An Improvised on Mining Frequent Item Sets on Large Uncertain Databases

Smt. Sushama Anantrao Deshmukh
Department of Computer Science & Engineering
Government College of Engineering Aurangabad, Maharashtra, India

Abstract— Data processed in emerging applications, such as site-based services, sensor monitoring systems and data integration, are often inaccurate. In this paper, the important problem of extracting sets of frequent objects from a large uncertain database, interpreted under the possible World Seminar (PWS) is presented. This problem is technically difficult because an uncertain database contains an exponential number of possible worlds. By observing that the mining process can be modeled as a binomial distribution of Poisson, an algorithm has been developed, which makes it possible to discover efficiently and precisely sets of frequent objects in a very uncertain database. The important issue of maintaining the mining result for a scalable database (e.g. by inserting a tuple) can be presented. More precisely, the proposed exploration algorithm can refresh the probabilistic results of the set of frequent objects (PFI). This reduces the need to re-run the entire extraction algorithm on the new, often more expensive and unnecessary database. The proposed algorithm can support progressive extraction and provides accurate results on uncertain database extraction. The in-depth evaluation of the actual data defined to validate the approach is carried out.

Key words: Uncertain dataset, Frequent item sets, Approximate algorithm, Incremental mining, PFI, pmf

I. INTRODUCTION

Databases that are used in different important and considerably new applications are more or less uncertain. For an instance , if user location is obtained through the usage of RFID along with GPS or independently using GPS then the systems are not accurate due to measurement errors [19], [20]. As another example, data collected from sensors in habitat monitoring systems (e.g., temperature and humidity) are noisy [17]. Buyers' purchasing behaviors, as reflected in supermarket basket databases, contain statistical information to predict what a buyer will buy in the future [3], [6]. The integration and registration binding tools also associate the confidence values with the output tuples as a function of the matching quality [16]. In structured information extractors, confidence values are added to pattern extraction rules from unstructured data [19]. To meet the growing application requirements for processing a large amount of uncertain data, uncertain databases have recently been developed [10], [16], [19], [20], and [23]. The realization of data mining under possible global semantics (PWS) can be technically difficult. In fact, the extraction of uncertain data has recently attracted the attention of research [3]. For example, in [20], an efficient clustering algorithm has been developed for uncertain objects in [20], Bayes and decision tree classifiers designed for uncertain data were studied. Here, the algorithm for finding sets of frequent objects (i.e. sets of attribute values that appear together frequently in tuples) for uncertain databases has been developed. The proposed algorithm can be applied to two important uncertainty models: attribute uncertainty and tuple uncertainty, where each tuple is associated with a probability to indicate whether it exists [15], [16], and [19]. The sets of frequent objects discovered from uncertain data are naturally probabilistic, in order to reflect the confidence placed in the mining results.

A frequent probabilistic element (PFI) is a set of attribute values that frequently occurs with a sufficiently high probability. A database induces a set of possible worlds, each giving a support account (different) for a given set of objects. Therefore, the support of a set of frequent objects is described by a probability mass function (pmf). A simple way to find PFIs is to conduct frequent schemes of all possible universes obtained by the possible world semantics (PWS), and then to record the probabilities of occurrences of these models. This is impractical, because of the exponential number Of algorithms has been recently developed to successfully retrieve PFIs without instantiating all possible worlds [6], This algorithm can verify if a set of objects Is a PFI in time O (n2) (where n is the number of tuples contained in the database.) However, experimental results reveal that they may require a long time to complete (e.g., Of real data of 300k, the dynamic programming algorithm of [6] requires 30.1 hours to find all the PFIs.) The pmf support of a PFI can be captured by a Poisson binomial distribution, both for Data of Uncertainty and tuple index. The proposed algorithm uses this intuition to propose a method of approximating the pmf of a PFI with a Poisson distribution, which can be estimated efficiently and efficiently. This algorithm can check a PFI in O (n2) and is therefore more suited to large databases. The algorithm can be used to exploit the sets of frequent objects whose probabilities of being sets of frequent objects are greater than a threshold defined by the user [6]. The algorithm needs only a very short time to find all the PFIs with respect to the existing algorithm, i.e. four orders of magnitude faster than the method used in [6].

The important issue of maintaining mining results for changing or evolving databases is presented here. The evolving data type that we are discussing here is the addition or insertion of a batch of tuples into the database. The insertion of Tuple is common in the applications we consider. For example, a GPS system may have to manage location values due to the registration of a new user. In an on-line market application, information about new purchasing transactions can be important. The important issue of maintaining operating results for changing or changing databases is presented here. The evolving data type that we are discussing here is the addition or insertion of a batch of tuples into the database. The insertion of Tuple is common in the applications we consider. For example, a GPS system may have to manage location values due to the registration of a new user.

II. LITERATURE REVIEW

The sets of frequent mining objects are an important problem in data mining, and are also the first step to derive the rules of association [4]. As a result, many effective article-setting activities an algorithm (e.g., Apriori [4] and FP-growth [18]) has been proposed. Although these algorithms work well for databases with precise values, it is not clear how they can be used to extract probabilistic data. Here, an algorithm for extracting sets of frequent objects from uncertain databases has been developed. Although the algorithm is developed on the basis of the Apriori framework, they can be considered to support another algorithm (eg FP growth) for the management of uncertain data. For uncertain databases, Aggarwal et al. [2] and Chui et al. [14] has developed an efficient algorithm for frequent model extraction based on the expected number of accounts on the models. Therefore, they proposed to calculate the likelihood of a pattern being frequent and introduced the notion of PFI. In [6], solutions based on dynamic programming have been developed to recover PFIs from uncertain databases. However, their algorithms calculate the probabilities and verify that a set of objects is a PFI in time O (n2). The algorithm avoids the use of dynamic programming and can check a PFI much faster (in O (n)).

Zhang et al. [21] only considered as extracting singletons (that is, sets of unique articles) is the solution discovers patterns with more than one element. However, it does not support the uncertain attribute data considered in this paper, other work on the retrieval of frequent patterns from imprecise data include: [9], which studied frequent patterns on noisy data; [21], which examined the association rules on fuzzy sets. However, none of these solutions is developed on the uncertainty models studied here. For the evolution of databases. Some mining algorithms that work for the data have been developed. For example, in [11], the fast update algorithm (FUP) has been proposed to efficiently maintain sets of frequent elements for a database to which new tuples are inserted. The mining frame is inspired by FUP. In [12], the FUP2 algorithm was developed to handle both addition and deletion of tuples. A model based approach [23] is developed to handle uncertainty of data at attribute and tuple level to extract threshold-based PFIs from large uncertain databases.

ZIGZAG [1] also examines the efficient maintenance of maximum frequent element sets for ever-changing databases. In [13], a data structure, called CATS Tree, was introduced to maintain sets of frequent objects in scalable databases. Another structure, called CanTree [1], organizes the tree nodes in an order that is not affected by the frequency changes of the elements. The data structure is used to support extraction on a changing database. To the best knowledge, the maintenance of frequent article sets in evolving uncertain databases has not been discussed before. The proposed algorithm can also support uncertainty and uncertainty models of tuples

III. PROPOSED SYSTEM

Let V be a set of objects. The algorithm adopts the following variant [6]: a database D contains only tuples or transactions. Each transaction tj is associated with a set of elements drawn from V. Each element vεV exists in tj with an existing probability Pr $tj\varepsilon tj$ ε 0, 1, which indicates the randomness that v belongs to tj. Under the possible global semantics, D generates a set of possible worlds. Each world consists of a subset of attributes of each transaction, occurs with the probability Pr(wi). The probabilities are one, and the number of possible worlds is exponentially large. The goal is to discover frequent patterns by performing updation and deletion of retail dataset without widening D in possible worlds. Each transaction $tj \in D$ is associated with a set of elements and an existential probability Pr $tj \in$ (0, 1) which indicates that tj exists in D with probability Pr(tj). Again, the number of possible worlds

Many efficient item set mining algorithms (e.g., Apriori and FP-growth) have been proposed. While these algorithms work well for databases with precise values, it is not clear how they can be used to mine Probabilistic data. We develop algorithms for extracting frequent item sets from uncertain databases. Although our algorithms are developed based on the Apriori framework, they can be considered for supporting other algorithms for handling uncertain data. The efficient frequent pattern mining algorithms based on the expected support counts of the patterns is used for uncertain databases. The use of expected support may render important patterns missing. Hence, they proposed to compute the probability that a pattern is frequent, and introduced the notion of PFI. Dynamic-programming based Solutions were developed to retrieve PFIs from attribute-uncertain databases.

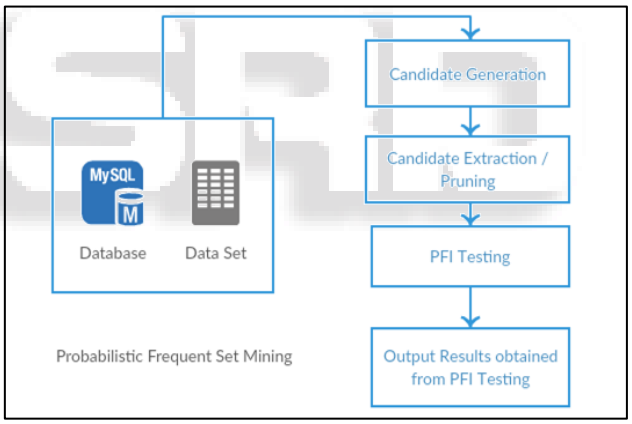

Fig. 3.1: Proposed Architecture

A. *Improvised Apriori based PFI Mining Algorithm*

Input: Uncertain database D, minsup , minprob

Output: All PFI: F={ $F_1, F_2 \dots\dots\dots F_m$ }// F_k is set of k-PFIs

Begin

μ_m =MinExpSup(minsup,minprob,D)

C_1 .GenerateSingleItemCandidates(D);

k=1;j=0;

while l C_k l≠ 0 do‹‹‹

for each I ϵ C_k do

I.μ = 0;

while (++j) ≤ n and l C_k l≠ 0 do

foreach I ϵ C_k do

I.μ = I.μ +pr(I ⊆ t_j);

if I.μ ≥ μ_m then

F_k .push(I);
C_k . remove(I);
else if j≥ n- l μ_m l then
if pruning (I, μ_m ,j,n) == true then
C_k . remove(I);
C_{k+1} .GenerateSingleItemCandidate(F_k);
k= k+1; j=0;
return F;
End

B. Improvised Uncertain Tuple Fast Update Algorithm

The algorithm takes the uncertain database D, minsup (minimum support) and minprob (minimum probability), Probabilistic Frequent Item F^D and d as evolving data (i.e. addition, deletion or update of data). The output of the algorithm is set of k-PFIs (Probabilistic Frequent Item). First load the PFIs which retrieve from above algorithm 1. Algorithm retrieve PFIs from evolving data start from k=1, size K- PFI are generated, then size (k+1) candidate item sets are driven from k-PFIs, based on which the (k+1)-PFIs are found. This process goes to repeated, until no longer candidate item sets can be discovered. The old PFI and new PFI combined at each level which used for finding probabilistic frequent itemset.

Input: D, d, F^D , minsup, mmprob
Output: Approximate PFIs in D:
$+¿ F^{¿}$ = { $+¿ F_1^{¿}$, $+¿ F_2^{¿}$,…… $+¿ F_m^{¿}$ }
Begin
$+¿ F^{¿}$ = Ø;
.GenerateSingleton(d, F_1^D);
k = 1;
μ_m ($+¿ D^{¿}$ =MinExpSup(minsup,minprob, $+¿ D^{¿}$);
μ_m (D) ¿ = MinExpSup(minsup, minprob, D);
while | $+¿ C_k^{¿}$ |≠ 0 do
Prune(d, $F_k^D, -¿ \mu_m^{¿}$);
if | $+¿ C_k^{¿}$ | ≠ 0 then
$+¿ F_k^{¿}$ ← $+¿ C_k^{¿}$ .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

C. Standard statistical properties of s-pmf

An interesting observation about s (I) is that it is essentially the number of successful trials of fish [21]. To explain, let XjI be a random variable, which is equal to one if I am a subset of the elements associated with the transaction tj, or no other. Note that Pr (I C tj) can be easily calculated in our uncertainty models.

– For the uncertainty of the attributes, Pr (I tj) = II Pr (v Σ tj).
– For Tuple uncertainty,

Given a database of size n, each I is associated with random variables X1, X2 .... Xn. In the two uncertainty models considered in this paper, all tuples are independent. Consequently, these n variables are independent and represent n trials of fish. In addition, XI = Σnj = 1 XI follows a Poisson binary distribution. Then, we observe an important relationship between -

XI and PrI (i)
PrI (i) = Pr (XI = i).

It is simply because XI is the number of elements that exist in the database. Therefore, the s-pmf of I, ie PrI (i) is the pmf of XI, a Poisson binomial distribution. Using the above formula, we can rewrite the formula that calculates the probability of Frequency of I, as,

Prfreq (I) = Σ Pr ((XI = i).
= Pr (XI> msc (D)).

Therefore, if the cumulative distribution function (cdf) of XI is known, Prfreq (I) can also be evaluated. Next, we discuss an approach to address this cdf, in order to efficiently calculate Prfreq (I).

D. Jaccard Coefficient

We base our item set mining approach on the similarity of item covers rather than on item set support. In order to measure the similarity of a set of item covers, we start with the Jaccard index [22], which is a well-known statistic for comparing sets. For two arbitrary sets A and B it is defined as J(A, B) = |A ∩ B|/|A ∪ B|. Obviously, J(A, B) is 1 if the sets coincide (i.e. A = B) and 0 if they are disjoint (i.e. A∩B = ∅). For overlapping sets its value lies between 0 and 1. The core idea of using the Jaccard index for item set mining lies in the insight that the covers of (positively) associated items are likely to have a high Jaccard index, while a low Jaccard index indicates independent or even negatively associated items. However, since we consider also item sets with more than two items, we need a generalization to more than two sets (here: item covers). In order to achieve this, we define the carrier LT (I) of an item set I w.r.t. a transaction database T as LT (I) = {k ∈ INn | I ∩ tk 6= ∅} = {k ∈ INn | ∃i ∈ I : i ∈ tk} = S i∈I KT ({i}). The extent rT (I) of an item set I w.r.t. a transaction database T is the size of its carrier, that is, rT (I) = |LT (I)|. Together with the notions of cover and support (see above), we can define the generalized Jaccard index of an item set I w.r.t. a transaction database T as its support divided by its extent, that is, as

$$Jt(I) = \frac{sT(I)}{rT(I)} = \frac{Kt(I)}{Lt(I)}$$

Clearly, this is a very natural and straightforward generalization of the Jaccard index. Since for an arbitrary item a ∈ B it is obviously KT (I ∪ {a}) ⊆ KT (I) and equally obviously LT (I ∪ {a}) ⊇ LT (I), we have sT (I ∪ {a}) ≤ sT (I) and rT (I ∪ {a}) ≥ rT (I). From these two relations it follows JT (I ∪ {a}) ≤ JT (I) and thus that the generalized Jaccard index w.r.t. a transaction database T over an item base B is an anti-monotone function on the partially ordered

set (2B, ⊆). Given a user-specified minimum Jaccard value Jmin, an item set I is called Jaccard-frequent if JT (I) ≥ Jmin. The goal of Jaccard item set mining is to identify all item sets that are Jaccard-frequent in a given transaction database T. Since the generalized Jaccard index is anti-monotone, this task can be addressed with the same basic scheme as the task of frequent item set mining. The only problem to be solved is to find an efficient scheme for computing the extent rT (I).

E. Key Index Parameters for Result Classification

In information retrieval with binary classification, precision (also called positive predictive value) is the fraction of retrieved instances that are relevant, while recall (also called sensitivity) is the fraction of the relevant instances that are retrieved. Precision and recall are therefore based on understanding and measuring relevance.

In simple terms, high accuracy means that an algorithm returns significantly more relevant than irrelevant results, while a high recall means that an algorithm has yielded the most relevant results.

The most important category measurements for binary categories are:

Precision

$$P = TP/(TP + FP)$$

Recall

$$R = TP/(TP + FN)$$

F Measure

$$tp + tn/tp + tn + fp + fn$$

IV. CONCLUSION

In this paper, we propose a model-based approach to extract probability based and mean-based PFIs from large uncertain databases. Its main idea is to approximate the s-pmf of a PFI by some common probability model, so that a PFI can be verified quickly. We also study two incremental mining algorithms for retrieving PFIs from evolving databases. They support both attribute and tuple uncertain data. We will examine how to use the model based approach to develop other mining algorithms (e.g., clustering and classification) on uncertain data. It is also interesting to study efficient mining algorithms for handling tuple updates and deletion

REFERENCES

[1] Adriano Veloso and Wagner Meira Jr. and Marcio de Carvalho and ´ Bruno Possas and Srinivasan Parthasarathy and Mohammed Javeed ˆ Zaki. Mining Frequent Itemsets in Evolving Databases. In SDM, 2002.
[2] C. Aggarwal, Y. Li, J. Wang, and J. Wang. Frequent pattern mining with uncertain data. In KDD, 2009.
[3] C. Aggarwal and P. Yu. A survey of uncertain data algorithms and applications. TKDE, 21(5), 2009.
[4] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules ´ between sets of items in large databases. In SIGMOD, 1993.
[5] O. Benjelloun, A. D. Sarma, A. Halevy, and J. Widom. ULDBs: databases with uncertainty and lineage. In VLDB, 2006.
[6] T. Bernecker, H. Kriegel, M. Renz, F. Verhein, and A. Zuefle. Probabilistic frequent itemset mining in uncertain databases. In KDD, 2009.
[7] C. J. van Rijsbergen. Information Retrieval. Butterworth, 1979.
[8] L. L. Cam. An approximation theorem for the Poisson binomial distribution. In Pacific Journal of Mathematics, volume 10, 1960.
[9] H. Cheng, P. Yu, and J. Han. Approximate frequent itemset mining in the presence of random noise. Soft Computing for Knowledge Discovery and Data Mining, pages 363–389, 2008.
[10] R. Cheng, D. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In SIGMOD, 2003.
[11] D. Cheung, J. Han, V. Ng, and C. Wong. Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique. In ICDE, 1996.
[12] D. Cheung, S. D. Lee, and B. Kao. A General Incremental Technique for Maintaining Discovered Association Rules. In DASFAA, 1997.
[13] W. Cheung and O. R. Za¨ıane. Incremental mining of frequent patterns without candidate generation or support constraint. In IDEAS, 2003.
[14] C. K. Chui, B. Kao, and E. Hung. Mining frequent itemsets from uncertain data. In PAKDD, 2007.
[15] G. Cormode and M. Garofalakis. Sketching probabilistic data streams. In SIGMOD, 2007.
[16] N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In VLDB, 2004.
[17] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In VLDB, 2004.
[18] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In SIGMOD, 2000.
[19] J. Huang et al. MayBMS: A Probabilistic Database Management System. In SIGMOD, 2009.
[20] R. Jampani, L. Perez, M. Wu, F. Xu, C. Jermaine, and P. Haas. MCDB: A Monte Carlo Approach to Managing Uncertain Data. In SIGMOD, 2008
[21] Q. Zhang, F. Li, and K. Yi. Finding frequent items in probabilistic data. In SIGMOD, 2008.
[22] P. Jaccard. Etude comparative de la distribution florale dans une portion des Alpes ´ et des Jura. Bulletin de la Soci´et´e Vaudoise des Sciences Naturelles 37, 547-579. France 1901
[23] Liang Wang, David Wai-Lok Cheung, Reynold Cheng, Sau Dan Lee and Xuan S. Ya, "Efficient Mining of Frequent Item Sets on Large Uncertain Databases", IEEE Transactions on Knowledge and Data Engineering, vol.24, no.12, pp.2170,2183, Dec. 2012.

