# Hesitation Mining: A Survey Approach

**Arpita Agarwal[1], Sujeet Singh Bhadouria[2]**
[1, 2] Department of CSE
[1, 2] NITM, Gwalior

***Abstract-*** *The traditional Apriori algorithm has a problem that it deals only with the items that are sold and that is not sold. In this survey, we focused on hesitation information about items. Hesitation information of items is precious knowledge for the design of good selling strategies. We also present vague set theory which is capable of handling hesitation information of items.*

***Keywords-*** frequent itemset, data mining, fuzzy association rule mining, minimum support.

## I. INTRODUCTION

Data mining is a process of extracting interesting knowledge or patterns from large databases. There are several techniques that have been used to discover such kind of knowledge, most of them resulting from machine learning and statistics. The greater part of these approaches focus on the discovery of accurate knowledge [1]. Though this knowledge may be useless if it does not offer some kind of surprisingness to the end user. The tasks performed in the data mining depend on what sort of knowledge someone needs to mine.

Data mining field has wide scope for research. The main area for research is data warehouse, association rule mining, classification, prediction, clustering, etc. For mining frequent patterns and generating rule from that is useful in large application . The term Data Reduction in the context of data mining is usually applied to projects where the goal is to aggregate or amalgamate the information contained in large datasets into manageable (smaller) information. Data reduction methods can include simple tabulation, aggregation (computing descriptive statistics) or more sophisticated techniques like clustering, principal components analysis, etc.

Data preparation and cleaning is an often neglected but extremely important step in the data mining process. The old saying "garbage-in-garbage-out" is particularly applicable to the typical data mining projects where large data sets collected via some automatic methods (e.g., via the Web) serve as the input into the analysis. Often, the method by which the data where gathered was not tightly controlled, and so the data may contain out-of-range values (e.g., Income: -100), impossible data combinations (e.g., Gender: Male, Pregnant: Yes), and the like. Analyzing data that has not been carefully screened for such problems can produce highly misleading results, in particular in predictive data mining.

## II. ASSOCIATION RULE MINING

Association rules discovery is one of the most important technologies which was given by Mr. Agrawal in 1993. It gives the information like "if-then" statements. These rules are invoked from the dataset[2]. It generates from calculation of the support and confidence of each rule that can show the frequency of occurrence of a given rule.

An association rule, a well researched method for discovering interesting relations between the items in large databases, is popular and effective for decision – making in financial mathematics, medical, intrusion detection and web analysis in recent years. In association rule mining, frequent patterns are patterns (eg. itemsets, subsequences, or substructures) that appear frequently in the dataset. For example, a set of items such as milk and bread that appear frequently together in a transaction dataset is a frequent itemset. The occurrence frequency of an itemset is the number of transactions that contain the itemset. This is also known, simply, as the frequency, support count, or count of the itemset.

## III. NOTATION OF ASSOCIATION RULE MINING

### A. Support

Let A and B are the items in the database. Support of the rule $A \Rightarrow B$: denotes the frequency of the rule within all transactions in the database, i.e., the transaction that contains both A and B.

$$\text{support}(A \Rightarrow B) = \frac{\text{No. of transactions containing both A and B}}{\text{Total no. of transactions}}$$

### B. Confidence

Confidence is the measure of how often items in A appear in transactions that contain B. Strength of implication in the rule is denoted by confidence. Confidence 'c' in the transaction set D, where c is the percentage of transaction in D containing A that also contain B. This is taken to be the conditional probability, P(B|A).

$$\text{confidence}(A \Rightarrow B) = \frac{\text{No. of transactions containing both A and B}}{\text{No. of transactions containing A}}$$

## IV. HESITATION MINING USING VAGUE SET THEORY

The hesitation information of an item describes using following two scenarios:

A customer wants to buy a car but hesitate to buy it because of some reasons like he cannot afford the price of a car. If seller offered him loan for a car then his hesitation percentage decreases, but if seller is ready to provide a loan according to that customer, then his hesitation status removes and comes in to the category of favour and buy it.

In the real world there are vaguely specified data values[7] in many applications, such as sensor information. Fuzzy set principle has been proposed to address such vagueness with the aid of generalizing the notion of membership in a set. Essentially, in a Fuzzy Set (FS) every detail of an element is associated with a point-value selected from the unit interval [0,1], which is called the grade of membership in the set. A vague Set (VS), as well as an Intuitionistic Fuzzy Set (IFS), is a in addition generalization of an FS. Rather than using point-based membership as in FSs, interval-based membership is used in a vague sets (VS). The interval-based membership value in VSs is greater expressive in capturing vagueness of records than point – based membership values. Fuzzy set idea has lengthy been introduced to deal with inexact and vague facts by using Zadeh's seminal paper in [4], seeing that within the actual international there's vague records approximately distinctive applications, together with in sensor databases, we will formalize the measurements from extraordinary sensors to a vague set. In fuzzy set concept, every item $u \in U$ is assigned a single actual price, known as the grade of membership, among zero and one. (here U is a classical set of items, called the universe of discourse.). In [5], Gau et al. factor out that the disadvantage of using point-based value in fuzzy set concept is that the proof for $u \in U$ and the proof against $u \in U$ are in reality combined together. With the intention to tackle this problem, Gau et al. propose the perception of vague sets (VSs), which allow the use of interval-based membership instead of the use of point-based values as in FSs. The interval-based club generalization in VSs is extra expressive in capturing vagueness of facts. For that reason, the thrilling functions for handling indistinct information which are unique to VSs are largely left out.

## V. VARIOUS ALGORITHMS ON ASSOCIATION RULE MINING

### A. Traditional Apriori algorithm

In traditional Apriori algorithm, first, the set of frequent1- itemsets is found by scanning the database to accumulate count for every item, and collection of those items that satisfy user specified minimum support(min_sup). The resulting set obtained is denoted by L1. Next, L1 is used to find L2, frequent2-itemsets, which is then used to find L3 frequent3-itemset, and so on until no more frequent-itemsets can be found. The process of finding of each $L_k$ requires full scan of the database once.

Traditional Apriori algorithm works on Boolean logic. Boolean association rule mining uses the concept of crisp sets. Because of this reason Boolean association rule mining has several drawbacks. It works on only yes and no form. It is inappropriate to handle imprecise and inexact data. Because in real life domain, there are various vague situations or hesitation information. To overcome those drawbacks the concept of fuzzy association rule mining came. For example, when we go to mall for shopping, then we put items in our basket, but sometimes at the time of transactions, our mind is changed suddenly and we walkout for some items. This type of situation is not handled by crisp sets, but by applying fuzzy theory, we can somewhat handle those situations.

**Advantages of Apriori Algorithm**

1. Uses property of large item set.
2. Parallelized easily
3. Implement is easy.
4. The Apriori algorithm implements level-wise search using frequent item property

**Drawbacks of traditional Apriori algorithm-**

We can easily see by the analysis, traditional Apriori algorithm has following bottlenecks [3]:

(1) Scans the large database repeatedly to produce $L_k$ has reduced efficiency of the algorithm. Items in the candidate itemsets must scan database one time to decide whether it can be joined to the $L_k$ frequent itemset. So it needs to scan the transaction database as the same number as the elements of the frequent itemset.

(2) When it goes on the k-th scanning, the algorithm does not use the previous results. On the other hand, the key of the

improved Apriori algorithms is to reduce the number of scans.

(3) Generation of candidate itemsets is expensive (in both space and time)

(4) Support counting is expensive

To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property is used to reduce the search space.

**APRIORI PROPERTY** – All non-empty subsets of a frequent itemsets must also be frequent.

**B. Improved Apriori Algorithm**

Let Tid-set(C) denote the set of transaction Tid which has item C in D, then the amount of transaction which contain C in D is the amount of item of Tid-set(C), sup_count(C) can be computed by |Tid-set(C)|. The transaction set in D, which have C and B, is the intersection of Tid-set(C) and Tid-set(B), and sup-count(C□B) can be computed as |Tid-set(C)∩TID-set(B)|.

The join and the pruning of Apriori algorithm is improved correspondingly: all non-empty subset of frequent itemset must be frequent; to produces $L_k$-candidates[4], we only join the set whose k-2 item is same in $L_k$-1. So following improvement to the traditional Apriori algorithm exists :

(1) Minimum sup-count computed by min-sup*|D|.

(2) Scan the database once to produce L1-candidates, simultaneously construct Tid-set(Z1) for each item. After scan, compute minsup-count for each item and further find the set of frequent items L1.

(3) L2-candidates can be produced from L1*L1.Scanning L1, we can find Tid-set(Z2) and sup-count of each itemset, deleted the patterns whose frequencies don't satisfy the min support count, and then further find L2.

(4) To produce Lk(k≥3), join itemsets which satisfy the join rule. For Scanning (Lk-1), we can find Tid-set(Zk) and perform sup-count of each itemset and then deletes the patterns whose frequencies does not under user specified support count, and finally get Lk.

**C. Fuzzy association rule mining**

Fuzzy association rule mining uses fuzzy logic or fuzzy theory to generate interesting association rules which works on inexact and imprecise data. These association rules can help in decision making for various hesitation information. Fuzzy rules is a variant of classical association rule mining. Classical association rule mining uses the concept of crisp sets which we discussed above.

Fuzzy logic[5] is introduced to handle imprecise and inexact data , since in the real world, there is a vague information or knowledge about different applications such as in sensor databases. Fuzzy set uses point-value selected from the unit interval [0,1] which is the limitations in the sets to tackle with some typical vague data. Sometimes, there is a even more vague situations came in the real life domain where fuzzy rules are also inappropriate to examine greater results. Thus, there is a wide scope of improving concepts of association rule mining in the future work. As data mining is the most important concept in real life applications like in organizations, shops, etc, there scope has been widely increased.

**D. FP-Growth algorithm**

FP-growth finding frequent itemsets without candidate generation[14]. The first scan of the database is the same as Apriori, which derives the set of frequent items(1-itemsets) and their support counts(frequencies). The FP-tree is mined as follows. Start from each frequent length-1 pattern(as an initial suffix pattern), construct its conditional pattern base (a "sub-database" which consists of theprefix paths in the FP-tree co-occuring with the suffix pattern), then construct its(conditional) FP-tree, and perform mining recursively on the tree. The pattern growth is achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree.FP-tree is also not a fuzzy representation of the database.The FP-growth method described in[6] transforms the problem of finding long frequent patterns to searching for shorter ones recursively and then concatenating the suffix. It uses the least frequent items as a suffix, offering good selectivity. The method substantially reduces the search costs.

When the database is large, it is sometimes unrealistic to construct a main memory based FP-tree. An interesting alternative is to first partition the database into a set of projected databases, and then construct an FP-tree and mine it in each projected database. Such a process can be recursively applied to any projected database if its FP-tree still cannot fit in main memory. A study on the performance of the FP-growth method shows that it is efficient and scalable for mining both long and short frequent patterns, and is about an order of magnitude faster than the Apriori algorithm. It is also faster than a Tree-Projection algorithm, which recursively projects a database into a tree of projected databases.

**Algorithm:** FP growth. Mine frequent itemsets using an FP-tree by pattern fragment growth.

Input: *D*, a transaction database;
*min sup*, the minimum support count threshold.
Output: The complete set of frequent patterns.

(a) Method: 1. The FP-tree is constructed in the following steps: (a) Scan the transaction database *D* once. Collect *F*, the set of frequent items, and their support counts. Sort *F* in support count descending order as *L*, the *list* of frequent items.
(b) Create the root of an FP-tree, and label it as "null." For each transaction *Trans* in *D* do the following.

Select and sort the frequent items in *Trans* according to the order of *L*. Let the sorted frequent item list in *Trans* be [*p*j*P*], where *p* is the first element and *P* is the remaining list. Call insert tree ([*p*j*P*], *T*), which is performed as follows. If *T* has a child *N* such that *N.item-name=p.item-name*, then increment *N*'s count by 1; else create a new node *N*, and let its count be 1, its parent link be linked to *T*, and its node-link to the nodes with the same *item-name* via the node-link structure. If *P* is nonempty, call insert tree(*P*, *N*) recursively.
2. The FP-tree is mined by calling FP growth(*FP tree*, *null*), which is implemented as follows.
procedure FP growth(*Tree*, a)

(1) If *Tree* contains a single path *P* then
*(2)* For each combination (denoted as b) of the nodes in the path *P*
(3) Generate pattern b[a with *support count = minimum support count of nodes in* b;
*(4)* Else for each *ai* in the header of *Tree {*
(5) Generate pattern b = *ai* [a with *support count = ai*: *support count*;
(6) Construct b's conditional pattern base and then b's conditional FP tree *Tree*b;
(7) if *Tree*b 6= /0 then
(8) call FP growth(*Tree* s, □); }

**E. Mining Frequent Itemsets using vertical data format**

Both the Apriori and FP-growth methods mine frequent patterns from a set of transactions in TID-itemset format, where TID is a transaction ID and itemset is the set of items bought in transaction TID[2]. This is known as the horizontal data format. Alternatively, data can be presented in item-TID_set format(i.e.,{item: TID_set}), where item is an item name, and TID_set is the set of transaction identifiers containing the item. This is known as the vertical data format.

**REFERENCES**

[1] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From Data Mining toKnowledge Discovery in Database [J]. American Association for Artificial Intelligence, AAAI Press,July,1996, 37-54.

[2] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases", Proc. of the 20th International Conference on Very Large Data Bases, VLDB, Page(s): 487-499, 1994.

[3] W.W. Chen. Data Warehouse and Data Mining[M]. BeiJing: Tsinghua University Press 2006.

[4] Department of Department of Computer Science, Shandong Institute of Business and Technology, Shandong, 264005, China

[5] Ashish Mangalampalli, Vikram Pudi: Fuzzy Association Rule Mining Algorithm for Fast and Efficient Performance on Very Large Datasets. FUZZ-IEEE 2009, Korea, ISSN: 1098-7584, E-ISBN: 978-1-4244-3597-5, Page(s): 1163 – 1168, August 20-24, 2009.

[6] Han and M. Kamber, Data Mining: Concepts and Techniques, San Francisco: Morgan Kaufmann Publishers,2001.

[7] A. Lu and Wilfred Ng, "Vague Sets or Intuitionistic Fuzzy Sets for Handling Vague Data- Which One Is Better", 2005.