International Journal of Computer Architecture and Mobility
(ISSN 2319-9229) Volume 3 -Issue 8, October 2015

A Comparative Analysis of Data Mining Techniques with Major Issues

Pavitra Sharma
Oriental University, Indore
sharma.pavitra07@gmail.com

Abstract *- In today's scenario data is increasing at a tremendous speed. It is very necessary to find some useful and novel information out of the large amount of data. Data mining is the process to find novel and useful patterns. In order to find such novel and useful pattern various data mining techniques exists for classification, clustering and association analysis, like Apriori algorithm, FP-growth, K-mean, K-Nearest-Neighbour, bagging and boosting. This paper includes comparison of data mining techniques on the basis of various parameters such as applicability, advantages, disadvantages, review current trends of data mining techniques. Proposal also focuses on major issues of data mining.*

I. INTRODUCTION

With the tremendous improvement in the speed of computer and the decreasing cost of data storage, huge volumes of data are created. However, data itself has no value. Only if data can be changed to information, it be-comes useful. In order to generate meaningful information, or knowledge from database, the field of data mining was born. It has been observed that the traditional statistical techniques were not adequate to handle the mass amount of data. There is need to recognize the better, faster and cheaper ways to deal with the dramatic increase in the amount of data [1].

In this paper, we would first give the basic principles of some commonly used data mining techniques with their advantages and disadvantages. Next, we discuss some issues and challenges of data mining.

II. APRIORI ALGORITHM

There are several mining algorithms of association rules. One of the most popular algorithms is Apriori
that is used to extract frequent itemsets from large database and getting the association rule for discovering the knowledge[2]. Steps of Apriori algorithm:-

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.

III. ADVANTAGES OF APRIORI

1. Uses large itemset property
2. Easily parallelized
3. Easy to implement

IV. DISADVANTAGES OF APRIORI ALGORITHM

1. Needs several iterations of the data.
2. Uses a uniform minimum support threshold.
3. Difficulties to find rarely occurring events.
4. Alternative methods (other than appriori) can address this by using a non-uniform minimum support threshold
5. Some competing alternative approaches focus on partition and sampling

V. FP-GROWTH ALGORITHM

The FP-Growth Algorithm is an alternative way to find frequent itemsets without using candidate generations, thus improving performance[3]. For so much it uses a divide-and-conquer strategy. The core of this method is the usage of a special data structure named frequent-pattern tree (FP-tree), which retains the itemset association information.

Two step approach:

Step 1: Build a compact data structure called the FP-tree
Built using 2 passes over the data-set.

Step 2: Extracts frequent itemsets directly from the FP-tree
Traversal through FP-Tree.

VI. ADVANTAGES OF FP-GROWTH

1. Only two passes over data-set.
2. Compresses data-set
3. No candidate generation
4. Much faster than Apriori.

A. Disadvantages of FP-Growth

1. FP-Tree may not store in memory.
2. FP-Tree is expensive to build.

VII. K-MEAN ALGORITHM

The k-means algorithm is an algorithm to cluster n objects based on attributes into k partitions, where k < n. Simply, k-means clustering is an algorithm to classify or to group the objects based on attributes/features into K number of group. K is positive integer number. The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

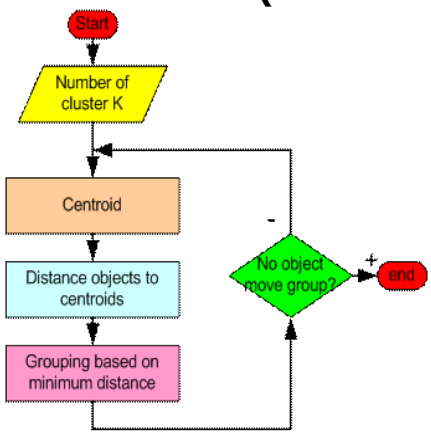

Fig. 1 Working Of K-MEAN Clustering Algorithm

VIII. DISADVANTAGES OF K-MEAN CLUSTERING

1. When the numbers of data are not so many, initial grouping will determine the cluster significantly.
2. The number of cluster, K, must be determined beforehand.
3. We never know the real cluster, using the same data.
4. It is sensitive to initial condition. Different initial condition may produce different result of cluster.
5. The algorithm may be trapped in the local optimum.

IX. K-NEAREST NEIGHBOR ALGORITHM

A nearest neighbor classifier is a technique for classifying elements based on the classification of the elements in the training set that are most similar to the test example. With the k-nearest neighbor technique, this is done by evaluating the k number of closest neighbors [4]. In pseudo code, k-nearest neighbor classification algorithm can be expressed fairly compactly [4]:-

k - number of nearest neighbors
for each object *X* in the test set **do**
calculate the distance D(*X*,*Y*) between *X* and every object *Y* in the training set
neighborhood <- the *k* neighbors in the training set closest to *X*
X.class <-SelectClass(*neighborhood*)
end for

All of the training samples are stored in an n-dimensional pattern space. When given an unknown sample, a k-nearest neighbor classifier searches the pattern space for the k training samples that are closest to the unknown sample. "Closeness" is defined in terms of Euclidean distance, where the Euclidean distance, where the Euclidean distance between two points, X=(x1,x2,……,xn) and Y=(y1,y2,….,yn) is [7] :

$$d(X, Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

X and Y are the two compared objects and n is their number of attributes. The overlap metric simply tests for equality between two values, so that different values get distance 1 whereas equal values get distance 0[5]:

$d_{doverlap}(x , y) = 0$ when $x=y$

And

$d_{overlap}(x , y) = 1$, when $x \neq y$

The unknown sample is assigned the most common class among its k nearest neighbors. When k=1, the unknown sample is assigned the class of the training sample that is closest to it in pattern space. Nearest neighbor classifiers are instance-based or lazy learners in that they store all of the training samples and do not build a classifier until a new (unlabeled) sample needs to be classified. This contrasts with eager learning methods, such a decision tree induction and back propagation, which construct a generalization model before receiving new samples to classify.

The *k-nearest neighbors' algorithm* is amongst the simplest of all machine learning algorithms. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its *k* nearest neighbors. *k* is a positive integer, typically small. If $k = 1$, then the object is simply assigned to the class of its nearest neighbor. In binary (two class) classification problems, it is helpful to choose *k* to be an odd number as this avoids tied votes.

X. BAGGING

A learning algorithm is unstable for small data set if small changes in the training data set will generate very diverse classifiers. Breiman [6] proposed the use of bagging to improve performance by taking advantage of this effect. A learning algorithm combination is informally called unstable if "small" changes in the training data lead to significantly different classifiers and relatively "large" changes in accuracy. In general, bagging improves recognition for unstable classifiers because it effectively averages over such discontinuities. However, there are no convincing theoretical derivations or simulation studies showing that bagging will help all unstable classifiers.

XI. BAGGING ALGORITHM IS DESCRIBED IN DETAIL AS FOLLOWS:

Step1. First we get the training data set which has samples, then set the cycle times *K* and selected number *nnN*< when we do bagging.

Step2. Sampling randomly from data set *D* and receiving a new data set D_k. For any given algorithm, a general classifier C_k will be generated from D_k.

Step3. Repeat Step2 *K* times and we will get *K* classifiers.

Step4. When comes a test sample, the final classification decision is base on the vote of above gained classifiers.

XII. BOOSTING

Boosting generates new classifier ensembles by readjusting the weight attached to each instance in a way that new ensemble classifiers will focus on difficult cases. The training set for each ensemble depends on the performance of previous classifier(s). AdaBoost Among all the theoretically provable boosting techniques, the most successful one in practical applications has been AdaBoost due to Freund and Schapire [7]. The explanation of its success comes from two reasons, first its simplicity and seconds a property of AdaBoost that previous boosting algorithms [8] lacked of, namely, adaptively". The algorithm adapts its strategy to the situation being used, which free its user from the difficulty of determining algorithmic parameters.

In AdaBoost each training pattern receives a weight that determines its probability of being selected for a training set for an individual component classifier. If a training pattern is accurately classified, then its chance of being used again in a subsequent component classifier is reduced; conversely, if the pattern is not accurately classified, then its chance of being used again is raised. In this way, AdaBoost "focuses in" on the informative or "difficult" patterns. Specifically, we initialize the weights across the training set to be uniform. On each iteration *k*, we draw a training set at random according to these weights, and then we train component classifier C_k on the pattern selected. Next we increase weights of training patterns misclassified by C_k and decrease weights of the patterns correctly classified by C_k. Patterns chosen according to this new distribution are used to train the next classifier C_{k+1}, and the process is iterated.

XIII. DATA MINING ISSUES AND CHALLENGES

1. **Data collection and data organization**
   What data has been collected and where is it?
   How do I combine legacy systems with current data systems?
2. **Modeling issues and data difficulties**
   Data Preparation, Rare or Unknown Targets, Over Sampling, Undercoverage, Dirty Data, Errors, Missing Values, Dimension Reduction (Variable Selection), Under and Over Fitting, Temporal Infidelity, Model Evaluation.
3. **Data integrity:** Another issue is that of data integrity. Clearly, data analysis can only be as good as the data that is being analyzed [9].
4. **Interpretation of results:** Currently, data mining output may require experts to correctly interpret the results, which might otherwise be meaningless to the average database user.
5. **Visualization of results:** To easily view and understand the output of data mining algorithms, visualization of the results is helpful.
6. **Noisy data:** Some attribute value might be invalid or incorrect. These values are often corrected before running data mining applications.
7. **Irrelevant data:** Some attributes in the database might not be of interest to the data mining task being developed.
8. **Missing data:** During the pre-processing phase of knowledge discovery in databases (KDD), missing data may be replaced with estimates.
9. **Changing data:** Databases cannot be assumed to be static. However, most data mining algorithms do assume a static database. This requires that the algorithm be completely rerun anytime the database changes.
10. **Distributed data:** The data to be mined is stored in distributed computing environments on heterogeneous platforms. Both for technical and for organizational reasons it is impossible to bring all the data to a centralized place. Consequently, development of algorithms, tools, and services is required that facilitate the mining of distributed data [10].
11. **Massive data**: Development of algorithms for mining large, massive and high-dimensional data sets (out-of-memory, parallel, and distributed algorithms) is needed. Complex data types: Increasingly complex data sources, structures, and types (like natural language text, images, time series, multi-relational and object data types etc.) are emerging. Grid-enabled mining of such data will require the development of new methodologies, algorithms, tools, and grid services.
12. **Data privacy, security, and governance:** Automated data mining in distributed environments raises serious issues in terms of data privacy, security, and governance. Grid-based data mining technology will need to address these issues.

XIV. CONCLUSION

Data Mining brings a lot of benefits to businesses, society, governments as well as individual. However privacy, security and misuse of information are the major problems if they are not addressed and resolved properly. This paper analyzes the behavior and diversity of various methods available for classification and learning purpose along with the major challenges, issues and application have been focused which help in business strategy formulations, decision making and analysis to the business, society and governments. This work also highlighted the ensemble methods available for combining number of individual classifiers with two schemes.

XV. REFERENCES

[1] Savasre A., Omienciski E., and Navathe S.,(1995), An efficient algorithm for mining association rules in large

databases. In the proceeding of 21st international conference on VLDB, pp. 432-444.
[2] Agrawal R, Srikant R, Fast algorithms for mining association rules. In: Proceeding of the 20th VLDB conference, pp 487–499, 1994.
[3] J. Han, H. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In: Proc. Conf. on the Management of Data (SIGMOD'00, Dallas, TX), ACM Press, New York, NY, USA 2000.
[4] Pang-Ning Tan, Michael Steinbach and Vipin Kumar. Introduction to Data Mining, Addison Wesley, 2006
[5] D. Randall Wilson and Tony R. Martinez. Functions. Improved Heterogeneous Distance In Journal of Artificial Intelligence Research (January 1997), pp. 1-34.
[6] L. Breiman, "Bagging predictors," Mach. Learn., vol. 24, pp. 123–140,1996.
[7] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learningand an Application to Boosting" Journal of computer and system sciences 55, pp.119-139, 1997
[8] Y. Freund, "Boosting a weak learning algorithm by majority", Inform. and Comput. 121, No. 2 (September 1995), 256_285; an extended abstract appeared in ``Proceedings of the Third Annual Workshop on Computational Learning Theory, 1990
[9] S. L. Ting, C. C. Shum, S. K. Kwok, A. H. C. Tsang, W. B. Lee, Data Mining in Biomedicine: Current Applications and Further Directions for Research, Software Engineering & Applications, 2009, 2: 150-159.
[10] Sue Walsh, Data Mining, Higher Education Consulting SAS.
.
.