Volume 3, Issue 8, August 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Data Mining: Techniques and Algorithms

Divya Chaudhary*

Department of Computer Science & Applications
M.D. University, India

Abstract*— Data mining is the process of extraction of relevant information from data warehouse. It also refers to the analysis of the data using pattern matching techniques. With the continuous and extensive use of database for storage, there arises a need for the database management and retrieval of the required information. This paper discusses the data mining techniques used for the knowledge discovery of the databases. It also surveys the various data mining algorithms for the optimized mining of information.

***Keywords*— Data Mining, Apriori algorithm, KDD, k-means algorithm, AdaBoost algorithm**

I. INTRODUCTION

With the rapid growth and the development of the society along with rises the need of storing of the data leading to creation of huge number of databases. A large number of databases give way to the creation of data warehouses. A data warehouse refers to a central repository created by integration of data from one or more databases. It stores both the current as well as the historical data. They help in the creation of the trending reports using the information stored. Data warehouses are subdivided into data marts. A data mart refers to the storage of the related information.

In essence, the goal of data mining is to extract knowledge from data. Data mining is an inter-disciplinary field, whose core is at the intersection of machine learning, statistics and databases. We emphasize that in data mining – unlike for example in classical statistics – the goal is to discover knowledge that is not only accurate but also comprehensible for the user. Comprehensibility is important whenever discovered knowledge will be used for supporting a decision made by a human user. After all, if discovered knowledge is not comprehensible for the user, he/she will not be able to interpret and validate it. In this case, probably the user will not trust enough the discovered knowledge to use it for decision making. This can lead to wrong decisions.

There are several data mining tasks, including classification, regression, clustering, dependence modelling, etc. Each of these tasks can be regarded as a kind of problem to be solved by a data mining algorithm. Therefore, the first step in designing a data mining algorithm is to define which task the algorithm will address.

The continuous [1] development of database technology and the extensive applications of database management system, the data volume stored in database increases rapidly and in the large amounts of data much important information is hidden. If the information can be extracted from the database they will create a lot of potential profit for the companies and the technology of mining information from the massive [2] database is known as data mining.

Data can now be stored in many different types of databases. One database architecture that has recently emerged is the data warehouse, a repository of multiple heterogeneous data sources, organized under a unified schema at a single site in order to facilitate management decision making. Data warehouse technology includes data cleansing, data integration, and On-Line Analytical Processing (OLAP), that is, analysis techniques with functionalities such as summarization, consolidation and aggregation, as well as the ability to view information at different angles.

Data mining tools can forecast the future trends and activities to support the decision of people. For example, through analysing the whole database system of the company the data mining tools [3] can answer the problems such as “Which customer is most likely to respond to the e-mail marketing activities of our company, why”, and other similar problems. Some data mining tools can also resolve some traditional problems which consumed much time, this is because that they can rapidly browse the entire database and find some useful information experts unnoticed.

The rest of this paper is organized as follows. The concepts of data mining are discussed in Section II. It also describes the process of discovery of data. The emerging algorithms for knowledge extraction are discussed in Section III. It highlights various algorithms used for knowledge extraction with a number of security solutions. Finally, the conclusions and the future works are discussed in Section IV.

II. DATA MINING

Data mining is an inter-disciplinary field, whose core is at the intersection of machine learning, statistics and databases. The abundance of data, coupled with the need for powerful data analysis tools, has been described as a data rich but “information poor” situation. The fast-growing, tremendous amount of data, collected and stored in large and numerous databases, has far exceeded our human ability for comprehension [4] without powerful tools. As a result, data collected in large databases become “data tombs” data archives that are seldom revisited. Consequently, important decisions are often made based not on the information-rich data stored in databases but rather on a decision maker's intuition, simply because the decision maker does not have the tools to extract the valuable knowledge embedded in the vast amounts of data [5]. A data warehouse is a repository of information collected from multiple sources, stored under a unified schema, and which usually resides at a single site. Data warehouses are constructed via a process of data cleansing, data transformation, data integration, data loading, and periodic data refreshing.

Simply stated, data mining refers to extracting or "mining" knowledge from large amounts of data. The term is actually a misnomer. For example the mining of gold from rocks or sand is referred to as gold mining rather than rock or sand mining. Thus, "data mining" should have been more appropriately named "knowledge mining from data", which is unfortunately somewhat long. [6] "Knowledge mining", a shorter term, may not reflect the emphasis on mining from large amounts of data. Nevertheless, mining is a vivid term characterizing the process that finds a small set of precious nuggets from a great deal of raw material. Thus, such a misnomer which carries both "data" and "mining" became a popular choice. There are many other terms carrying a similar or slightly different meaning to data mining, such as knowledge mining from databases, knowledge extraction, data/pattern analysis, data archaeology, and data dredging.

A. Data Mining Process

Knowledge discovery as a process consists of an iterative sequence of the following steps [7]:

- Data Cleaning:
  To remove noise or irrelevant data
- Data Integration:
  Where multiple data sources may be combined
- Data Selection:
  Where data relevant to the analysis task are retrieved from the database
- Data Transformation:
  Where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations
- Data Mining:
  An essential process where intelligent methods are applied in order to extract data patterns
- Pattern Evaluation:
  To identify the truly interesting patterns representing knowledge based on some interestingness measures
- Knowledge Presentation:
  Where visualization and knowledge representation techniques are used to present the mined knowledge to the user

Therefore,

"Data Mining is the process of discovering interesting knowledge from large amounts of data stored either in databases, data warehouses, or other information repositories."

B. Data Mining Architecture

Based on this view, the architecture of a typical data mining system may have the [8] following major components:

1. **Database, data warehouse, or other information repository.** This is one or a set of databases, data warehouses, spread sheets, or other kinds of information repositories. Data cleaning and data integration techniques may be performed on the data.

2. **Database or data warehouse server.** The database or data warehouse server is responsible for fetching the relevant data, based on the user's data mining request.

3. **Knowledge base.** This is the domain knowledge that is used to guide the search, or evaluate the interestingness of resulting patterns. Such knowledge can include concept hierarchies, used to organize attributes or attribute values into different levels of abstraction. Knowledge such as user beliefs, which can be used to assess a pattern's interestingness based on its unexpectedness, may also be included. Other examples of domain knowledge are additional interestingness constraints or thresholds, and metadata (e.g., describing data from multiple heterogeneous sources).

4. **Data mining engine.** This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association analysis, classification, and evolution and deviation analysis.

5. **Pattern evaluation module.** This component typically employs interestingness measures and interacts with the data mining modules so as to focus the search towards interesting patterns. It may access interestingness thresholds stored in the knowledge base. Alternatively, the pattern evaluation module may be integrated with the mining module, depending on the implementation of the data mining method used. For efficient data mining, it is highly recommended to push the evaluation of pattern interestingness as deep as possible into the mining process so as to confine the search to only the interesting patterns.

6. **Graphical user interface.** This module communicates between users and the data mining system, allowing the user to interact with the system by specifying a data mining query or task, providing information to help focus the search, and performing exploratory data mining based on the intermediate data mining results. [9] In addition, this component allows the user to browse database and data warehouse schemas or data structures, evaluate mined patterns, and visualize the patterns in different forms.

Data mining involves an integration of techniques from multiple disciplines such as database technology, statistics, machine learning, high performance computing, pattern recognition, neural networks, data visualization, information retrieval, image and signal processing, and spatial data analysis. In principle, data mining should be applicable to any kind of information repository. This includes relational databases, data warehouses, transactional databases, advanced database systems, flat files, and the World-Wide Web. [10] Advanced database systems include object-oriented and object-relational databases, and specific application-oriented databases, such as spatial databases, time-series databases, text databases, and multimedia databases. The challenges and techniques of mining may differ for each of the repository systems.

Data Mining tasks can be classified into two categories: descriptive and predictive.

- Descriptive mining tasks characterize the general properties of the data in the database.
- Predictive mining tasks perform inference on the current data in order to make predictions.

III. DATA MINING ALGORITHMS

The algorithms used for speeding up the data mining process are:

A. *C4.5*

Systems that construct classifiers are one of the commonly used tools in data mining. Such systems take as input a collection of cases, each belonging to one of a small number of classes and described by its values for a fixed set of attributes, and output a classifier that can accurately predict the class to which a new case belongs. C4.5 generates classifiers expressed as decision trees, [11] but it can also construct classifiers in more comprehensible rule set form. It uses two heuristic criteria to rank possible tests: information gain, which minimizes the total entropy of the subsets {*Si}* (but is heavily biased towards tests with numerous outcomes), and the default gain ratio that divides information gain by the information provided by the test outcomes.

Attributes can be either numeric or nominal and this determines the format of the test outcomes. An attribute *A* with discrete values has by default one outcome for each value, but an option allows the values to be grouped into two or more subsets with one outcome for each subset.

To avoid over fitting, the initial tree is then pruned. The pruning algorithm is based on a pessimistic estimate of the error rate associated with a set of *N* cases, *E* of which do not belong to the most frequent class. Instead of *E/N*, C4.5 determines the upper limit of the binomial probability when *E* events have been observed in *N* trials, using a user-specified confidence whose default value is 0.25. Pruning is carried out from the leaves to the root. The estimated error at a leaf with *N* cases and *E* errors is *N* times the pessimistic error rate as above. CART prunes trees using a cost-complexity model whose parameters are estimated by cross-validation; C4.5 uses a single-pass algorithm derived from binomial confidence limits. A hill-climbing [12] algorithm is used to drop conditions until the lowest pessimistic error rate is found. Greatly improved scalability of both decision trees and (particularly) rule sets is seen. Scalability is enhanced by multi-threading; C5.0 can take advantage of computers with multiple CPUs and/or cores.

B. *The K-Means Algorithm*

The k-means algorithm is a simple iterative method to partition a given dataset into a user specified number of clusters, *k*. Gray and Neuhoff [34] provide a nice historical background for k-means placed in the larger context of hill-climbing algorithms.

The algorithm operates on a set of *d*-dimensional vectors, *D* = {**xi** | *i* = 1, . . . , *N*}, where **xi** ∈ _*d* denotes the *i*th data point. The algorithm is initialized by picking *k* points in _*d* as the initial *k* cluster representatives or "centroids". Techniques for selecting these initial seeds include sampling at random from the dataset, setting them as the solution of clustering a small subset of the data or perturbing the global mean of the data *k* times. Then the algorithm iterates between two steps till convergence:

- *Data Assignment*. Each data point is assigned to its *closest* centroid, with ties broken arbitrarily. This results in a partitioning of the data.
- *Relocation of "means"*. Each cluster representative is relocated to the center (mean) of all data points assigned to it. If the data points come with a probability measure (weights), then the relocation is to the expectations (weighted mean) of the data partitions.

The algorithm converges when the assignments (and hence the **cj** values) no longer change. Each iteration needs *N* × *k* comparisons, which determines the time complexity of one iteration. [13] The number of iterations required for convergence varies and may depend on *N*, but as a first cut, this algorithm can be considered linear in the dataset size.

The greedy-descent nature of k-means on a non-convex cost also implies that the convergence is only to a local optimum, and indeed the algorithm is typically quite sensitive to the initial centroid locations. The local minima problem can be countered to some extent by running the algorithm multiple times with different initial centroids, or by doing limited local search about the converged solution.

k-means is a limiting case of fitting data by a mixture of *k* Gaussians with identical, isotropic covariance matrices (_ = *σ*2**I**), when the soft assignments of data points to mixture components are hardened to allocate each data point solely to the most likely component. So, it will falter whenever the data is not well described by reasonably separated spherical balls, for example, if there are non-convex shaped clusters in the data. The cost of the optimal solution decreases with increasing *k* till it hits zero when the number of clusters equals the number of distinct data-points. This makes it more difficult to (a) directly compare solutions with different numbers of clusters and (b) to find the optimum value of *k*. If the desired *k* is not known in advance, one will typically run k-means with different values of *k*, and then use a suitable criterion to select one of the results. [14] For example, SAS uses the cube-clustering-criterion, while X-means adds a complexity term (which increases with *k*) to the original cost function and then identifies the *k* which minimizes this adjusted cost. Alternatively, one can progressively increase the number of clusters, in conjunction with a suitable stopping criterion. Bisecting k-means achieves this by first putting all the data into a single cluster, and then recursively splitting the least compact cluster into two using 2-means. The celebrated LBG algorithm used for vector quantization doubles the number of clusters till a suitable code-book size is obtained. Both these approaches thus alleviate the need to know *k* beforehand.

K-Means remains the most widely used partitioned clustering algorithm as it is simple, easily understandable and reasonably scalable and can be easily modified to deal with streaming data.

C. The Apriori Algorithm

It is one of the most popular data mining approaches is to find frequent item sets from a transaction dataset and derive association rules. Finding frequent itemsets (itemsets with frequency larger than or equal to a user specified minimum support) is not trivial because of its combinatorial explosion. Once frequent itemsets are obtained, it is straightforward to generate association rules with confidence larger than or equal to a user specified minimum confidence.

Apriori is a seminal algorithm for finding frequent itemsets using candidate generation. It is characterized as a level-wise complete search algorithm using anti-monotonicity of itemsets, "if an itemset is not frequent, any of its superset is never frequent". [15] By convention, Apriori assumes that items within a transaction or itemset are sorted in lexicographic order.

Let the set of frequent itemsets of size *k* be *Fk* and their candidates be *Ck*. Apriori first scans the database and searches for frequent itemsets of size 1 by accumulating the count for each item and collecting those that satisfy the minimum support requirement. It then iterates on the following three steps and extracts all the frequent itemsets.

1. Generate *Ck*+1, candidates of frequent itemsets of size *k* +1, from the frequent itemsets of size *k*.
2. Scan the database and calculate the support of each candidate of frequent itemsets.
3. Add those itemsets that satisfies the minimum support requirement to *Fk*+1.

Function apriori-gen in line 3 generates *Ck*+1 from *Fk* in the following two step process:

1. Join step:
2. Prune step:

The Apriori achieves good performance by reducing the size of candidate sets. However, in situations with very many frequent itemsets, large itemsets, or very low minimum support, it still suffers from the cost of generating a huge number of candidate sets and scanning the database repeatedly to check a large set of candidate itemsets The most outstanding improvement over Apriori would be a method called FP-growth (frequent pattern growth) that succeeded in eliminating candidate generation. It adopts a divide and conquer strategy by compressing the database representing frequent items into a structure called FP-tree (frequent pattern tree) that retains all the essential information and dividing the compressed database into a set of conditional databases, each associated with one frequent itemset and mining each one separately. Apriori SMP uses this principle using richer expressions than itemset.

D. The EM algorithm

Finite mixture distributions provide a flexible and mathematical-based approach to the modelling and clustering of data observed on random phenomena. We focus here on the use of normal mixture models, which can be used to cluster continuous data and to estimate the underlying density function. [14] These mixture models can be fitted by maximum likelihood via the EM (Expectation–Maximization) algorithm. Finite mixture models are being increasingly used to model the distributions of a wide variety of random phenomena and to cluster data sets. These E- and M-steps are alternated until the changes in the estimated parameters or the log likelihood are less than some specified threshold.

E. AdaBoost

It deals with methods which employ multiple learners to solve a problem. The generalization ability of an ensemble is usually significantly better than that of a single learner, so ensemble methods are very attractive. [16] The AdaBoost algorithm proposed by Yoav Freund and Robert Schapire is one of the most important ensemble methods, since it has solid theoretical foundation, very accurate prediction, great simplicity, and wide and successful applications.

In order to deal with multi-class problems, Freund and Schapire presented the AdaBoost.M1 algorithm which requires that the weak learners are strong enough even on hard distributions generated during the AdaBoost process. Another popular multi-class version of AdaBoost is AdaBoost.MH which works by decomposing multi-class task to a series of binary tasks. Boosting has become the most important "family" of ensemble methods.

AdaBoost and its variants have been applied to diverse domains with great success. Many empirical study shows that AdaBoost often does not over fit, i.e., the test error of AdaBoost often tends to decrease even after the training error is zero.

Many real-world applications are born with high dimensionality, i.e., with a large amount of input features. There are two paradigms that can help us to deal with such kind of data, i.e., dimension reduction and feature selection. Dimension reduction methods are usually based on mathematical projections, which attempt to transform the original features into an appropriate feature space. After dimension reduction, the original meaning of the features is usually lost. Feature selection methods directly select some original features to use, and therefore they can preserve the original meaning of the features, which is very desirable in many applications.

IV. CONCLUSIONS AND FUTURE SCOPE

This paper presents a detailed description of the data mining concepts. Data mining is an inter-disciplinary field, whose core is at the intersection of machine learning, statistics and databases. The abundance of data, coupled with the need for powerful data analysis tools, has been described as a data rich but "information poor" situation. Therefore, Data Mining is the process of discovering interesting knowledge from large amounts of data stored either in databases, data warehouses, or other information repositories. The process and the architecture of data mining are elaborated. The process of knowledge discovery is well depicted. The various algorithms used for the mining of data are specified in detail. The data extraction process offers a huge potential for the future development in this field. The knowledge discovery of the algorithms could be speed up by generation of new data mining algorithms. The future scope provides enhancement and efficiency of data in the system. They could lead to better, faster and qualitative exaction of data with better tools and techniques.

REFERENCES

[1] Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proceedings of the 20th VLDB conference, pp. 487–499

[2] Ahmed S, Coenen F, Leng PH (2006) Tree-based partitioning of date for association rule mining. Knowl Inf Syst 10(3):315–331

[3] Banerjee A, Merugu S, Dhillon I, Ghosh J (2005) Clustering with Bregman divergences. J Mach Learn Res 6:1705–1749

[4] Bezdek JC, Chuah SK, Leep D (1986) Generalized k-nearest neighbour rules. Fuzzy Sets Syst 18(3):237–256. http://dx.doi.org/10.1016/0165-0114(86)90004-7

[5] Bloch DA, Olshen RA, Walker MG (2002) Risk estimation for classification trees. J Comput Graph Stat 11:263–288

[6] Bonchi F, Lucchese C (2006) on condensed representations of constrained frequent patterns. Knowl Inf Syst 9(2):180–201

[7] Breiman L (1968) Probability theory. Addison-Wesley, Reading. Republished (1991) in Classics of mathematics. SIAM, Philadelphia

[8] Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Wadsworth, Belmont

[9] Brin S, Page L (1998) The anatomy of a large-scale hypertextualWeb Search Engine. Comput Networks 30(1–7):107–117

[10] Cheung D W, Han J, Ng V, Wong C Y (1996) Maintenance of discovered association rules in large databases: an incremental updating technique. In: Proceedings of the ACM SIGMOD international conference on management of data, pp. 13–23

[11] Chi Y, Wang H, Yu PS, Muntz RR (2006) Catch the moment: maintaining closed frequent itemsets over a data stream sliding window. Knowl Inf Syst 10(3):265–294

[12] Cost S, Salzberg S (1993) A weighted nearest neighbour algorithm for learning with symbolic features. Mach Learn 10:57.78 (PEBLS: Parallel Exemplar-Based Learning System)

[13] Kuramochi M, Karypis G (2005) Gene Classification using Expression Profiles: A Feasibility Study. Int J Artif Intell Tools 14(4):641–660

[14] Langville AN, Meyer CD (2006) Google’s PageRank and beyond: the science of search engine rankings. Princeton University Press, Princeton

[15] Leung CW-k, Chan SC-f, Chung F-L (2006) A collaborative filtering framework based on fuzzy association rules and multiple-level similarity. Knowl Inf Syst 10(3):357–381

[16] Li T, Zhu S, Ogihara M (2006) Using discriminant analysis for multi-class classification: an experimental investigation. Knowl Inf Syst 10(4):453–472