Discovring Frequent Item Sets from attribute and tuple uncertainty data model

Mr. Avinash Arun Pund, Prof.Prabhudev I.

Abstract— the database used in upcoming applications like RFID system, GPS system, location based services; sensor monitoring systems are frequently inaccurate in nature. The important problem of finding frequent item sets from a large uncertain database which is interpreted under the Possible World (PW). This issue is technically difficult because an uncertain database contains an exponential number of possible worlds. This can be solved by creating approximate algorithm that can efficiently and precisely find frequent item sets in a large uncertain database. The important problem of maintaining the mining result of a database that is growing (e.g. inserting new tuple).In particular incremental mining algorithms which allow Probabilistic Frequent Item set (PFI) results to be refreshed. So by using incremental mining algorithm reduces the need of re-executing the whole mining algorithm on the new database again, which is a lot more expensive and unnecessary. We observe how an existing algorithm that find exact item sets as well as approximate algorithm which support incremental mining. Also finding frequent item sets when providing range by using decision tree

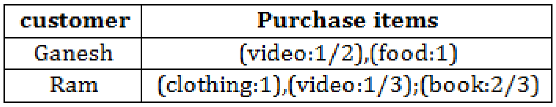
Index Terms— Approximate algorithm, Decision tree, frequent item sets, Incremental mining and uncertain data set.

.

1 Introduction

The databases used in many applications like the locations based services obtained through RFID and GPS systems, sensor monitoring system etc. are not exact due to measurement errors. Another example data collected from sensors in environment monitoring systems e.g. temperature and humidity which contain noise. Supermarket basket databases contain customer purchase behaviors and statistical information for predicting what a customer will buy in the future. The databases used in such type of application are called uncertain databases. In ordered information extractors confidence values are appended to rules for extracting patterns from unstructured data. Fig. 1 shows simple example of uncertain database which contain probabilistic information. The value associated with each item shows chance that a customer may buy that item in the future. These probability values may be obtained by analyzing the user browsing histories. For example if Ganesh visited the marketplace 10 times in the previous week out of that video item were clicked 5 times, then the marketplace application may conclude that Ganesh has a 50 percent chance of buying videos. To interpret uncertain databases the Possible World (PW) is frequently used. Although PW is perceptive and useful querying or mining under this notion is expensive because an uncertain database has an exponential number of possible worlds. Therefore Performing data mining under PW can be technically challenging.

Frequent item sets is nothing but sets of attribute values that appear together frequently in tuple of uncertain databases. Two important uncertainty models are attribute uncertainty model (shown in Fig. 1) and tuple uncertainty model. In this every tuple is associated with a probability to indicate whether it exists. The frequent item sets extracted from uncertain data are normally probabilistic in order to show the confidence placed on the mining results.

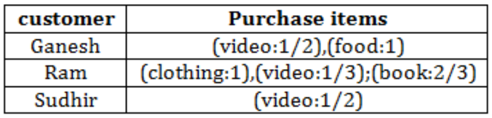


customer	Purchase items
Ganesh	(video:1/2),(food:1)
Ram	(clothing:1),(video:1/3);(book:2/3)

Fig 1 Uncertain database

PFI is defined as set of attribute values that occurs frequently with a sufficiently high probability. It is also called as support count that contains an item set. By using PW a database induces a set of possible worlds and each giving a different support count for a given item set. The support of a frequent item set is described by a probability mass function. For example by considering all possible worlds where item set {video} occurs twice then corresponding probability is 1/6.The support-pmf of a PFI can be capture by a Poisson binomial distribution for both attribute and tuple uncertain data model.

The growing database is the appending or insertion of tuple to the database. Tuple insertion is common in the applications that we consider.



customer	Purchase items
Ganesh	(video:1/2),(food:1)
Ram	(clothing:1),(video:1/3);(book:2/3)
Sudhir	(video:1/2)

Fig. 2 The new database after inserting new customer information.

In Fig. 2 it is shows a new database. After inserting new tuple mining result may changes. Therefore we need to obtain the PFIs for the new database. A simple way of refreshing the mining results is to re-evaluate the whole mining algorithm on the new database but this can be expensive however when new tuple are appended to the database at different time instants. Now consider D is old database and D+ is new database. If the new database D+ is similar to its older version D then it is probable that most of the PFIs extracted from D remain valid for D+ database. Incremental mining algorithms use the PFI of D to get the PFI of D+ instead of finding them from scratch.

By using approximate incremental mining algorithm will get the either exact or approximate frequent item set from the uncertain database. But when uncertain database is very large then finding exact frequent item set it is very expensive. So by providing range if we want certain frequent item set then by using decision tree we can classify the frequent item set we will get PFI. Old decision tree classifiers work with data whose values are known or exact. So extend such classifiers to handle data with uncertain data. The value uncertainty is represented by probability distribution function. Processing pdf is computationally more expensive. So by using series of pruning techniques that can significantly improve the efficiency. An easy way to handle data uncertainty is to abstract probability distributions by statistics such as means and variances. Another method is to consider the complete information passed by the probability distributions to build a decision tree. It is also called as Distribution based approach

2 LITRATURE SURVEY

Mining frequent item sets is an important problem in data mining and it is also the first step of deriving association rules. For this Apriori [4] and FP-growth [9] algorithms are used. These algorithms work well for databases with exact values but it is not clear how they can be used to mine probabilistic data. Therefore it is necessary to develop new algorithms for extracting frequent item sets from uncertain databases based on the Apriori framework and they can be considered for supporting other algorithms like FP-growth for handling uncertain data for uncertain databases.

Efficient frequent pattern mining algorithms based on the expected support counts of the patterns developed by Aggarwal et al. and Chui et al[3], [8]. But it is found that the use of expected support may leave important patterns missing.

For data mining for uncertain database Dynamic programming based algorithm were developed to retrieve PFIs from attribute uncertain databases [5]. But these algorithms compute exact probabilities and verify that an item set is a PFI in O (n^2) time complexity. It is necessary to develop new algorithms to avoid the use of dynamic programming and is able to verify a PFI much faster in O (n) time complexity. Approximate algorithms for deriving threshold based PFIs from tuple uncertain data streams were developed. While in Zhang et al. [9] only considered the extraction of sets of single items but it must be discovers patterns with more than one item. Sun et al. [7] developed an exact threshold based PFI mining algorithm. But it does not support attribute uncertain data. None of these solutions are developed on the uncertainty models.

Only few incremental mining algorithms that work for exact data have been developed. For example in the Fast Update algorithm [6] was proposed to efficiently maintain frequent item sets for the database to which new tuple are inserted. Incremental mining framework is motivated by FUP. In the FUP2 [7] algorithm was developed to use both addition and deletion of tuple. ZIGZAG [6] is another algorithm that examines the efficient maintenance of maximal frequent item sets for databases that are frequently changing. CATS Tree was introduced to maintain frequent item sets in growing databases. Another structure called Can Tree arranges tree nodes in an order that is not affected by changes in item frequency. The data structure is used to support mining on a changing database. But maintaining frequent item sets in evolving or growing uncertain databases has not been examined before. It is necessary to develop new incremental mining algorithms for both exact, approximate and in range PFI discovery. Also that algorithm must support both attribute and tuple uncertainty models but all this algorithms refer to algorithms that do not handle database changes. Therefore any change in the database requires a complete execution of these algorithms.

3 PROPOSED SYSTEM

Developing the algorithm that can extract exact and approximate frequent item set from the uncertain database. Also examining how to use the model based approach to develop classification by using decision tree and providing the range to finds the frequent item set.

3.1 Implementation:

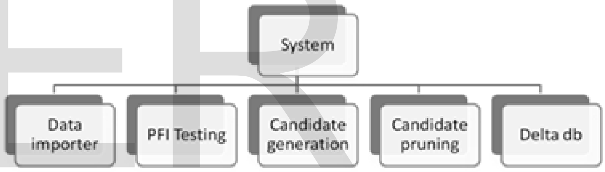


Fig 3 block diagram of proposed system

Atribute and touple uncertainty

Let *V* be a set of items. In the attribute uncertaint model every attribute value carries some uncertain information. A database *D* contains *n* tuple. Each transaction *tj* is related with a set of items taken from *V* .Each item *v* ϵ *V* exists in *tj* with an existential probability $\Pr(v \in tj) \in (0,1]$, which shows the chance that *v* belongs to *tj.*

In the tuple uncertainty models every tuple is associated with a probability value. By assuming the following variant each transaction *tj* ϵ *D* is associated with a set of items and an existential probability $\Pr(tj) \in (0,1]$ which indicates that *tj* exists in D with probability *Pr (tj).*

Probabilistic Frequent Item Sets

Let $I \subseteq V$ is the set of items the support of *I* denoted by s (*I*) and is the number of tuple in which *I* appears in a transaction database. In accurate databases s (*I*) is a single value. This is not possible in uncertain databases due to in different possible worlds s (*I*) can have different values.Let S (wj, *I*) be the support count of *I* in possible world wj. Then the probability that s (*I*) has a value of i.

$$pr^{I}(i) = \sum_{w_j \in W, S(w_j, I)=i} \Pr(w_j).$$

3.2 Threshold Based PFI mining

To obtain PFI from large uncertain database basic apriori algorithm is not sufficient which is used in [6].By using the PFI testing method instead of frequentness probability computation we can check quickly whether an item set *I* is threshold based PFI.

PFI testing Method:

Given the value of minsupport and minprobability can check whether *I* is threshold based PFI

Stage 1: finding real number μ_m satisfying the equation

Minprobability =1- *F* (msc (*D*) - 1, μ_m)

Stage 2: compute $\mu^I = \sum_{j=1}^{n} pIj$

In stage 2 databas e D has to be scanned once.

Stage 3: If $\mu^I \geq \mu_m$ then we conclude that I is a PFI otherwise not a PFI.

After this process we will gate the threshold based PFI by using the apriori based algorithm.

Algoritm used for finding frequent item sets

Input: Uncertain data D, minsup, minprob

Output: PFI: *F= {F1, F2… Fm}* F_k is set of k-PFIs

```
Start
μm = minExpsup (minsup, minprob, D);
C1.GenerateSingleItemCandidates (D);
k=1; j=0;
while |Ck| ≠ 0 do
for I ϵ Ck do
I.μ=0;
while (j++) ≤ n and | Ck | ≠0 do
for I ϵ Ck do
I.μ= I.μ + Pr (I ⊆ tj);
if I.μ ≥ μm then
Fk.push (I);
Ck.remove(I);
else if j ≥ n - | μm | then
if pruning (I, μm, j, n) = = true then
Ck.remove(I);
Ck+1.GenerateCandidate(Fk);
k=k+1; j=0;
return F;
end
```

in this algorithm all steps require to frequentness probability computation are replaced by PFI tesing steps.

3.3 Incremental Mining Process

Fig 4 shows the incremental mining process. Efficiently maintain a set of PFIs in a growing database when new tuple appended to it. Every tuple has a timestamp attribute which indicates the time that it is created. This timestamp is not used for mining. It is only used for differentiate new tuple from presented ones. Assume that D is the old database that contains n tuple and d (small d) is a delta database of n+ tuple and its timestamps are larger than those of tuple in D. And D+ is a new database which is a concatenation of the tuple in D and d and has a size of n. Given the set of PFIs and their s-pmf in D(old database) main challenge is to discover PFIs on D+ under the same minimum support and minimum probability values used to mine the PFIs of D.

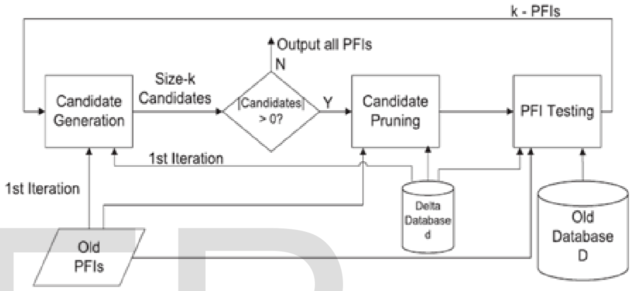


Fig 4. System architecture of incremental mining process.

Mining D+ is equivalent to updating the mining results for the arrival of units. The design of uncertain Fast Update algorithm is motivated by FUP. That algorithm maintains frequent item set results in a growing database whose attribute values are exact. The uFUP algorithm find frequent item sets in an Apriori manner. It utilizes a bottom up approach as shown in fig 4 and undergoes three phases in the kth iteration starting from k = 1.

Phase 1: Candidate Generation
Consider two cases of generating size-k candidate item sets in this phase: 1) k = 1 and 2) K > 1.

Phase 2: Candidate Pruning
The goal of this phase is to remove infrequent item sets from a set of size-k candidates.

Phase 3: PFI Testing
The objective of this phase is to verify whether these candidates are really k-PFIs.

The algorithm require for incremental mining process is described below the steps.

Input: *D*, d, F^D, minsup, minprob

```
Output: approx PFIs in D: $F^+ = \{F_1^+, F_2^+, ....F_m^+\}$
Start
$F^+ = \emptyset$;
$C_1^+$.GenerateSingleton (d, $F_k^D$);
k=1;
$\mu_m(D^+)$ = MinExpSup(minsup,minprob,$D^+$);
$\mu_m(D)$= MinExpSup(minsup,minprob,D);
$\mu_m = \mu_m(D^+) - \mu_m(D)$;
while $| C_k^+ | \neq 0$ do
    $C_k^+$.prune (d, $F_k^D$, $\mu_m$);
    if $| C_k^+ | \neq 0$ then
        $F_k^+ \leftarrow C_k^+$.test (D, d, $F_k^D$, $\mu_m(D^+)$);
    else
        break;
    $C_{k+1}^+$.GenerateCandidate ( $F_k^+$ );
    $k$=$k$+1;
return $F^+ = \{F_1^+, F_2^+, ....F_{k-1}^+\}$;
end
```

3.4 Decision tree

By using all this method we will get frequent item set either exact or approximate but if we want certain item sets when range is given then by using decision tree for uncertain database we can extract frequent item sets. In many application data uncertainty is common in that we have to consider probability distribution function. To construct decision trees from uncertain data using Averaging based approach and Distribution based approach. In uncertainty model a characteristic value is represented not a single value or point value but a probability distribution function. The tuple splitting is based on probability values that can give a natural elucidation to the splitting as well as the result of classification.

3.5 Generate Tree

Building a decision tree on tuple with numerical point value data is computationally demanding. A numerical attribute has a possibly infinite domain of real numbers inducing a larger searching space for the best "split point". By giving a set of n training tuple and a numerical attribute there are as many as n-1 binary split points or ways to partition the set of tuple into two nonempty groups. So finding the best split point is therefore computationally expensive.

An approach is considering the complete information carried by the probability distributions to build a decision tree. This approach is called as Distribution based. The goal is to invent an algorithm for building decision trees from uncertain data using the Distribution based approach. And establish a foundation on which pruning techniques are derived that can considerably improve the computational efficiency of the Distribution based algorithms.

4 System Design and Implementation

System Architecture

Fig 5 show the system architecture

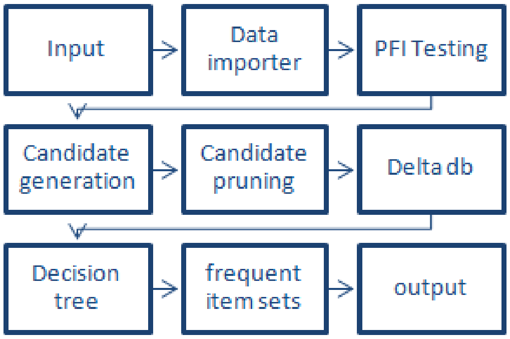
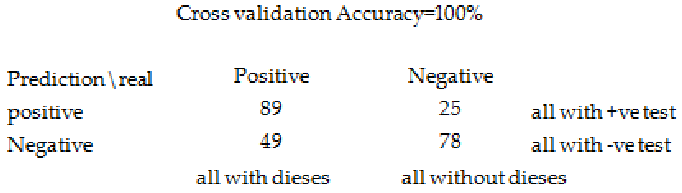


Fig 5. System Architecture

5 Result Analysis

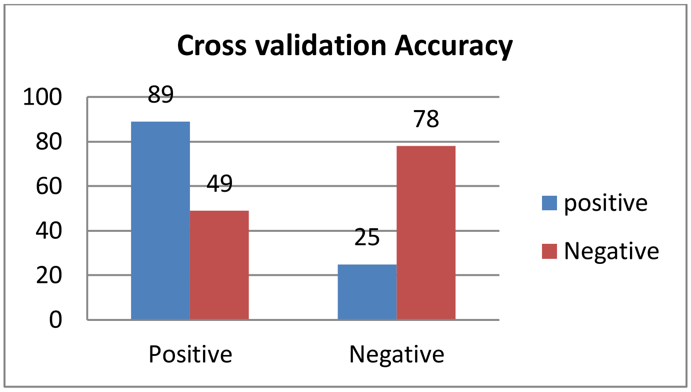
Input to the system is Data set which contents uncertain data of the patient with the attribute uncertainty.

Following confusion matrix and chart shows the cross validation accuracy.



Cross validation Accuracy=100%

Prediction \ real	Positive	Negative	
positive	89	25	all with +ve test
Negative	49	78	all with -ve test
	all with dieses	all without dieses	



6 CONCLUSION

We have learned attribute and tuple model based method to find threshold based Probability frequent item sets from large uncertain databases. The main idea is to approximate the support probability mass function of a PFI so that a PFI can be verified fast by adding the PFI testing steps into the basic algorithm. We also study how to maintain data mining result when there is some update in database by using two incremental mining algorithms. And this algorithm support both attribute and tuple uncertain data model. We examine how to extract frequent item sets when range is given with the help of decision tree. We will study how to implement and use the model based approach to develop other data mining algorithm for clustering and classification on uncertain data.

REFERENCES

[1] Liang wang,david wai-lok cheung,reynold cheng,member,IEEE,sau dan lee,and xuan S,yang “Efficient mining of frequent item sets on large uncertain databases”,2012

[2] Smith Tsang, Ben Kao, Kevin Y. Yip Wai-Shing Ho, Sau Dan Lee “Decision Trees for Uncertain Data”.

[3] C. Aggarwal, Y. Li, J. Wang, and J. Wang, “Frequent Pattern Mining with Uncertain Data,” Proc. 15th ACM SIGKDD Int’l Conf.Knowledge Discovery and Data Mining (KDD), 2009.

[4] R. Agrawal, T. Imieli_nski, and A. Swami, “Mining Association Rules between Sets of Items in Large Databases,” Proc. ACM SIGMOD Int’l Conf. Management of Data, 1993.

[5] T. Bernecker, H. Kriegel, M. Renz, F. Verhein, and A. Zuefle,“Probabilistic Frequent Itemset Mining in Uncertain Databases,”Proc. 15th ACM SIGKDD Int’l Conf. 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: An Incremental Updating Technique,” Proc. 12th Int’l Conf. Data Eng. (ICDE), 1996.

[7] D. Cheung, S.D. Lee, and B. Kao, “A General Incremental Technique for Maintaining Discovered Association Rules,” Proc. Fifth Int’l Conf. Database Systems for Advanced Applications (DASFAA), 1997.

[8] C.K. Chui, B. Kao, and E. Hung, “Mining Frequent Itemsets from Uncertain Data,” Proc. 11th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD), 2007.

[9] J. Han, J. Pei, and Y. Yin, “Mining Frequent Patterns without Candidate Generation,” Proc. ACM SIGMOD Int’l Conf. Management of Data, 2000.