* for pattern evaluation: They may be used to guide the mining process or, after discovery, to evaluate the discovered patterns. Different kinds of knowledge may have different interestingness measures. For example, interestingness measures for association rules include *support* and *confidence*. Rules whose support and confidence values are below user-specified thresholds are considered uninteresting.
- The expected *representation for visualizing* the discovered patterns: This refers to the form in which discovered patterns are to be displayed, which may include rules, tables, charts, graphs, decision trees, and cubes.

(b) How data mining is different from knowledge discovery in databases (KDD)? Explain.

Ans:

- *Presentation and visualization of data mining results:* Discovered knowledge should be expressed in high-level languages, visual representations, or other expressive forms so that the knowledge can be easily understood and directly usable by humans. This is especially crucial if the data mining system is to be interactive. This requires the system to adopt expressive knowledge representation techniques, such as trees, tables, rules, graphs, charts, crosstabs, matrices, or curves.
- *Handling noisy or incomplete data:* The data stored in a database may reflect noise, exceptional cases, or incomplete data objects. When mining data regularities, these objects may confuse the process, causing the knowledge model constructed to overfit the data. As a result, the accuracy of the discovered patterns can be poor. Data cleaning methods and data analysis methods that can handle noise are required, as well as outlier mining methods for the discovery and analysis of exceptional cases.

Q3(a) Use the two method below to normalize the following group of data:
200, 300, 400, 600, 1000
(a) min-max normalization by setting min = 0 and max = 1
(b) Z-score normalization

Ans:

Answer:

(a) min-max normalization by setting *min* = 0 and *max* = 1

original data	200	300	400	600	1000
[0,1] normalized	0	0.125	0.25	0.5	1

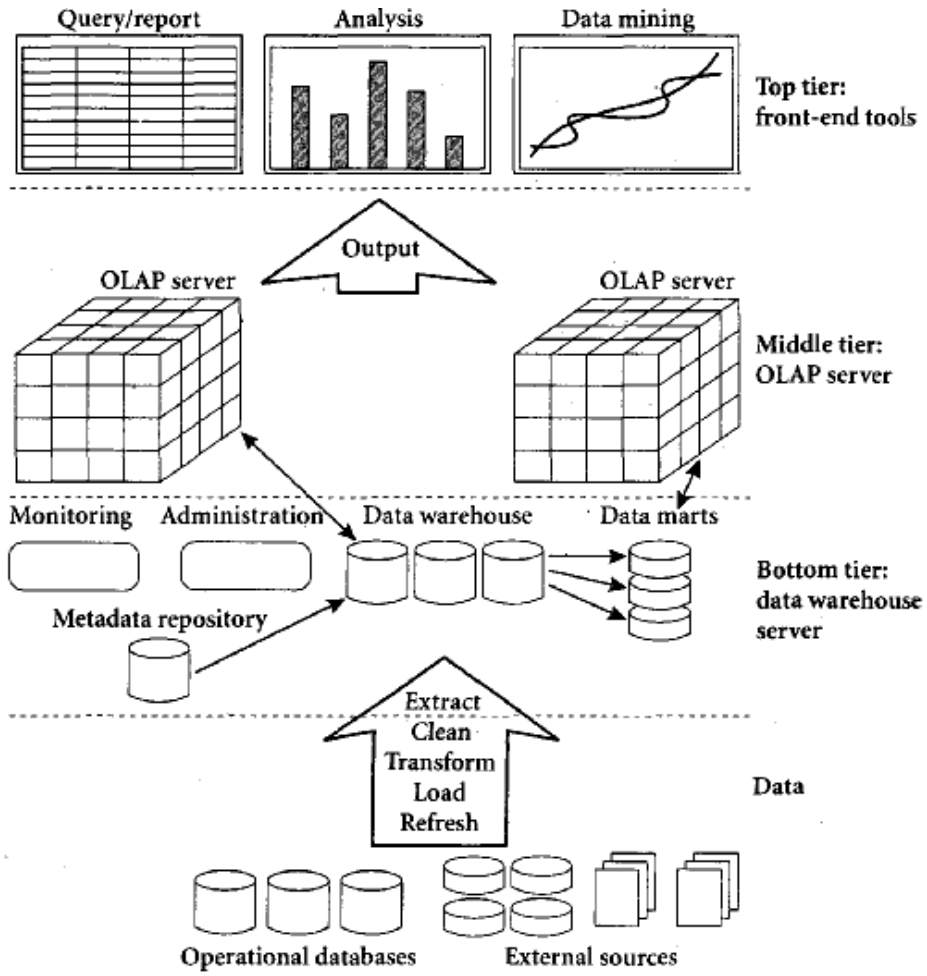
(b) z-score normalization

original data	200	300	400	600	1000
z-score	-1.06	-0.7	-0.35	0.35	1.78

■

(b) Explain the Three-tier data warehouse architecture. What are the three data warehouse models from architecture point of view.

Ans:



A three-tier data warehousing architecture.

1. The bottom tier is a **warehouse database server** that is almost always a relational database system. *"How are the data extracted from this tier in order to create the data warehouse?"* Data from operational databases and external sources (such as customer profile information provided by external consultants) are extracted using application program interfaces known as **gateways**. A gateway is supported by the underlying DBMS and allows client programs to generate SQL code to be executed at a server. Examples of gateways include ODBC (Open Database Connection) and OLE-DB (Open Linking and Embedding for Databases), by Microsoft, and JDBC (Java Database Connection).

2. The middle tier is an **OLAP server** that is typically implemented using either (1) a **relational OLAP (ROLAP)** model, that is, an extended relational DBMS that maps operations on multidimensional data to standard relational operations; or (2) a **multidimensional OLAP (MOLAP)** model, that is, a special-purpose server that directly implements multidimensional data and operations.

3. The top tier is a **client**, which contains query and reporting tools, analysis tools, and/or data mining tools (e.g., trend analysis, prediction, and so on).

Q4(a) Name the different types of problems in data, which the data-cleaning methods can deal. What are the different methods to deal with "missing values"?

Ans (a) Data cleaning routines attempt to fill in MISSING VALUES, smooth out NOISE while identifying outliers, and correct INCONSISTENCIES in data.

The following methods can be adopted to clean out missing values:

1. **Ignore the tuple**: This is usually done when the class label is missing (assuming the mining task involves classification or description). This method is not very effective, unless the tuple contains several attributes with missing values. It is especially poor when the percentage of missing values per attribute varies considerably.
2. **Fill in the missing value manually**: In general, this approach is time-consuming and may not be feasible given a large data set with many missing values.
3. **Use a global constant to fill in the missing value**: Replace all missing attribute values by the same constant, such as a label like "*Unknown*" or $-\infty$. If missing values are replaced by, say, "*Unknown*," then the mining program may mistakenly think that they form an interesting concept, since they all have a value in common—that of "*Unknown*." Hence, although this method is simple, it is not recommended.
4. **Use the attribute mean to fill in the missing value**: For example, suppose that the average income of customers is $28,000. Use this value to replace the missing value for *income*.
5. **Use the attribute mean for all samples belonging to the same class as the given tuple**: For example, if classifying customers according to *credit_risk*, replace the missing value with the average *income* value for customers in the same credit risk category as that of the given tuple.
6. **Use the most probable value to fill in the missing value**: This may be determined with regression, inference-based tools using a Bayesian formalism, or decision tree induction. For example, using the other customer attributes in your data set, you may construct a decision tree to predict the missing values for *income*.

(b) State why, for the integration of multiple heterogeneous information source, many companies in industry prefer the update-driven approach (which constructs and uses data warehouse), rather than the query-driven approach (which applies wrappers and integrators). Describe situation where the query-driven approach is preferable over the update-driven approach

Ans (b) For decision-making queries and frequently asked queries, the update-driven approach is more preferable. This is because expensive data integration and aggregate computation are done before query processing time. For the data collected in multiple heterogeneous databases to be used in decision-making processes, any semantic heterogeneity problems among multiple databases must be analyzed and solved so that the data can be integrated and summarized. If the query-driven approach is employed, these queries will be translated into multiple (often complex) queries for each individual database. The translated queries will compete for resources with the activities at the local sites, thus degrading their performance. In addition, these queries will generate a complex answer set, which will require further filtering and integration. Thus, the query-driven approach is, in general, inefficient and expensive. The update-driven approach employed in data warehousing is faster and more efficient since most of the queries needed could be done off-line. For queries that either are used rarely, reference the most current data, and/or do not require aggregations, the query-driven approach is preferable over the update-driven approach. In this case, it may not be justifiable for an organization to pay the heavy expenses of building and maintaining a data warehouse if only a small number and/or relatively small-sized databases are used. This is also the case if the queriesrely on the current data because data warehouses do not contain the most current information

Q5(a) How does data mining relate to information processing and OLAP. Discuss in detail.

Ans:

Information processing, based on queries, can find useful information. However, answers to such queries reflect the information directly stored in databases or computable by aggregate functions. They do not reflect sophisticated patterns or regularities buried in the database. Therefore, information processing is not data mining.

On-line analytical processing comes a step closer to data mining since it can derive information summarized at multiple granularities from user-specified subsets of a data warehouse. Such descriptions are equivalent to the class/concept descriptions discussed in Chapter 1. Since data mining systems can also mine generalized class/concept descriptions, this raises some interesting questions: *"Do OLAP systems perform data mining? Are OLAP systems actually data mining systems?"*

The functionalities of OLAP and data mining can be viewed as disjoint: OLAP is a data summarization/aggregation *tool* that helps simplify data analysis, while data mining allows the *automated discovery* of implicit patterns and interesting knowledge hidden in large amounts of data. OLAP tools are targeted toward simplifying and supporting interactive data analysis, whereas the goal of data mining tools is to automate as much of the process as possible, while still allowing users to guide the process. In this sense, data mining goes one step beyond traditional on-line analytical processing.

An alternative and broader view of data mining may be adopted in which data mining covers both data description and data modeling. Since OLAP systems can present general descriptions of data from data warehouses, OLAP functions are essentially for user-directed data summary and comparison (by drilling, pivoting, slicing, dicing, and other operations). These are, though limited, data mining functionalities. Yet according to this view, data mining covers a much broader spectrum than simple OLAP operations because it not only performs data summary and comparison, but also performs association, classification, prediction, clustering, time-series analysis, and other data analysis tasks.

Data mining is not confined to the analysis of data stored in data warehouses. It may analyze data existing at more detailed granularities than the summarized data provided in a data warehouse. It may also analyze transactional, spatial, textual, and multimedia data that are difficult to model with current multidimensional database technology. In this context, data mining covers a broader spectrum than OLAP with respect to data mining functionality and the complexity of the data handled.

Since data mining involves more automated and deeper analysis than OLAP, data mining is expected to have broader applications. Data mining can help business managers find and reach more suitable customers, as well as gain critical business insights that may help to drive market share and raise profits. In addition, data mining can help managers understand customer group characteristics and develop optimal pricing strategies accordingly, correct item bundling based not on intuition but on actual item groups derived from customer purchase patterns, reduce promotional spending, and at the same time increase the overall net effectiveness of promotions.

(b) For class characterization, what are the major differences between a data cube-based implementation and a relational implementation such as attribute-oriented induction? Discuss which method is most efficient and under what conditions this is so.

Ans (b): For class characterization, the major differences between a data cube-based implementation and a relational based implementation such as attribute-oriented induction include the following:

•Process control:

Under a data cube-based approach, the process is user-controlled at every step. This includes the selection of the relevant dimensions to be used as well as the application of OLAP operations such as roll-up, roll-down, slicing and dicing. A relational approach does not require user interaction at every step, however, as attribute relevance and ranking is performed automatically.

•Supported data types and measures:

The relational approach supports complex data types and measures, which restrictions in current OLAP technology do not allow. Thus, OLAP implementations are limited to a more simplified model for data analysis.

•Precomputation:

An OLAP-based implementation allows for the precomputation of measures at different levels of aggregation analysis.

•Precomputation:

An OLAP-based implementation allows for the precomputation of measures atdi fferent levels of aggregation

Based upon these differences, it is clear that a relational approach is more efficient when there are complex data types and measures being used, as well as when there are a very large number of attributes to be considered. This is due to the advantage that automation provides over the efforts that would be required by a user to perform the same tasks. However, when the data set being mined consists of regular data types and measures that are well supported by OLAP technology, and then the OLAP-based implementation provides an advantage in decency. This results from the time saved by using precomputed measures, as well as the flexibility in investigating mining results provided by OLAP functions.

Q6 (a) Discuss the criteria used to compare and evaluation of the classification and prediction method.

Ans: Classification and prediction methods can be compared and evaluated according to the following criteria:

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

ii. Speed: This refers to the computational costs involved in generating and using the given classifier or predictor.

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

iv. Scalability: This refers to the ability to construct the classifier or predictor efficiently given large amounts of data.

v. Interpretability: This refers to the level of understanding and insight that is provided by the classifier or predictor. Interpretability is subjective and therefore more difficult to assess.

(b) Association rule mining often generates a large number of rules. Discuss effective methods that can be used to reduce the number of rules generated while still preserving most of the interesting rules.

Ans: There are several approaches to reduce the number of rules. Here we list a few:

•Mine only closed frequent patterns to reduce the number of redundant rules.

•Use multilevel rule mining and generate lower-level rules only when they are nonredundant given the high-level rules. For example, we may find rules at the product category level first. If we find that milk→cheese[support =0.1,conf =0.9] and at the lower level we get milk 2%→provolone[support =0.01,conf =0.92], this may be redundant, i.e., this would be the expected support and confidence given the high-level rule.

•Use domain knowledge to define templates for the rules to be mined and define minimum support, confidence, and correlation measures

Q7 (a) What is boosting? State why it may improve the accuracy of decision tree induction.

Ans:

Boosting is a technique used to help improve classifier accuracy. We are given a set S of s tuples. For iteration t, where $t = 1, 2, \ldots, T$, a training set S_t is sampled with replacement from S. Assign weights to the tuples within that training set. Create a classifier, C_t from S_t. After C_t is created, update the weights of the tuples so that the tuples causing classification error will have a a greater probability of being selected for the next classifier constructed. This will help improve the accuracy of the next classifier, C_{t+1}. Using this technique, each classifier should have greater accuracy than its predecessor. The final boosting classifier combines the votes of each individual classifier, where the weight of each classifier's vote is a function of its accuracy.

■

(b) How classification is done by back-propagation. Give an example of a general multilayered feed-forward neural network.

Ans:

Backpropagation learns by iteratively processing a set of training samples, comparing the network's prediction for each sample with the actual known class label. For each training sample, the weights are modified so as to minimize the mean squared error between the network's prediction and the actual class. These modifications are made in the "backwards" direction, that is, from the output layer, through each hidden layer down to the first hidden layer (hence the name *backpropagation*). Although it is not guaranteed, in general the weights will eventually converge, and the learning process stops.

The backpropagation algorithm performs learning on a **multilayer feed-forward** neural network. An example of such a network is shown in Figure . The inputs correspond to the attributes measured for each training sample. The inputs are fed simultaneously into a layer of units making up the **input layer**. The weighted outputs of these units are, in turn, fed simultaneously to a second layer of "neuronlike" units, known as a **hidden layer**. The hidden layer's weighted outputs can be input to another hidden layer, and so on. The number of hidden layers is arbitrary, although in practice, usually only one is used. The weighted outputs of the last hidden layer are input to units making up the **output layer**, which emits the network's prediction for given samples.

The units in the hidden layers and output layer are sometimes referred to as **neurodes**, due to their symbolic biological basis, or as **output units**. The multi-

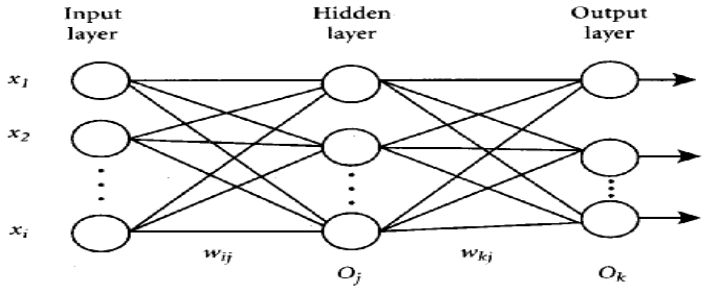


Figure A multilayer feed-forward neural network: A training sample, $X = (x_1, x_2, \ldots, x_i)$, is fed to the input layer. Weighted connections exist between each layer, where w_{ij} denotes the weight from a unit j in one layer to a unit i in the previous layer.

layer neural network shown in Figure has two layers of output units. Therefore, we say that it is a **two-layer** neural network. Similarly, a network containing two hidden layers is called a *three-layer* neural network, and so on. The network is **feed-forward** in that none of the weights cycles back to an input unit or to an output unit of a previous layer. It is **fully connected** in that each unit provides input to each unit in the next forward layer.

Multilayer feed-forward networks of linear threshold functions, given enough hidden units, can closely approximate any function.

Q8 (a) Discuss the various types of typical requirements for clustering in data-mining.

Ans: (a) The following are the typical requirements for clustering in data mining:

- **Scalability:** Many clustering algorithms work well on small data sets containing fewer than 200 data objects; however, a large database may contain millions of objects. Clustering on a *sample* of a given large data set may lead to biased results. Highly scalable clustering algorithms are needed.
- **Ability to deal with different types of attributes:** Many algorithms are designed to cluster interval-based (numerical) data. However, applications may require clustering other types of data, such as binary, categorical (nominal), and ordinal data, or mixtures of these data types.
- **Discovery of clusters with arbitrary shape:** Many clustering algorithms determine clusters based on Euclidean or Manhattan distance measures. Algorithms based on such distance measures tend to find spherical clusters with similar size and density. However, a cluster could be of any shape. It is important to develop algorithms that can detect clusters of arbitrary shape.
- **Minimal requirements for domain knowledge to determine input parameters:** Many clustering algorithms require users to input certain parameters in cluster analysis (such as the number of desired clusters). The clustering results can be quite sensitive to input parameters. Parameters are often hard to determine, especially for data sets containing high-dimensional objects. This not only burdens users, but also makes the quality of clustering difficult to control.
- **Ability to deal with noisy data:** Most real-world databases contain outliers or missing, unknown, or erroneous data. Some clustering algorithms are sensitive to such data and may lead to clusters of poor quality.
- **Insensitivity to the order of input records:** Some clustering algorithms are sensitive to the order of input data; for example, the same set of data, when presented with different orderings to such an algorithm, may generate dramatically different clusters. It is important to develop algorithms that are insensitive to the order of input.
- **High dimensionality:** A database or a data warehouse can contain several dimensions or attributes. Many clustering algorithms are good at handling low-dimensional data, involving only two to three dimensions. Human eyes are good at judging the quality of clustering for up to three dimensions. It is challenging to cluster data objects in high-dimensional space, especially considering that such data can be very sparse and highly skewed.
- **Constraint-based clustering:** Real-world applications may need to perform clustering under various kinds of constraints. Suppose that your job is to choose the locations for a given number of new automatic cash-dispensing machines (i.e., ATMs) in a city. To decide upon this, you may cluster households while considering constraints such as the city's rivers and highway networks, and customer requirements per region. A challenging task is to find groups of data with good clustering behavior that satisfy specified constraints.
- **Interpretability and usability:** Users expect clustering results to be interpretable, comprehensible, and usable. That is, clustering may need to be tied up with specific semantic interpretations and applications. It is important to study how an application goal may influence the selection of clustering methods.

(b) What are model based clustering methods. Discuss the two major approaches viz. Statistical approach and neural network based approach of clustering based methods.

Ans: Model-based clustering methods attempt to optimize the fit between the given data and some mathematical model. Such methods are often based on the assumption that the data are generated by a mixture of underlying probability distributions. Model based clustering methods follow two basic approaches viz. statistical approach and neural network based approach. Refer book chapter 8.

Q9 (a) Discuss how the data mining can be applied for Biomedical & DNA Data analysis.

Ans:

The past decade has seen an explosive growth in biomedical research, ranging from the development of new pharmaceuticals and advances in cancer therapies to the identification and study of the human genome by discovering large-scale sequencing patterns and gene functions. Since a great deal of biomedical research has focused on DNA data anlysis, we study this application here. Recent research in DNA analysis has led to the discovery of genetic causes for many diseases and disabilities, as well as the discovery of new medicines and approaches for disease diagnosis, prevention, and treatment.

An important focus in genome research is the study of DNA sequences since such sequences form the foundation of the genetic codes of all living organisms. All DNA sequences are comprised of four basic building blocks (called *nucleotides*): adenine (A), cytosine (C), guanine (G), and thymine (T). These four nucleotides are combined to form long sequences or chains that resemble a twisted ladder.

Human beings have around 100,000 genes. A gene is usually comprised of hundreds of individual nucleotides arranged in a particular order. There are almost an unlimited number of ways that the nucleotides can be ordered and sequenced to form distinct genes. It is challenging to identify particular gene sequence patterns that play roles in various diseases. Since many interesting sequential pattern analysis and similarity search techniques have been developed in data mining, data mining has become a powerful tool and contributes substantially to DNA analysis in the following ways.

Semantic integration of heterogeneous, distributed genome databases: Due to the highly distributed, uncontrolled generation and use of a wide variety of DNA data, the semantic integration of such heterogeneous and widely distributed genome databases becomes an important task for systematic and coordinated analysis of DNA databases. This has promoted the development of integrated data warehouses and distributed federated databases to store and manage the primary and derived genetic data. Data cleaning and data integration methods developed in data mining will help the integration of genetic data and the construction of data warehouses for genetic data analysis.

Similarity search and comparison among DNA sequences: We have studied similarity search methods in time-series data mining. One of the most important search problems in genetic analysis is similarity search and comparison among DNA sequences. Gene sequences isolated from diseased and healthy tissues can be compared to identify critical differences between the two classes of genes. This can be done by first retrieving the gene sequences from the two tissue classes, and then finding and comparing the frequently occurring patterns of each class. Usually, sequences occurring more frequently in the diseased samples than in the healthy samples might indicate the genetic factors of the disease; on the other hand, those occurring only more frequently in the healthy samples might indicate mechanisms that protect the body from the disease. Notice that although genetic analysis requires similarity search, the technique needed here is quite different from that used for time-series data. For example, data transformation methods such as scaling, normalization, and window stitch-

(b) What are the major challenges faced in bringing data mining research to market? Illustrate one data mining research issue that, in your view, may have a strong impact on the market and on society. Discuss how to approach such a research issue.

Ans: Due to the high demand for transforming huge amounts of data found in databases and other information repositories into useful knowledge, it is likely that data mining will become a thriving market. There are, however, several bottlenecks remaining for data mining research and development. These include:

The handling of increasingly complex data: Such data include unstructured data from hypertext, documents, spatial and multimedia data, as well as from legacy databases, active databases, and the Internet.

•Visualization and data mining: The visualization of database contents would help users comprehend mining results and redirect miners in the search for promising patterns. This requires the development of easy-to-use and “easy-to-see” tools.

• The integration of mined knowledge into a knowledge-base, an expert system, a decision support system, or even a query optimizer.

• Market or domain-specific in-depth data mining with the goal of providing business-specific data mining solutions.

• Invisible data mining, where systems make implicit use of built-in data mining functions

Many may believe that the current approach to data mining has not yet won a large share of the market for system applications owing to the fact that the importance and usefulness of this kind of knowledge has not completely been made aware to the public and the market. Currently, not every university offers undergraduate courses on this topic in computing science departments. Offering more courses on data mining may be a good start. Furthermore, success stories regarding the use of data mining could be featured more prominently in the media.

TextBook

Data Mining, Concepts and Techniques, Jiawei Han and Micheline Kamber, Elsevier, Second Edition