International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 5, Issue 6, November - December 2016 ISSN 2278-6856

Fast Computation of Probabilistic Frequent Patterns in Uncertain Databases

Dr. Telkapalli Murali Krishna [1], Amsalu Tomas [2]

[1, 2] Department of CS, School of Informatics, Wolaita Sodo University, Ethiopia

Abstract

In practice the data in the database is always stochastic by nature. The probabilistic assumption of data in deterministic databases can be also formulated as uncertainty of data with a value 1.0. The uncertain nature of the data may come from dynamic error-rapid change in the environment or scene of the data source, drifting in reading value like temperature, and noise of the equipment, distance etc. This nature can be seen in both object and attribute level uncertainties. In general, preprocessing in data warehouse− typically ETL, data integration, data granularity, ambiguous entities or missing data values can be main causes of the uncertain database. Mining such databases needs more attention in computing support count of itemset that are uncertain with their corresponding existential probabilities. This paper focuses on faster and efficient method to compute probabilistic support count of an itemset using divide and conquer approach as transforms on a sample dataset. The sampling technique has been seen as a crucial technique to minimize the complexity as the size of dataset and support computation are proportional. The probabilistic support can be found from inverse transform of the interpolated values of probabilistic support function. The performance of this approach is compared with the existing probabilistic pattern mining algorithms.

Keywords: uncertainty, computing support,probabilistic pattern mining, transforms,

1. Introduction

In many applications such as location-based services, natural habitat monitoring, web data integration, and biometric applications, the values of the underlying data are inherently noisy or imprecise due to reasons such as imprecise measurement, outdated sources, partially complete, or sampling errors. For example, in the scenario of moving objects (such as vehicles or people), it is impossible for the database to track the exact locations of all objects at all-time instants. Therefore, the location of each object is associated with uncertainty between updates [2]. In market basket analysis, the purchasing behaviors of customers are captures continuously so that it contains probabilistic information for predicting what a customer will buy in near future. This uncertainty of datasets/data sources needs high consideration as it results in non-atomic data values in the database so that the data mining quality can be improved.

Both level of uncertainties, object/tuple and attribute, may occur because of data granularity, data integration, ambiguous entities or missing data values cases in the database. For example, if a database has of attribute values SSN,EmpId,Age,Signature,… one of the values might be missed when they are recorded, which result in tuple uncertainty. Attribute uncertainty is the uncertain nature of every attribute values in the database that result the whole database to be uncertain.

Unlike deterministic databases where computing support is a fixed occurrence counting of an itemset, uncertain databases itemset is uncertain with probability which is a random variable with a support to be computed by considering all the possible outcomes. This is not a trivial as the size of database and itemsets are large. Thus, it is important to redefine frequent itemset under uncertain environments before heading to discover frequent itemsets.

Many efforts have been done on sampling techniques in these scenario to find a smaller sized database. Some of the various techniques are shown by Tioveninetal(1996), B. Chandra etal(2009) for deterministic databases and T. Caldersetal(2010) for uncertain databases with a focus on minimizing number of scans and I/O cost besides the size of the database.

B. Chandra & S. Baskar(2011) mentioned that hub-averaging algorithm can be used to rank the transactions that can represent the whole database about by concluding 0.25% database size with 99.7 accuracy. The paper is organized as follows: review of related works of existing algorithms for counting probabilistic support in section-2, the proposed sampling technique followed by faster approach to compute probabilistic support in section-3 and finally the comparative experimental results are discussed.

2. Related Works

Let I = {i_1 , i_2 , . . . , i_n } be a set of distinct items. Given an uncertain transaction database UDB as shown in table 1 below, each transaction is denoted as a tuple <Tid, Y >

where Tid is the transaction identifier, and Y = {$y_1(p_1)$, $y_2(p_2)$, . . . , $y_m(p_m)$} where an item y_i may be present with a probability p_i, in the Tid tuple/transaction. The number of transactions containing X in a given UDB with a probability in the range (0,1) is a random variable, the probability of the presence of item X in the given transaction, denoted as *Sup(X).* There are many algorithms devised to mine frequent patterns from uncertain databases: generate–test, hyper-structures, and pattern growth algorithms. There are two approaches to compute support for frequent patterns assuming that the occurrence probability,p_i, mutually independent:

2.1. Expectation based approach

The expectation of the support of an itemset to measure whether this itemset is frequent. C. C. Aggarwal etal(2008) and Chun K. etal(2007) shown that *eSup(X) is* the sum of products of probabilities of an itemset in a given possible world by a single database scan through the dataset D as

eSup(X) = $\sum_{j=1}^{D}\prod_{x\in X} P_{tj}(x)$, P_{tj}(x) is the probability of X in j^{th} transaction at possible world W_i.

However, as it is show in [3] this approach results in inconsistent and misleading mining results as it is not concerned with the frequentness probability.

2.2. Probabilistic approach

This approach uses the probability of the support of an itemset to measure its frequency.[1]So, an itemset is frequent if and only if the frequent probability of such itemset is larger than a given probabilistic threshold. [1] The probabilistic problem definition of this problem is:

Definition 2.2.1: An uncertain item is an item x 2 I whose presence in a transaction t 2 T is defined by an existential probability P(x 2 t) 2 (0; 1). A certain item is an item where P(x 2 t) 2 f0; 1g. I is the set of all possible items.

Definition 2.2.2: An uncertain transaction t is a transaction that contains uncertain items. A transaction database T containing uncertain transactions is called an uncertain transaction database.

It considers the possible world semantics which is stated as,P(w), is:

$$P(w)=\prod_{t\in I}\left(\prod_{x\in t} P\,(X\in t) * \prod_{x\in T-t}\left(1-P(X\in t)\right)\right)$$

Definition 2.2.3 A Probabilistic Frequent Itemset (PFI) is an itemset with a frequentness probability of at least τ.

To formulate the problem in general, Given an uncertain transaction database T, a minimum support scalar *minSup* and a *frequentness probability threshold* τ, *find all probabilistic frequent itemsets.*

(Thomas Bernecker and etal,2012) they have shown that how to efficiently compute probabilistic support using the following probability generating functions.

$$F^i=\prod_{t\in\{t_1\ldots t_i\}}\left(1-P(X\in t)\right)+P(X\in t).x$$
$$=\sum_{k\in\{0\ldots i\}} c_k\, z^k$$

Moreover they have shown that for the coeficient in the probability function generation where k is greater that *minSup* are pruned sothat the complexity is minimized to O(minSup.N).

Despite lots of efforts to approximate and directly compute probabilistic support of items, efficient sampling techniques can minimize the computation with a careful selection of samples, much smaller in size than input database, that can represent the whole database.

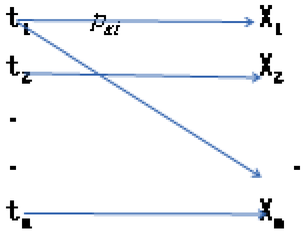
3. Proposed Algorithm

3.1. Sampling

It is clear that the assumption in frequent pattern mining is that the database has a binary itemset values 0 and 1 for the absence and existence respectively, which needs to be transformed to binary matrix matrix before beginning mining task. Similarly, the itemsets in uncertain databases have also existential probabilities which in turn needs to be transformed into probability matrix, called uncertain data model. In order to speed up frequent pattern mining it is important to reduce the database activities in such a way that we can have a sample of the probabilistic database that can hold probably almost all rules to be generated. Minimal size of the database is influential factor dealing about I/O and memory versus accuracy tradeoffs. But this task is cumbersome in deterministic databases where every item is certain and the samples may miss lots of itemsets for rule generation.

However, the uncertain data model can be sampled where the final results of the original and sampled differs very slightly. As [26] stated the sampling technique used is by generating random number r in [0,1] for every transaction ti and reject the item if its existential probability is less than r, which is rejection sampling. This method is naïve and suffers from rejecting many transactions for very low r so that [26] recommends to sample it for some n times. [29] Sampling by using Hub-Averaging algorithm introduce for finding frequent 1-itemsets by B. Chandra and S. Bhaskar(2011) has been used to extract samples from uncertain dataset. As [29] stated the transactions are hubs and items are authority.

Definition: Given uncertain database(UDB) D with item exitential probabilities p_i in each transaction, a weighted directed bipartite graph G(u,v) that has an edge every occurrence of item X from hub(i) to X where the weight of the edge is its p_i.



for all i 0,1,...m.

Then,

$$a(i) = \sum_{j \in H(i)} h(j) \,.\, p_j(X) \qquad (1)$$

$$h(i) = \frac{\sum_{j \in H(i)} a(j).p_j(X)}{|A(i)|} \qquad (2)$$

3.2. Probabilistic Support Computation

Given an uncertain database or transactions where items are with existential probabilities ***p***which can be written as ***P(X,t)*** | *t*□*D,X*□*I*where ***X*** is an itemset and t is transaction Id where X to exists, the minimum support threshold ***minSup***, and the probabilistic support threshold **τ**, then existence of an itemset *X* in the transaction *t*, *P(X,t)*, can be formulated as a independent Bernoulli trials. Lets assume success be the existence of itemset *X* in t alternatively represented as 1 with probability *p* and failure when the itemset *X* is absent in transaction t with probability *1-p* = *q* which is represented by 0. The support count of an itemset X, *Sup(X)*, can be easily seen in each of possible world representations *W* = $w_1, w_2, \dots w_n$. Therefore, *Sup(X)* is dependent on probability mass function of each of these possible worlds which goes to be a binomial random variable computed over these sequence of independent Bernoulli trials. and computed as:

$S(X) = \sum_{i \geq minSup}^{n} P(X = i)$, which is *cdf*of *X* for

$0 < i <= minSup$

$$= 1 - \sum_{i=1}^{minSup-1} P(X = i) \qquad (3)$$

The probabilistic support count of an itemset X is frequent, which is at least greater than minSup threshold, can be calculated as follows:

The frequentness probability of an itemset X can be computed using probability generating function as a discrete binomial random variable, X~B(n,p), from a Bernoulli trials , the number of successes in *n* trials, with probability *p* of success in each trial, is:

$$G(z) = \sum_{k=1}^{n} \binom{n}{i} p^i (1-p)^{n-i} z^i, \; for\ i=0,1,\dots n$$

$$=[(1-p)+pz]^n \; = (q+pz)^n \qquad (4)$$

Since our assumption is the existential probability of itemset/s in each transaction is mutually independent ,n will be 1. Then, for transaction t□{$t_1, t_2, \dots t_k$} where itemset X is uncertain, the pmf of the sum of itemset is the convolution of individual pmf of each itemset with an existential probability in (0,1) as (T. Bernecker and etal, 2009) stated.

$$G_X(z) = \prod_{i=1}^{k} \left(1 - P(X = t_i)\right) + P(X = t_i).z$$

$$= \sum_{k=0}^{i} p_x(k) z^k \qquad (5)$$

We can see that $G_X(z)$can be computationally highly expensive for an uncertain database where uncertainty is high for the itemsets (i.e. for large *k* int$_k$and large *minSup*) which is about O(N^2) and optimaly reduced to O(*minSup*.N) as shown in [2]. To minimize the computation lets consider Discrete Fourier Transform(DFT) and define $G_X(z)$ as *DFT[8]* where the coefficients belongs to z^k is the value of probabilistic support of a given itemset in the first i transactions is k.

Lets consider a vector V_i contains coefficients $p_x(k)$ of each t_i in *(3)*:

$$V_i = (p_0, p_1, \dots p_n) \text{ , for i = 0,1,...,k}$$

and vector C of size m to store the probabilities $p_x(k)$ of the convolution of two subsequent $V_{i.}$

As we shown in (2) each $V_i$has coefficients p and q that corresponds to each t_i ,

$$G^i(z) = (1 - P_i(X \in t) + P_i(X \in t) \cdot z)$$

$$= q_i + p_i z \qquad (6)$$

and vectors $V_i = (q_i, p_i)$ (7)

Hence, the function *G(z)* can be computed by applying repetitive FFT of these vectors subsequently, that is a convolution of all $G^i(z)$ for i = 0,1,. . . ,n-1.

$$G(z) = G^0(z).G^1(z). \dots G^{n-1}(z)$$

$$=[(1-P_i(X \in t) + P_i(X \in t) \cdot z)] \;.\; [(1-P_{i+1}(X \in t) + P_{i+1}(X \in t) \cdot z)]. \;\dots\; .[(1-P_{n-1}(X \in t) + P_{n-1}(X \in t) \cdot z)]$$

$$=[q_i + p_i z].[q_{i+1} + p_{i+1} z]. \dots .[q_{n-1} + p_{n-1} z]$$

and $G(z)_0 = 1$ since $(q+pz)^0 = 1$

Hence, $G(z) = \prod_{j=0}^{n-2} G(z)_j$

By taking the iterative version of FFT of vectors V_is of each $G^i(z)$,we can find the coeficient vector for G(z) by taking the inverse of convolution of all vectors V_is. As we know from properties of FFT the optimal complexity is when the size of the two vectors is a power of 2. Therefore, if the size of coefficient vector V is padded by zeros to the nearest ceiling power of 2. Since our goal is to find the probability, which is a coefficient, of z^k for $k \leq$ minSup the worst case complexity of non-recursive computation is O(n) and we need at least three multiplications for a single convolution. The recurrence relation is then

$$T(n) = 2T(minSup/2) + O(n)$$

The complexity is $O(minSup.logN)$

Algorithm:Compute-probabilistic support of uncertain itemset in the database

```
Input : coefficient vectors $V_i$ of uncertain itemset in a transaction $t_i$
Output: probabilistic support of the  itemset
functiongeneratePro()
$G(z)_0 \leftarrow 1$
for j=1 to n  begin
if($V_i$.size() >= minSup-1) then
for k=0 to minSup-1
$V_k = V_k$
$G(z)_i = FFT(V_j) * G(z)_{i-1}$
end
$C = FFT^{-1}(G(z)_n)$
return C[minSup-1]
```

4. Experimental Results

As I have discussed in the survey report of uncertain database mining, when the size of database is getting very large, big-data, it's merely impossible to compute probabilistic item sets efficiently in the case of PCs specification on which we working on. Even though enough resources are available, the time is so expensive and grows in exponential time in the size of the database and number of distinct items in the database. Sampling, if carefully taken and representative of the original database, can be one of efficient way to address this issue.

Considering the assumption, the items in the database are assigned a normal existential probability distribution.

Sampling over the itemsets was performed using a modified hub-averaging of [9] and run on machine with specifications: Intel Pentium 4, 6GBHDD,2GB RAM .

The datasets were are taken from KDD datasets: connect, mushroom, psumb, psumb_star, and T10I4D100K all used as shown in the figure below. The running time goes exponential as computation of hub and authority is recursive is shown below.

Table1: datasets and the running time

The following graph shows that the complexity of sampling as its recursive for each hub to select the top sample rate percent, grows exponentially to sample data.

Dataset	Sample Ratio			
	0.03	0.05	0.1	0.2
mushroom	0.79	1.36	2.43	4.53
pumsb	33.17	55.6	107.25	197.21
pumsb_star	53.2	86.87	165.22	308.86
connect	59.7	98.99	192.37	357.47

Comparison of the FFTprobFPgrowth with ProFPGrowth on the different database sizes our approach is giving better thresholds. The following figure shows the run times on different database sizes.

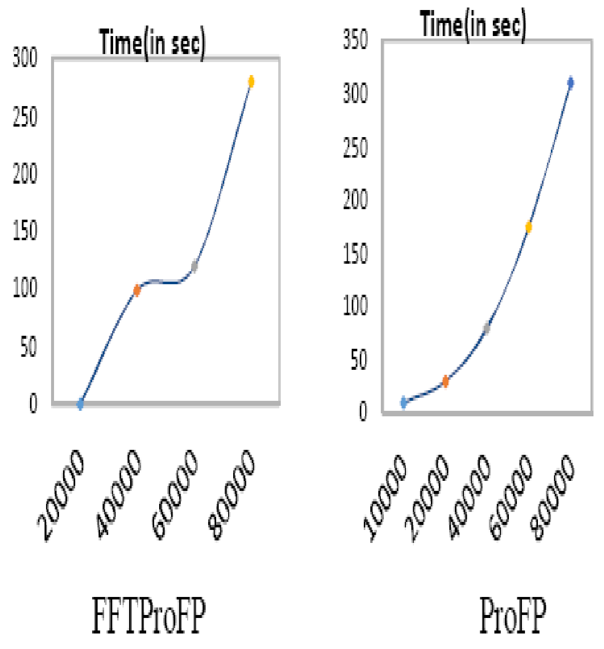


Fig 2: Run times with database sizes

As we can see from Fig1, and [9] B. Chandra & etal conclusion, the for a sampled data size about 0.02% can represent almost the whole dataset. Then as we have mentioned above in section 3.2, the probabilistic support computation can be done with an efficient way. Hence it is a new approach on uncertain database based to find frequent patterns using hub-averaging algorithm.

To compare its performance we have took sampling technique applied and shown in the table below. As we can see that based on the tight statistics bound for the sample size K, the retrieved itemsets with their corresponding time cost are shown.

Both time and number of PFIs are averaged from 10 runs for each K with a corresponding minsup threshold.

Table2: dataset and running time using random r.

Size of K	Support	connect		psumb		mushroom	
		Time(sec)	PFIs	Time(sec)	PFIs	Time(sec)	PFIs
2	97.0%	1.43	15	1.99	9	0.05	1
	75.0%	1.67	1427	2.19	1598	0.09	7
	45.0%	2.48	12422	3.9	8847	0.1	73
5	97.0%	2.09	23	2.57	12	0.08	3
	75.0%	2.55	25183	3.06	27243	0.11	21
	45.0%	8.4	200787	42.89	3525477	0.16	169
10	97.0%	2.51	124	3.21	11	0.12	7
	75.0%	3.18	52631	3.87	25928	0.15	31
	45.0%	52.6	3247984	56.01	5946376	0.16	221

As sample size iterator, K, increases there is an increase in confidence interval and a decrease in estimation error. However much of time taken by generating random numbers to select samples which resulted in not really clear boundary between the number of PFIs and time taken as this depends on type of the input database.

The accuracy of our algorithm is computed by considering frequent itemsets, maximal frequent itemsets and closed frequent itemsets in both not sampled data and sampled data. Since the rules can be generated from possible combinations of frequent itemsets found, the presence or absence of the itemset depicts the error more crucial than computing margin error of the two output itemsets. Inclusion of all the above three types of frequent itemsets is calculated as follows.

$$err(FI) = 1 - \frac{\sum_{i=1}^{|FI|} O(FI_i) \notin S(FI)}{|FI|} \qquad (8)$$

$$err(M) = 1 - \frac{\sum_{i=1}^{|M|} O(M_i) \notin S(M)}{|M|} \qquad (9)$$

$$err(Cl) = 1 - \frac{\sum_{i=1}^{|Cl|} O(Cl_i) \notin S(Cl)}{|Cl|} \qquad (10)$$

The above errors are calculated for each sample ratio chosen from 3% to 30% of the input dataset with the minimum support 6% to 8%. We also took various sizes of datasets, 50,000 to 990,000, to get the weighted average error of the above errors.

The Table3 below shows the weighted average error is subtracted from the original dataset to get the accuracy in each sample ration discussed above.

Table3: Accuracy of samples for input datasets

Dataset	Sample Ratio				
	0.03	0.05	0.1	0.2	0.3
mushroom	94.1	97.05	97.05	100	100
pumsb	93	92.86	92.86	97.62	100
connect	94.8	93.5	97.2	100	100
T10I4D100K	91.7	91.7	92.3	96.14	97.3

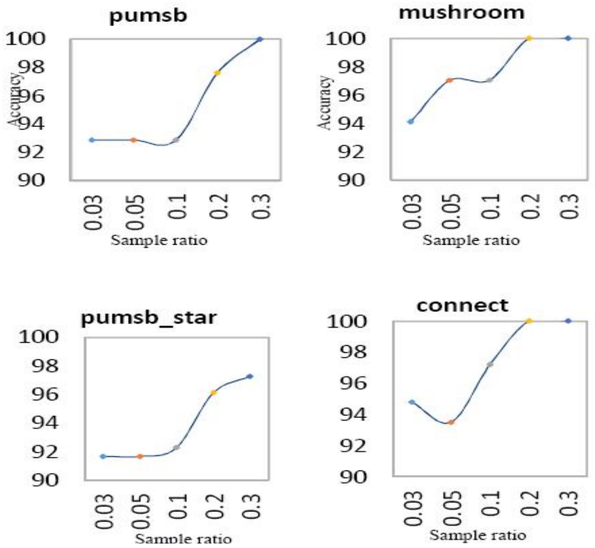


Fig2: Hub-averaging sampling for datasets

From Table3 and Fig(2) above we can see that the weighted average error is minimal, about 0.4 which in turn has accuracy about 96% for 30% dataset which is shown below on the figure.

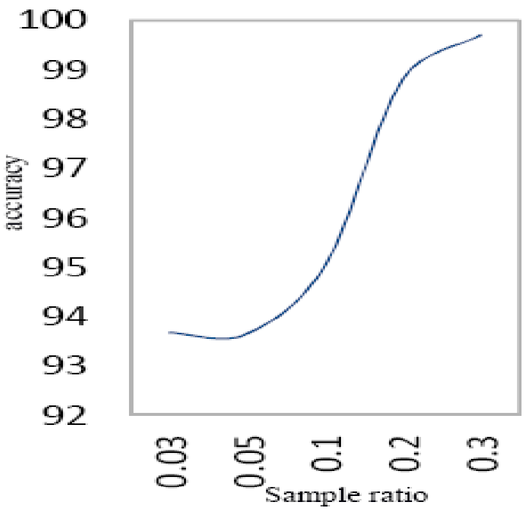


Fig 3: Average sample ratio versus accuracy

In conclusion, modified hub-averaging sampling technique for uncertain data outperforms the random sampler and can achieve more than 96% for dense uncertain datasets and 95% for sparse uncertain datasets by taking weighted hub-averaging of 0.3% of the total database size.

References

[1]. C. C. Aggarwal, Y. Li, J. Wang, and J. Wang. Frequent pattern mining with uncertain data. In Proc. of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009.

[2]. Chun K. Chui, Ben Kao, and Edward Hung. Mining frequent itemsets from uncertain data. In Zhi-Hua Zhou, Hang Li, and Qiang Yang, editors, PAKDD, volume 4426 of Lecture Notes in Computer Science, pages 47–58. Springer, 2007.

[3]. T. Bernecker, H.-P. Kriegel, M. Renz, F. Verhein, A. Züfle, Probabilistic frequent itemset mining in uncertain databases, in: Proc. of the 15th ACM SIGKDD Int. Conf. on knowledge discovery and data mining, KDD, 2009, pp. 119–127.

[4]. T. Bernecker, H-P. Kriegel, M. Renz, F. Verhein, A. Züfle, Probabilistic frequent pattern growth for itemset mining in uncertain databases, in: Proc. of the 24th Int. Conf. on Scientific and Statistical Database Management, SSDBM, 2012, pp. 38–55.

[5]. Chun K. Chui, Ben Kao, and Edward Hung. Mining frequent itemsets from uncertain data. In Zhi-Hua Zhou, Hang Li, and Qiang Yang, editors, PAKDD, volume 4426 of Lecture Notes in Computer Science, pages 47–58. Springer, 2007.

[6]. H. Toivonen. Sampling large databases for association rules. F'roc. of the Int'1 Conf. on Very Large DataBases (VLDB), 1996.

[7]. T Calders, C Garboni, B Goethals, Efficient Pattern Mining of Uncertain Data with Sampling, Advances in Knowledge Discovery and Data Mining,480-487,Springer, 2010.

[8]. E. Oran Brigham, The Fast Fourier Transform ans Its Applications, Signal Processing Series, Prentice-Hall, 1988.

[9]. B. Chandra, S. Bhaskar, ―A new approach for generating efficient sample from market basketdata, Expert Systems with Applications(38), Elsevier, 2011, pp. 1321–1325