International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Mining Frequent Item Set Using Cluster Approach from Large Uncertain Database

Naveen Sarawgi[1], C. Malathy[2]

[1]Master of Technology in Computer science and engineering, SRM University, Faculty of Engineering and Technology, Katankulthur 603203, Kancheepuram, Tamilnadu, India

[2]Professor, Department of Computer science and Engineering, SRM University, Faculty of Engineering and Technology, Katankulthur 603203, Kancheepuram, Tamilnadu, India

Abstract: ***The data handling in emerging application and technology like sensor systems, location based system and data integration, are often inaccurate and inexact in nature. In this paper we study the extracting of most frequent item set from large size of uncertain database. The main aim of frequent item set mining is to extract useful information and knowledge from uncertain databases. We propose frequent pattern and Fuzzy C-means algorithm. The combination of frequent pattern algorithm and fuzzy c-means algorithm provide fast and accurate mined information.***

Keywords: Frequent pattern algorithm, Fuzzy C-means algorithm, uncertain database.

1. Introduction

Data mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviours, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems.

Data mining tools can answer business questions that traditionally were too time consuming to resolve. Data mining techniques can be implemented rapidly on existing software and hardware platforms to enhance the value of existing information resources and can be integrated with new products and systems as they are brought on-line. Recent advances in technology have enabled many organizations to collect massive amount of data from their businesses. These datasets can be seen as valuable data for their company as they can find unknown knowledge by mining these datasets.

Frequent sets play an essential role in many Data Mining tasks that try to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers and clusters. The mining of association rules is one of the most popular problems of all these. The identification of sets of items, products, symptoms and characteristics, which often occur together in the given database, can be seen as one of the most basic tasks in Data Mining.

The problem of finding frequent patterns from database is called as Frequent Item set Mining (FIM). Frequent item set mining has become a fundamental task in the field of data mining because it has been widely used in many important data mining tasks such as mining associations, correlations etc. Frequent item set mining is used to predicting the buying behavior of the customers.

2. Objective

To implement an efficient mining algorithm which mine the frequent pattern from uncertain database and handles tuples and deletion problem of database and also the mined pattern result should appears in less time.

Scope: The existing system use apriori and u-apriori algorithm that works fine with small size of databases but for mining large size of databases the time complexity is high..

Frequent Pattern and Fuzzy C-means algorithm overcome the issues of existing system. These algorithms rectify the tuples update and deletion problem of evolving databases and also classify the mined pattern results.

3. Literature Review

3.1 Finding Frequent Items in Probabilistic Data

They propose a new definition based on the possible world semantics that has been widely adopted for many query types in uncertain data management, trying to find all the items that are likely to be frequent in a randomly generated possible world. Their approach leads to the study of ranking frequent items based on confidence as well.

Finding likely frequent items in probabilistic data turn out to be much more difficult. First they propose exact algorithms for offline data with either quadratic or cubic time. Next, they design novel sampling-based algorithms for streaming data to find all approximately likely frequent items with theoretically guaranteed high probability and accuracy. Their sampling schemes consume sub-linear memory and exhibit excellent scalability.

The problem in possible word semantics approach is It works well for small size of database but time complexity is high for large size of database and also this algorithm does not support

other uncertain data models for example graphical probabilistic models.

3.2 Probabilistic Frequent Item set Mining in Uncertain Databases.

In this paper, they introduce new probabilistic formulations of frequent item sets based on possible world semantics. In this probabilistic context, an item set X is called frequent if the probability that X occurs in at least minimum support transactions is above a given threshold. This was the first approach addressing this problem under possible world semantics. In consideration of the probabilistic formulations, they present a framework which is able to solve the Probabilistic Frequent Item set Mining (PFIM) problem efficiently.

An extensive experimental evaluation investigates the impact of their proposed techniques and shows that their approach is orders of magnitude faster than straight-forward approach. The Probabilistic Frequent Item set Mining (PFIM) problem is to find item sets in an uncertain transaction database that are (highly) likely to be frequent. This was first paper addressing this problem under possible world semantics.

They presented a framework for efficient probabilistic frequent item set mining. They theoretically and experimentally showed that their proposed dynamic computation technique is able to compute the exact support probability distribution of an item set in linear time with respect to the number of transactions instead of the exponential runtime of a non-dynamic computation. Furthermore, they demonstrated that their probabilistic pruning strategy allows us to prune non-frequent item sets early leading to a large performance gain. In addition, they introduced an iterative item set mining framework which reports the most likely frequent item sets first.

This was the first approach under possible word semantics so there was very concerns about this approach example high running time complexity, high memory resource uses as well.

3.3 A View of Effective Sampling for Frequent Item Set Mining

Sampling is a well established technique to speed up the process of discovering frequent item sets. They proposed mathematical bounds on the sample size required to probabilistically achieve approximately correct results. A particularly appealing feature of these bounds is that they are independent of the database row-cardinality. In this report, they demonstrate through an extensive empirical evaluation that the bounds, although theoretically elegant, are loose by as much as one to two orders of magnitude in practice. They therefore investigate the possibility of obtaining better bounds through prior knowledge of statistics on the datasets. The number of maximal frequent item sets in the data mining result is known in advance. However, even with such a strong assumption, the revised bound turns out to be several multiples of the required sample size and it identifies a reduced sample size that is sufficient to obtain accurate results.

To address this issue, they present VISTA, a voting-based iterative sampling algorithm for accurately discovering frequent item sets, whose sampling overheads are comparable to the ideal sample size for one-shot frequent item set mining. VISTA incrementally mines samples in small batches and uses the presence or absence of a frequent item set in each batch to determine its voting characteristics. The stopping condition is the reaching of a fix point in the identities of frequent item sets that receive a clear majority of the votes across the batches.

The only problem in this approach is that it requires huge amount of computational resources hence it is very expensive methodology.

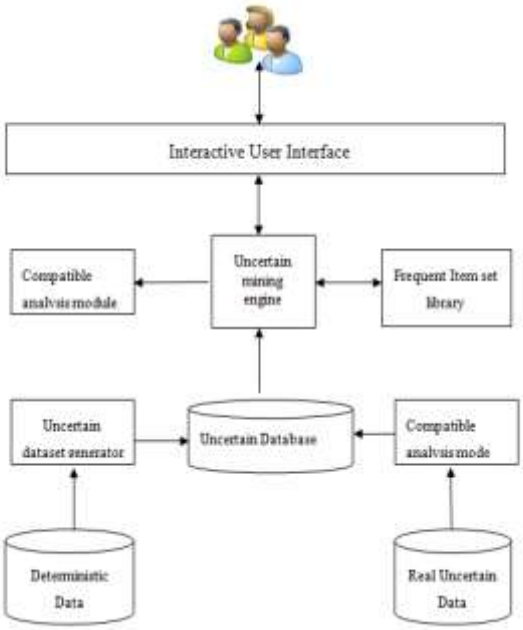

Figure 1: Existing system architecture of mining frequent item set

3.4 Approximate Frequent Item Set Mining in the Presence of Random Noise.

In this paper they propose a noise tolerant item set model, which they call approximate frequent item sets (AFI). Like frequent item sets, the AFI model requires that an item set has a minimum number of supporting transactions. However, the AFI model tolerates a controlled fraction of errors in each item and each supporting transaction.

Motivating this model are theoretical results which state that, in the presence of even low levels of noise, large frequent item sets are broken into fragments of logarithmic size; thus the item sets cannot be recovered by a routine application of frequent item set mining. By contrast, they provide theoretical results showing that the AFI criterion is well suited to recovery of block structures subject to noise.

They developed and implemented an algorithm to mine AFIs that generalizes the level-wise enumeration of frequent item sets by allowing noise.

They propose the noise-tolerant support threshold, a relaxed version of support, which varies with the length of the item

set and the noise threshold. They exhibit an Apriori property that permits the pruning of an item set if any of its sub-item set is not sufficiently supported. Several experiments presented demonstrate that the AFI algorithm enables better recoverability of frequent patterns under noisy conditions than existing frequent item set mining approaches. Noise-tolerant support pruning also renders an order of magnitude performance gain over existing methods.

Approximate frequent item set mining algorithm generalize frequent item set in level wise only and it does not support depth wise level mining of the uncertain database.

3.5 Mining association rules between set of items in large database.

They purposed an efficient algorithm for large size of customer transaction database that generates all significant association rules between items in the database. The algorithm incorporates buffer management and novel estimation and pruning techniques. They also presented results of applying this algorithm to sales data obtained from a large retailing company, which shows the effectiveness of the algorithm.

They introduced the problem of mining association rules between sets of items in a large database of customer transactions. Each transaction consists of items purchased by a customer in a visit. Having obtained the large item sets and their transactional support count, the solution to the second sub problem is rather straightforward. A simple solution to the first sub problem is to form all item sets and obtain their support in one pass over the data.

They presented the algorithm that works fine with the transactional database but if the database is continually evolving or database is not certain than this approach does not work.

4. Proposed Solution

Mining Frequent Item set using Cluster Approach from Large Uncertain Database

The combination of efficient FP-growth and fuzzy C-Means algorithm used that is used to mine frequent item set from large size of uncertain database. The FP-Growth algorithm is an efficient and scalable method for mining the complete set of frequent patterns by pattern fragment growth, using an extended prefix-tree structure for storing compressed and crucial information about frequent patterns named frequent-pattern tree (FP-tree).

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

Data clustering is the process of dividing data elements into classes or clusters so that items in the same class are as similar as possible, and items in different classes are as dissimilar as possible. Depending on the nature of the data and the purpose for which clustering is being used, different measures of similarity may be used to place items into classes, where the similarity measure controls how the clusters are formed.

In fuzzy clustering (also referred to as soft clustering), data elements can belong to more than one cluster, and associated with each element is a set of membership levels. These indicate the strength of the association between that data element and a particular cluster. Fuzzy clustering is a process of assigning these membership levels, and then using them to assign data elements to one or more clusters.

FP Growth Algorithm

The algorithm works as follows: first it compresses the input database creating an FP-tree instance to represent frequent items. After this first step it divides the compressed database into a set of conditional databases, each one associated with one frequent pattern. Finally, each such database is mined separately. Using this strategy, the FP-Growth reduces the search costs looking for short patterns recursively and then concatenating them in the long frequent patterns, offering good selectivity.

The algorithm requires two phases:

1. Build a compact data structure called the FP-tree.
2. Extracts frequent item sets directly from the FP-tree.

Phase Description:

1. Build a compact data structure called the FP-tree:
FP-Tree is constructed using 2 passes over the dataset

Pass 1:
1. Scan data and find support for each item.
2. Discard infrequent items.
3. Sort frequent items in decreasing order based on their support.

Use this order when building the FP-Tree, so common prefixes can be shared.

Pass 2:
1. FP-Growth reads 1 transaction at a time and maps it to a path.
2. Fixed order is used, so paths can overlap when transactions share items (when they have the same prefix). In this case, counters are increment.
3. Pointers are maintained between nodes containing the same item, creating singly linked lists (dotted lines).The more paths that overlap, the higher the compression. FP-tree may fit in memory.
4. Frequent item sets extracted from the FP-Tree.

Fuzzy C-Mean Algorithm:

Fuzzy C-Means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters.

Algorithm:

1. Initialize U = U_{ij} matrix, U (0)

2. At k-step: calculate the centers vectors C (k) = [c j] with U (k)

$$C_j = \frac{\sum_{i=1}^{N} U_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} U_{ij}^{m}}$$

3. Update U (k), U (k+1)

$$U_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{x_i - c_j}{x_i - c_k} \right)^{\frac{2}{(m-1)}}}$$

4. If $\| U^{(k+1)} - U^{(k)} \| < \in$ then STOP; otherwise return to step2.

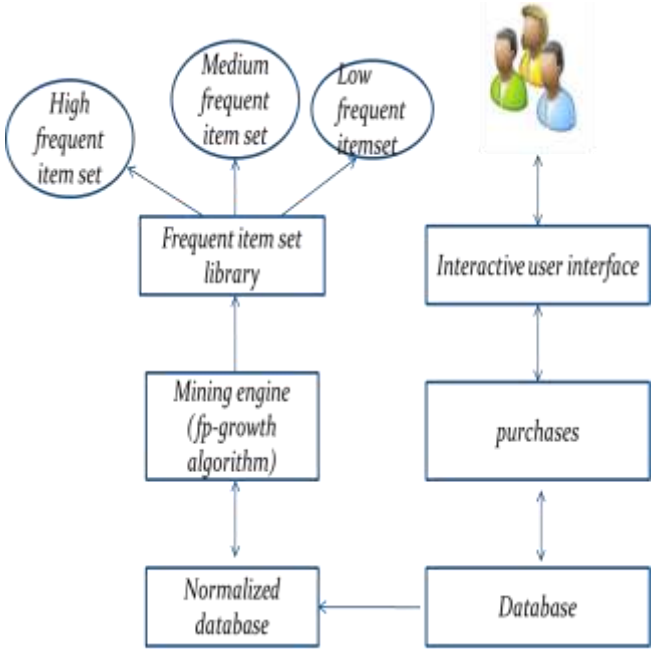

Figure 2: Mining Frequent Item Set System Architecture

The Figure 1: Mining Frequent Item Set System Architecture provides the overview of mining frequent item set from large uncertain database.

Interactive user interface module gets command from the user and executes the command on server site. All the transaction information of the customers stores into the database.

To handle the uncertain data normalization performed. Normalization removes noisy data and makes the database into proper structured format. In this project BCNF normalization technique used.

Mining engine apply the FP-growth and fuzzy C-Means algorithm on to the normalized database and store the resultant frequent item set into frequent item set library.

Frequent item set fetch the frequent item set from mining engine and store the frequent item set into 3 categories:

1. Most frequent item set.
2. Medium frequent Item set.
3. Low frequent item set.

The most frequent item automatically adds to the customer shopping cart and medium frequent and low frequent item set comes to customer display as a suggestion to buy.

Modules Description: These are the following modules that includes in the proposed architecture of mining frequent item set of the uncertain database using cluster approach.

Interactive User Interface:

The user interface is the space where interactions between humans and machines occur. The modules gets the command from users and perform the various operation like getting the data from customers, sending the customers data to server and showing the server's data to customer display screen, etc. The goal of the interaction is effective operation and control of the machine on the user's end, and feedback from the machine, which aids the operator in making operational decisions.

User interface modules consists of two kinds of plug-in

1. Web Item modules define links that are to be displayed in the UI at a particular location.
2. Web Section modules define a collection of links to be displayed together, in a section.

Web items and web sections (referred to collectively as 'web fragments') may be displayed in a number of different ways, depending on the location of the fragment and the theme under which it is being displayed.

Normalization of Database:

Database normalization is the process of organizing the fields and tables of a relational database to minimize redundancy. Normalization usually involves dividing large tables into smaller (and less redundant) tables and defining relationships between them.

The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database using the defined relationships.

When a fully normalized database structure is extended to allow it to accommodate new types of data, the pre-existing aspects of the database structure can remain largely or entirely unchanged. As a result, applications interacting with the database are minimally affected. Normalized tables are suitable for general-purpose querying. This means any queries against these tables, including future queries whose details cannot be anticipated, are supported.

BCNF technique used to perform normalization. Normalized database provide more accurate result. Mining algorithm

engine work faster on normalized data so time consumption to mine data become less.

Mining Engine

Data mining engine is the heart (or actually the brains) of our platform. It's where most data crunching occurs and it is where the key values that affect all platform components are calculated.

Mining engine module performs mining algorithm on the normalized database. It extracts the task relevant data to mine and then perform mining operations. Results will be in the form of graph, bar etc.

Data mining engine performs FP growth and Fuzzy c-means algorithm simultaneously on the normalize database. Initially FP growth algorithm applies on the database and delivers frequent item sets. To clustering the frequent item set, clustering algorithm apply to mined frequent set, hence frequent item set cluster into 3 categories;

1. Most frequent item set.
2. Medium frequent item set.
3. Low frequent item set.

Frequent Item Set library

Frequent item set library contains the resultant operation of mining engine that is frequent item set. Frequent item set library will store frequent data in the form of, Most frequent item set, Medium frequent item set, low frequent item set. There is interconnectivity between mining engine and frequent item set library. Frequent item set library fetch the mined frequent item set from the mining engine. If the frequency of the item changes then frequent item set library refresh by new item set.

Suggest Classified Frequent Item Set

In the module, customer logins to shopping cart next time, the most frequent item will be automatically added to his cart. The medium and low frequent item set will come as a suggested item to buy on the customer display screen.

It helps the customers to save time while shopping. Customers need not to search items on the shopping website, the user interest items will be automatically adds to the shopping cart.

5. Conclusion

We have implemented frequent pattern growth (FP-growth) and fuzzy C-means algorithm to mine frequent item set from uncertain database. Frequent pattern growth algorithm for storing compressing, crucial information about frequent patterns. Fuzzy C-means classify the search result.

There are several advantages of FP-growth and Fuzzy C-means algorithm over other approaches:

1. It constructs a highly compact frequent pattern tree, which is usually substantially smaller than the original database, and thus saves the costly database scans in the subsequent mining processes.
2. It applies a pattern growth method which avoids costly candidate generation and test by successively concatenating frequent 1-itemset found in the (conditional) FP trees. In this context, mining is not Apriori-like (restricted) generation-and-test but frequent pattern (fragment) growth only.
3. It applies a partitioning-based divide and conquers method which dramatically reduces the size of the subsequent conditional pattern bases and conditional FP-trees.
4. It solves the tuples updates and deletion problem of evolving database.
5. It classifies the mined research result in the form of cluster. Frequent pattern clusters are most frequent item set, medium frequent item set and less frequent item set.

References

[1] A.Veloso, W.Meira Jr., M. de Carvalho, B. Possas, S.Parthasarathy, and M.J. Zaki, "Mining Frequent Item sets in Evolving Databases," Proceeding Second SIAM International Conference Data Mining (SDM), 2002.
[2] Albrecht Zimmermann, KU Leuven, "Objectively evaluating interestingness measures for frequent item set mining," Proceeding ACM SIGMOD International Conference Management of Data, 2009.
[3] C.K. Chui, B. Kao, and E. Hung, "Mining Frequent Item sets from Uncertain Data," Proceeding 11th Pacific-Asia Conference Advances in Knowledge Discovery and Data Mining (PAKDD), 2007
[4] C. Aggarwal and P. Yu, "A Survey of Uncertain Data Algorithms and Applications," IEEE Transaction Knowledge and Data Eng., vol. 21, no. 5, pp. 609-623, May 2009.
[5] C. Aggarwal, Y. Li, J. Wang, and J. Wang, "Frequent Pattern Mining with Uncertain Data," Proceeding 15th ACM SIGKDD International Conference Knowledge Discovery and Data Mining (KDD), 2009.
[6] D. Cheung, J. Han, V. Ng, and C. Wong, "Maintenance of Discovered Association Rules in Large Databases," Proceeding ACM SIGMOD International Conference Management of Data, 2007.
[7] H. Cheng, P. Yu, and J. Han, "Approximate Frequent Item set Mining in the Presence of Random Noise," Proceeding Soft Computing for Knowledge Discovery and Data Mining, pp. 363-389, 2008.
[8] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," Proceeding ACM SIGMOD International Conference Management of Data, 2000.
[9] L. Sun, R. Cheng, D.W. Cheung, and J. Cheng, "Mining Uncertain Data with Probabilistic Guarantees," Proceeding 16th ACM SIGKDD International Conference Knowledge Discovery and Data Mining, 2010.
[10] Mudireddy, Jagadesh Babu, "A View of Effective Sampling for Frequent Item set Mining," Proceeding

ACM SIGMOD International Conference Management of Data, 2000.
[11] Q. Zhang, F. Li, and K. Yi, "Finding Frequent Items in Probabilistic Data," Proceeding ACM SIGMOD International Conference, 2008.
[12] R. Agrawal, T. Imieli_nski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," Proceeding ACM SIGMOD International Conference, 1993.
[13] T. Bernecker, H. Kriegel, M. Renz, F. Verhein, and A. Zuefle, "Probabilistic Frequent Item set Mining in Uncertain Databases," Proceeding 15th ACM SIGKDD International conference. Knowledge Discovery and Data Mining (KDD), 2009.

Author Profile

Naveen Sarawgi Pursuing M.Tech degrees in Computer Science and engineering from SRM University, Chennai, Under the Guidance of **Prof. C. Malathy** (Dept. of computer science and engineering) SRM University, Chennai**.**