Efficient Mining of Frequent Patterns From Uncertain Databases using Hierarchical Agglomerative Clustering

Sujeena S  
Student  
Department of Computer Science and Engineering  
Nehru College of Engineering and Research Centre  
Thrissur, India

Mary Mareena P V  
Assistant Professor  
Department of Computer Science and Engineering  
Nehru College of Engineering and Research Centre  
Thrissur, India

Abstract

In many real time applications such as sensors monitoring systems, location based systems etc data uncertainty is inherent. The uncertainty may occur as a result of evaluation errors. This uncertainty becomes a major issue while performing mining operations in the databases. Traditional mining methods are less efficient when dealing with uncertain databases. While considering uncertain databases one of the major issues is to mine frequent item sets from database. Here some algorithms which are used to perform frequent pattern mining in uncertain databases are provided. These algorithms discussed here are the extension of the frequent pattern mining algorithms which are used in deterministic databases where the values are precise. These algorithms are modified in such a way to handle uncertainty in the database. Among the various algorithms the U-prefixspan algorithm an extension of prefix span algorithm gives better performance. Clustering is used in this algorithm to further increase the efficiency.

Keywords: uncertain database; frequent sequential patterns; PWS; hierarchical agglomerative clustering

I. INTRODUCTION

In recent years the data collection is done using different methodologies. Sometimes the collected data may contain error or may be partially complete. Such databases are complex because the difficulty or challenge in representing probabilistic information. For example in sensor networks the data collected may contain large amount of uncertain datasets. That means that the data points which correspond to a particular object may contain some vagueness. These types of situations result in the need for uncertain data management approaches. The uncertain data is represented using probabilistic distributions whereas the traditional databases contain deterministic values. There are mainly two types of uncertainty they are tuple level and attribute level uncertainty. Tuple level uncertainty is the one in which the existence of the particular tuple in the particular database is uncertain. Attribute level uncertainty deals with the uncertainty in the existence of the particular attribute. Due to these types of uncertainty the important problem in handling uncertain database is possible world explosion. The possible World Semantics (PWS) is nothing but, if a particular tuple is facing attribute level uncertainty then that tuple will be repeated for all the possible attributes hence the size of the database increases exponentially. This problem is efficiently handled in U-prefixspan algorithm where the attribute with highest existential probabilities are only considered for possible world semantics. There are number of unique challenges when an uncertain database is considered. The main areas of research in this field are:

A. Modeling of Uncertain Data

It is a major issue whenever an uncertain database is considered. The representation of the uncertain data in the database is the problem here. The data can be represented using probability values. This gives the importance of probabilistic databases. In probabilistic databases is the one in which the possible instance will be associated with a probability value. Considering all possible instances will result in exponentially large number of instances. Hence a simplified model should be used for representing the data so that the mining and management can be done easily. Hence probabilistic tables are used in which the probability of a data or tuple present in database is calculated as the product of probabilities of the tuple present in the database to the probabilities of the tuple to be absent in the database.

B. Uncertain Data Management

Traditional data management applications such as query processing, joint processing, indexing, selective estimation etc can be used in uncertain data management. Some modifications are needed because the data deals with uncertainty that means probability values. There are mainly two semantic approaches used they are intentional and extensional semantics. In case of intentional semantics the uncertain database is represented in terms of event model that defines the possible worlds. And it uses a
tree like structures of inferences on the possible event combinations. Extensional semantics attempts to design a plan which can approximate the queries and without having to enumerate the entire tree for inferences.

C. Uncertain Data Mining
Data mining in uncertain database is a crucial task because of the underlying uncertainty in data. Traditional data mining applications are less efficient in case of uncertain databases. Each item is associated with an existential probability. An item is said to be frequent if the expected support of the item is at least equal the user specified threshold. There are a number of techniques available to handle this critical situation. A number traditional frequent pattern mining algorithms are extended to handle uncertain data. The expected support of a set of items in the transaction is obtained by simply multiplying the probabilities of the different items in the transaction. Data trimming or pruning can be done in order to increase the efficiency of frequent pattern mining. In data pruning the data with low existential probabilities are removed from further considerations. The algorithm is applied again in the remaining data.

II. RELATED WORKS

Some of the frequent pattern mining algorithms for uncertain databases are as shown below. These algorithms are an extension of the frequent pattern mining algorithms which are used in deterministic databases. Various frequent pattern mining algorithms are as follows:

A. U-Apriori Algorithm
It is an extension of Apriori algorithm [5]. The algorithm performs some step by step operations. Initially it finds all the expected support based items. Then it repeatedly joins all expected support based frequent i-itemsets to produce i+1 itemset candidates and test i+1-itemset candidates to obtain expected support based frequent i+1 itemsets. Finally it ends with no expected support based frequent i+1 itemsets are generated. U-Apriori algorithm [3] use the same steps of candidate generation and candidate pruning steps as used in Apriori algorithms. The major disadvantage regarding this approach is that multiple scans needed to be performed in the database so that each time the candidate generation phases to be completed. But it increases the execution time.

B. UF-Growth Algorithm
It is a tree based approach [2]. It is mainly suggested in order to avoid the overhead of candidate generation which is a major issue in case of Apriori algorithm. A tree like structure can be used efficiently used to represent the uncertainty in data. The algorithm mainly consists of mainly two operations they are construction of UF-trees and mining of frequent patterns from UF-trees. Each node in the tree contains the following information they are an item, its expected support, the number of occurrences of such expected support for such an item. After the construction of UF-tree the UF-growth algorithm recursively mines frequent patterns from the UF-tree. Even though the problem of candidate generation is solved large storage is required in order to store the tree structure as the size of the database grows.

C. Prefix-Projected Sequential Pattern Mining Algorithm
It is a prefix-projected sequential pattern growth algorithm [4] which can be effectively used to mine frequent patterns from deterministic or traditional databases. This also includes some step by step operations. At first the algorithm finds the frequent single items. And then generate a set of projected database for each of the frequent items. This projected database is mined recursively while concatenating each single frequent item into a frequent sequence. Each recursion results in reduction in the size of projected database. Hence reduces the execution time. It works well for mining large frequent sequences.

D. U-Prefixspan Algorithm
In case of probabilistic databases the presence of a particular pattern in a probabilistic sequence is represented in terms of presence probability [1]. The presence probability of a pattern \( \alpha \) in a probabilistic sequence \( s \) is given as

\[
Pr(\alpha \subseteq s) = \sum_{\alpha \subseteq s} Pr(pwi)
\]  

(1)

Where, \( s \) is a deterministic instance of probabilistic sequence \( s \) in the possible word \( pwi \). \( Pr(pwi) \) represents the existence probability of possible word \( pwi \). The expected support is used in probabilistic databases in order to find the frequency of pattern. The expected support of the pattern \( \alpha \) is denoted by expSup(\( \alpha \)). It is defined as the sum of the expected probabilities of the presence of \( \alpha \) in each of the sequences in the databases. The probabilistic frequency of a pattern can be defined as, given a probability threshold \( t_{\text{prob}} \) and a support threshold \( t_{\text{sup}} \) a pattern is said to be probabilistic frequent if and only if the particular pattern have minimum value of \( t_{\text{prob}} \) as well as \( t_{\text{sup}} \). In some old techniques only threshold support value is checked but it is not enough in case of pattern mining in probabilistic database. While considering threshold support alone some important patterns may get missed or some unwanted patterns may get retrieved. To avoid this problem U-prefixspan algorithm uses both probability and support threshold values.

This algorithm is an extension of prefixspan algorithm called sequence-level U- Prefixspan algorithm [1]. Here the algorithm is processed in three different sections. The sections include creating a pattern projected database and then the probabilistic frequency check and pruning process are performed.
1) Sequence-Level U-prefixspan
Given a sequence level probabilistic sequence and a pattern α then a projected probabilistic sequence can be obtained using this algorithm [1]. The si|α is obtained by projecting each deterministic sequence instance sij of sequence si (denoted by sij ∈ si) onto sij|α, excluding those instances that cannot be projected due to α is not a subset of sij. In order to achieve high space utility the sij|α is not stored as a suffix sequence of sij. It is sufficient to represent sij|α with a pointer to sij and the starting position of suffix sij|α in sij. In this algorithm each projected sequence instance is represented as a pair of <pos, sij> where pos represents the position before the staring position of suffix sij|α in sij. Each sij instance is represented as (sij|α, pr (sij)). It means a sequence and its existential probability value. If the probability value is close to 1 or 1 means that the accuracy of the sequence is high.

III. HIERARCHICAL AGGLOMERATIVE CLUSTERING

The advantages of sequence level U-prefixspan algorithm can be further increased by including the concept of clustering. The Hierarchical clustering algorithms are mainly done in two ways. They are either top-down or bottom-up. The first one that is the top-down approach is a divisive approach where the documents or groups are divided until individual single clusters are formed. But the later one that is the bottom-up approach is a reverse process where the individual document or element is combined to form a complete document or final single group. Using the hierarchical agglomerative clustering the efficiency of frequent pattern mining can be increased to greater extend. Mining multiple probabilistic frequent patterns is the main concept.

For a given input pattern the possible combinations are generated using the hierarchical agglomerative clustering. Initial the frequenctness of the input pattern is checked based on the threshold support and probability values. Then the input pattern is split into individual elements. Then keeping the first element in its actual position the other positions of other elements are changed. These combinations will be updated into the database. Then the element in the second position of the input pattern is brought to the first position and the other elements are arranged in different combinations. This procedure is continued until all the possible combinations of the given input pattern is generated. These possible combinations are matched in the database for frequent pattern.

The sequences which match the patterns are grouped into single clusters. Sometimes same sequence may appear in more than one cluster. From each cluster the count is made. There is a possibility that same pattern may be found more than one times in the same cluster in the same sequence. Hence these repetitions should be ignored that is the count of each repeated sequence must be taken once. Based on the count of the patterns found and the threshold support the frequent patterns can be found. The main advantage is that a number of patterns can be found once a single input pattern is given.

A. Finding the Pattern
The input pattern is verified against the sequence database. If any sequence has a match with a given pattern then this particular sequence will be included in the projected database. Further the matching is continued. Here in case of uncertain database the sequence will include a probability value associated with them. Hence the sequences which have a greater probability value than the given threshold probability are only considered for the projected database generation. The projected database is used further for the process of pattern growth. The projected database includes the sequence id, the sequence, the probability value and also the position where the particular pattern is found in the sequence. This is very useful for further growing the pattern.

B. Pattern Growth
The pattern growth is a major advantage as far as prefixspan algorithms are considred. In case of old pattern mining techniques the pattern growth is a complicate task because the whole that is original database is scanned for finding new patterns or growing the old pattern. This is not good if the original database is large. Hence the prefixspan algorithm is considered good in such situations. The Uprefixspan algorithm also uses the concept of pattern growth. If a new pattern which is generated by appending some elements to the previously verified pattern is to be verified frequent or not in an uncertain sequence database then the projected database is considered. Since the projected database contains the sequences which include the particular old pattern the new pattern which is formed by appending some new elements to the old pattern will be present in the particular sequences in the projected database.

The position field in the projected database plays a major role in this stage. The new pattern appended will be present only after the particular pattern found in the sequences in the projected database. The newly added elements are checked in the following positions of the old sequence. Hence by this the need for scanning the whole database once again can be avoided. This will reduce the execution time to a considerable amount. After performing the hierarchical agglomerative clustering process. Those patterns which seem to be frequent can be considered separately for the UPrefixsapan process and then the pattern growth can be done on them from the projected database.

The Uprefixspan algorithm follows the anti-monotonicity property. It means if a pattern A is not probabilistically frequent then a pattern B satisfying A subset of B is not frequent. This property is used in frequent sequential pattern mining algorithm while performing the pattern growth process.
IV. CONCLUSION

Frequent pattern mining is an important concept as far as databases and real time applications are concerned. Hence the algorithms used for these purpose should be as efficient as possible. The algorithms which are studied here are some important algorithms used for this purpose. The prefixspan algorithm performs better that the other algorithms studied here. Because in U-prefixspan algorithm the database is scanned completely only once and further scanning for the pattern is done on the projected database. In other algorithms multiple scan in the database is required for frequent pattern mining. By using the clustering approach large number of patterns can be found at the same time. Clustering the sequences which follow the different patterns can be done and from that the count of members in each cluster can be obtained. This can be used to find the chances for a pattern to be frequent in a sequence database based on the threshold support value.

REFERENCES

[1] Zhou Zhao, Da Yan and wilfred Ng , “Mining Probabilistically Frequent Sequential Patterns in Large Uncertain Databases,” IEEE Transactions on Knowledge and Data Engineering, Vol. 26, No. 5, May 2014.