# A Sampling Based Algorithm for Finding Association Rules from Uncertain Data

Zhu Qian, Pan Donghua, and Yang Guangfei

Institute Systems Engineering, Dalian University of Technology,
Dalian, China
zhuqianyx@gmail.com, gyise@dlut.edu.cn, yangguangfei@gmail.com

**Abstract.** Since there are many real-life situations in which people are uncertain about the content of transactions, association rule mining with uncertain data is in demand. Most of these studies focus on the improvement of classical algorithms for frequent itemsets mining. To obtain a tradeoff between the accuracy and computation time, in this paper we introduces an efficient algorithm for finding association rules from uncertain data with sampling-SARMUT, which is based on the FAST algorithm introduced by Chen et al. Unlike FAST, SARMUT is designed for uncertain data mining. In response to the special characteristics of uncertainty, we propose a new definition of “distance” as a measure to pick representative transactions. To evaluate its performance and accuracy, a comparison against the natural extension of FAST is performed using synthetic datasets. The experimental results show that the proposed sampling algorithm SARMUT outperforms FAST algorithm, and achieves up to 97% accuracy in some cases.

**Keywords:** uncertain data, association rule mining, sampling; data mining.

## 1 Introduction

Data mining techniques are heavily used to search for relationships and information that would be “hidden” in transaction data. Association rule mining is one of the most important data mining techniques and a means of identifying relationships among sets of items. An important step in the mining process is the extraction of frequent itemsets, or sets of items that co-occur in a major fraction of the transactions [1,2].

In the databases of precise data, users definitely know whether an item is present in a transaction in the databases. However, there are many situations in which users are uncertain about the presence of some items [3]. For instance, a medical dataset may contain a table of patient records, each of which contains a set of symptoms that a patient suffers. Applying association analysis on such a dataset allows us to discover any potential correlations among the symptoms and illnesses. In many cases, symptoms of subjective observations would best be represented by probabilities that indicate their presence in the patients' tuples [4].

These real-life situations in which data are uncertain call for efficient mining algorithms from uncertain data. This problem has been studied in a limited way in [1,3,5], and a variety of pruning strategies are proposed in order to speed up the

F.L. Wang et al. (Eds.): AICI 2010, Part I, LNAI 6319, pp. 124–131, 2010.

algorithm. Agarwal, C.C., et al. [6] examined the behavior of the extensions of well known classes of deterministic algorithms and found that the extensions of the candidate generate-and-test as well as the hyper-structure based algorithms are more effective than the pattern-growth algorithms.

However, there are still no new efficient methods proposed for frequent pattern mining from uncertain data so far. Since the inclusion of probability information is that the size of the dataset would be much larger than that under the quantized binary model, which makes the mining consume more time and memory usage. One popular approach is to run mining algorithms on a small subset of the data instead of the entire database, sometimes called sampling, which is often possible to explicitly tradeoff processing speed and accuracy of results [7]. Some sampling methods work very well and significantly reduce the time for association rule mining in precise data, but not uncertain data. Based on this, we investigate sampling methods that are designed to deal with mining algorithms for uncertain datasets.

In this paper, we propose an efficient algorithm based on sampling for association rules mining in existential uncertain transactional datasets called SARMUT, which is a modification and extending of FAST (Finding Association rules from Sampled Transactions), a classic algorithm that was presented in [8]. Through experiments, we will show that SARMUT is very efficient in terms of both CPU cost and accuracy, even performs better than the natural extension of FAST.

This paper is organized as follows. The next section gives related work and background. In Section 3, we introduce our SARMUT sample algorithm for mining frequent patterns from uncertain data. Section 4 shows experimental results that compare with FAST. Finally, conclusions are presented in Section 5.

# 2 Related Work and Background

## 2.1 Frequent Itemsets Mining Algorithm

Agrawal [9] proposed the Apriori algorithm using generate-and-test strategy to find all frequent patterns from a transaction database containing precise data. Each transaction in the database consists of items which all have a 100% probability of being present in that transaction.To improve efficiency of the mining process, Han et al.[10] proposed an alternative framework, namely a tree-based framework, to find frequent patterns.

## 2.2 Uncertain Data

A key difference between precise and uncertain data is that each transaction of the latter contains items and their existential probabilities. The existential probability $P(x,t_{\mathrm{i}})$ of an item $x$ in a transaction $t_{\mathrm{i}}$ indicates the probability of $x$ being present in $t_{\mathrm{i}}$. There are many items in each of the $|D|$ transactions in a transactional database $D$ and $x$ is items in an itemset $X$. Hence, the expected support of $X$, denoted by exp$Sup(X)$ can be computed as follows:

$$\exp Sup\left(X\right)=\sum_{i=1}^{|D|}\left(\prod_{x\in t_i}P\left(x,t_i\right)\right). \tag{1}$$

### 2.3 Sampling-Based Association Rule Mining

Sampling is a powerful data reduction technique that has been applied to a variety of problems in database systems. Zaki, et al. [12] employed a simple random sample to identify association rules and Toivonen [13] uses sampling to generate candidate itemsets but still requires a full database scan.

A classical algorithm-FAST (Finding Associations from Sampled Transactions), is introduced by Chen, et al. [8]. The novel two-phase strategy to collect samples of FAST is different from previous sampling-based algorithms. In the first Phase, a large initial sample of transactions is collected and used to estimate the support of each item in the database. In the second Phase, a small final sample is obtained from the initial sample so that the final sample is as close as possible to the original dataset.

## 3 Our Proposed SARMUT Algorithm

In this section, we will describe the SARMUT algorithm in details. First, we introduce the basic algorithm and some definitions we will use.

### 3.1 Overview of the Algorithm

**Notation 1.** Denote by $D$ the original database and by $S$ a simple random sample drawn without replacement from $D$. Let $|D|$ represent the number of transactions in $D$ and $|S|$ represent the number of transactions in $S$. Let $I(D)$ be the collection of itemsets that appear in $D$; a set of items $X$ is an element of $I(D)$ if and only if the items in $X$ appear jointly in at least one transaction $t \in D$. For a set of transactions $T$, let exp$Sup(X,T)$ be the expected support (computed according to Equation (1)) of $X$ in $T$. Then the support of $X$ in $S$ is given by sup($X$,$S$)= exp$Sup(X)/|S|$.

Given a specified minimum support min$Sup$, the SARMUT algorithm proceeds as follows:

1. Extract a relatively larger simple random sample $S$ from $D$;
2. Compute sup($X$,$S$) for each 1-itemset and the averages of all transactions;
3. Using the supports and sum computed in Step 2, trim outliers from $S$ to obtain the final small sample $S0$;
4. Implement a standard association rules mining algorithm against $S0$—with minimum support min$Sup$—to obtain the final set of association rules.

In Step 1, the random sampling approach we use performs without replacement. Steps 2 is straightforward and step 4 uses a standard association rule mining algorithm such as UApriori which is an extension of set-enumeration algorithm proposed in [6].

The crux of the algorithm is in step 3. Distance is denoted by the measure of difference between two transactional sets to choose which transactions should be trimmed away, which will be described in details in the following subsection.

### 3.2 The Distance Function

Unlike FAST which is designed just for precise count data applications, our proposed SARMUT is modified to fit for the association rules mining in uncertain data. The key modifications to FAST are as follows:

1. The method of expected support computation is changed to fit for uncertain data and it uses Equation (1) to calculate the value of each frequent itemset.

2. The measure used to sort out transactions contained in the final sample is modified in SARMUT. We add another characteristic, the average of all existential probabilities in a transaction to the definition of "distance".

Since the final sample has a chance of losing some information of the original database, a good sampling algorithm is able to select a collection of transactions that is very similar to the whole database in order to produce the same frequent patterns mining result. We use "distance" as a measure that compares the difference between the sample set and the whole set. In this context, the definition of the measure "distance" is very important to the effectiveness of the algorithm. The measure "distance" based on the characteristics of transaction database should be able to reflect useful information in the whole set. Chen, et al [8] used the difference with respect to the expected supports of all 1-itemsets to define "distance". In view of the property of transactional database with uncertain data, we add another characteristic to measure the distance between the sample set and the whole set.

According to Equation (1), we can deduce that the larger $P(x,t_i)$ is, the larger exp$Sup(X)$ ($x \in X$) is. Larger existential probabilities exert more remarkable influence on the value of the expected supports. It is necessary to select typical transactions with larger existential probabilities to the sample set. Based on this conclusion, we add the average of all existential probabilities in a transaction to the "distance", denoted by $T\_dist(t_i)$, which can be computed as follows:

$$ave\left(t_i\right) = \sum_{x \in t_i} P\left(x, t_i\right) / t_i\_size \ , \tag{2}$$

$$T\_dist\left(t_i\right) = \left( ave\left(t_i\right) - \sum_{j=1 \& j \neq i}^{|Sm|} ave\left(t_j\right) \right) / \left(|\, Sm \,| - 1\right) . \tag{3}$$

Let $t_i$ *_size* be the number of items in transaction *i*, and $t_i$ represent transaction *i*. Let *S* be the initial sample set, *Sf* be the final sample set and *Sm* be the current sample which is the subset of *S*. The discrepancy of *Sm* and *S* is computed by using the *I_dist* metric:

$$I\_dist(Sm, S) = \sum_{X \in I1(S)} \left| \sup(X, Sm) - \sup(X, S) \right| . \tag{4}$$

Equation (3) and (4) represent the vertical and horizontal characteristics of the database respectively. Then we merge these two characteristics into one equation:

$$Dist(Sm, S) = I\_dist(Sm, S) + 0.01 \times i\_size \times T\_dist(t_i) . \tag{5}$$

where $t_i = \{S\text{-}Sm\}$.

### 3.3 The Trimming Method

Given a desired number of transactions in the final sample *final_size*, our goal is to choose *fianl_size* transactions to constitute the final sample in order to make the *Dist*(*Sm,S*) (|*Sm*|=|*Sf*|) minimal. The problem can also be represented as follow:

$$\text{minimize } Dist(Sm, S)\,. \tag{6}$$

We can see that this combinatorial optimization problem is NP-complete. Since the database constituted by transactions can be broken into some small parts, we can use an alternative greedy algorithm to find an approximate solution. A greedy algorithm follows the problem solving metaheuristic of making the locally optimal choice at each stage with the hope of finding the global optimum. The defect of such methods is their high calculation burden in which case the quantity of data is very large. Therefore in SARMUT, the goal is to explicitly trade off speed and accuracy.

```
Extract a larger simple random sample $S$ from $D$;
Set $Sm = S$;
Compute sup $(X,S)$ for each 1-itemset $X \in I_1$ $(S)$;
Compute the sum of all $ave(t_i)$ for each transaction $t_i$ in $Sm$;
while($|Sm|$ >$fianl\_size$)
{   Divide $Sm$ into disjoint groups of $k$ transactions each;
    for each group $G$
    {
      Compute sup$(X,Sm)$ for each item $X$ in $Sm$;
      Set $Sm = Sm - \{t^*\}$,where
            $Dist(Sm - \{t^*\},S)$=min$_{ti \in G}$ Dist($Sm - \{ t_i \},S$);
    }
}
Implement a standard association rules mining algorithm against $Sf$ —with
minimum support min$Sup$;
Obtain the final set of association rules.
```

**Fig. 1.** The basic SARMUT Algorithm

By choosing a value of *k* between 1 and |*S*|, the algorithm starts with setting *Sm*=*S*. Then divide *Sm* into groups and each group contains *k* transactions. For each group, we compute every *Dist*(*Sm,S*) where $Sm=\{S\text{-}t_i\}$ and $t_i \in \{t_1, t_2,..., t_k\}$ in the group denoted by *G*. The minimal *Dist*(*Sm,S*) will be found and the corresponding transaction $t_i$ will be trimmed from the sample *Sm*, because its removal will result in the slightest variation in Equation (5). Repeat this procedure until the size of *Sm* reaches to *final_size* given by users. The basic SARMUT algorithm is given in Fig. 1.

## 4 Performance Study

In this section, we present the performance study for SARMUT and the extension of classical algorithms to find association rules from sampled transactions-FAST. The standard association rules mining algorithm we used is UApriori proposed in paper [6]. The experiments were conducted on a machine with 3.10 GHz CPU and 4G main memory installed. The operating system is GNU/Linux.

### 4.1 Experimental Methodology

The final sampling ratios chosen are 2.5%, 5%, 7.5%, 10% and 10%. Preliminary experiments showed that a value of *k*=20 for the group size worked well in both FAST and SARMUT, and we therefore use this value throughout. We select 30% transactions of the whole dataset to obtain the initial sample for SARMUT and FAST.

Two data sets were used in the experiments-T40I10D100K and T25I15D320k, which were modified in literature [6]. For each dataset and algorithm, we run the algorithm 50 times, and then for each result we adopt the averages of the 50 observations.

**Notation 2.** Denoted by *S_T* the number of frequent itemsets that sampling-based mining algorithm produced from sample *S* and *D_T* the number of frequent itemsets that standard mining algorithm produced from the whole dataset. And let *T_T* be the number of the association rules both found by the two kinds of algorithms. We take the measure of accuracy as follows:

$$\text{Accuracy 1} = 1 - \frac{|S\_T - T\_T| + |D\_T - T\_T|}{S\_T + D\_T}, \tag{7}$$

### 4.2 Experimental Results

**Accuracy vs. Sampling Ratio**

Table 1 and 2 illustrate the accuracies of the different algorithms on the synthetic databases. As shown in the figures, SARMUT outperforms FAST in most cases. It also can be seen from the figures that as the final sampling radio increases, the algorithms get better accuracy, because large samples can reflect the whole database well and truly and they contain more useful information. The similar trend can be observed in variety of min*Sup* with respect to accuracy, for the higher min*Sup* lead much fewer patterns produced both by SARMUT and FAST with expected support ≥ min*Sup*.

**Execution Time**

Table 3 shows the sampling part execution and total execution time (sampling plus subset frequent itemsets mining) of SARMUT, FAST and UApriori on T40I10D100K database as a function of the final sampling ratio. We consider only 30% transactions of the dataset as the initiative sample in both SARMUT and FAST. Experimental results indicate that, when *final_size* decreases, more transactions should be removed from the initiative sample, and thus longer runtime for sampling part is required. Moreover, both SARMUT and FAST run much faster than the non-sampling algorithm UApriori.

**Table 1.** Accuracy 1 vs. Sampling Ratio on T40I10D100K

| *final_size* | 2.5% | | 5% | | 7.5% | | 10% | |
|---|---|---|---|---|---|---|---|---|
| min*Sup*(%) | SARMUT | FAST | SARMUT | FAST | SARMUT | FAST | SARMUT | FAST |
| 0. 4 | 0.846 | 0.824 | 0.888 | 0.871 | 0.909 | 0.886 | 0.918 | 0.901 |
| 0. 6 | 0.857 | 0.836 | 0.895 | 0.888 | 0.915 | 0.903 | 0.927 | 0.919 |
| 0. 8 | 0.865 | 0.855 | 0.903 | 0.895 | 0.921 | 0.916 | 0.930 | 0.925 |
| 1.0 | 0.871 | 0.866 | 0.911 | 0.904 | 0.929 | 0.921 | 0.936 | 0.929 |
| 1.2 | 0.883 | 0.870 | 0.915 | 0.908 | 0.933 | 0.923 | 0.943 | 0.934 |
| 1.4 | 0.892 | 0.879 | 0.930 | 0.917 | 0.938 | 0.930 | 0.947 | 0.938 |
| 1.6 | 0.913 | 0.902 | 0.945 | 0.927 | 0.947 | 0.941 | 0.955 | 0.946 |
| 1.8 | 0.939 | 0.917 | 0.950 | 0.941 | 0.958 | 0.952 | 0.970 | 0.949 |
| 2.0 | 0.948 | 0.934 | 0.968 | 0.953 | 0.973 | 0.964 | 0.975 | 0.968 |

**Table 2.** Accuracy 1 vs. Sampling Ratio on T25I15D320K

| *final_size* | 2.5% | | 5% | | 7.5% | | 10% | |
|---|---|---|---|---|---|---|---|---|
| min*Sup*(%) | SARMUT | FAST | SARMUT | FAST | SARMUT | FAST | SARMUT | FAST |
| 0.1 | 0.779 | 0.715 | 0.858 | 0.824 | 0.884 | 0.869 | 0.905 | 0.888 |
| 0.2 | 0.864 | 0.859 | 0.905 | 0.900 | 0.923 | 0.920 | 0.933 | 0.931 |
| 0.3 | 0.864 | 0.859 | 0.905 | 0.901 | 0.924 | 0.918 | 0.934 | 0.930 |
| 0.4 | 0.867 | 0.857 | 0.908 | 0.902 | 0.927 | 0.919 | 0.935 | 0.930 |
| 0.5 | 0.874 | 0.864 | 0.912 | 0.903 | 0.927 | 0.922 | 0.939 | 0.930 |
| 0.6 | 0.888 | 0.878 | 0.924 | 0.913 | 0.936 | 0.932 | 0.949 | 0.938 |
| 0.7 | 0.904 | 0.891 | 0.929 | 0.924 | 0.949 | 0.939 | 0.951 | 0.946 |
| 0.8 | 0.917 | 0.903 | 0.945 | 0.934 | 0.952 | 0.946 | 0.960 | 0.954 |
| 0.9 | 0.944 | 0.926 | 0.967 | 0.947 | 0.968 | 0.956 | 0.970 | 0.964 |

**Table 3.** Execution of FAST,SARMUT and UApriori when min*Sup*=0.2% with T40I10D100K

| *final_size* | FAST (sampling part) | SARMUT (sampling part) | UApriori on subset | FAST (total) | SARMUT (total) | UApriori |
|---|---|---|---|---|---|---|
| 2.50% | 30.02 | 40.3 | 20.28 | 50.3 | 60.58 | 351 |
| 5.00% | 27.55 | 36.91 | 28.71 | 56.26 | 65.62 | 351 |
| 7.50% | 25.07 | 33.51 | 36.55 | 62.07 | 70.51 | 351 |
| 10.00% | 22.61 | 30.10 | 45.82 | 68.43 | 75.92 | 351 |

## 5 Conclusions

In this paper we studied the problem of finding association rule from existential uncertain data. We solved this problem with the sampling method and introduced SARMUT, which is a variant of FAST algorithm. Based on the two-phase trimming strategy of FAST, we modify the algorithm to fit for uncertainty and add the vertical and horizontal characteristics of the uncertain database to the "distance" function in order to remove "outlier". The experimental results show that the sample we get is more similar to the whole database and contains more useful information so that the accuracy is better than the natural extension of FAST. Moreover, SARMUT runs much faster than the standard mining algorithm while its accuracy is still very high.

## References

1. Chui, C.-K., Kao, B., Hung, E.: Mining frequent itemsets from uncertain data. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 47–58. Springer, Heidelberg (2007)
2. Dai, X., Yiu, M.L., Mamoulis, N., Tao, Y., Vaitis, M.: Probabilistic spatial queries on existentially uncertain data. In: Bauzer Medeiros, C., Egenhofer, M.J., Bertino, E. (eds.) SSTD 2005. LNCS, vol. 3633, pp. 400–417. Springer, Heidelberg (2005)
3. Leung, C.K.-S., Mateo, M.A.F., Brajczuk, D.A.: A tree-based approach for frequent pattern mining from uncertain data. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 653–661. Springer, Heidelberg (2008)
4. Leung, C.K.-S., Carmichael, C.L., Hao, B.: Efficient mining of frequent patterns from uncertain data. In: Proc. IEEE ICDM Workshops, pp. 489–494 (2007)
5. Chui, C.-K., Kao, B.: A decremental approach for mining frequent itemsets from uncertain data. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 64–75. Springer, Heidelberg (2008)
6. Aggarwal, C.C., Li, Y., Wang, J.: Frequent pattern mining with uncertain data. In: Proc. KDD, pp. 29–37 (2009)
7. Brönnimann, H., Chen, B., Dash, M., Haas, P., Qiao, Y., Scheuerman, P.: Efficient data-reduction methods for online association rule discovery. In: Data Mining: Next Generation Challenges and Future Directions, pp. 190–208. AAAI Press, Menlo Park
8. Chen, B., Haas, P., Scheuermann, P.: A new two-phase sampling based algorithm for discovering association rule. In: Proc. 8th ACM SIGKDD, Edmonton, Alberta, Canada, July 23-26 (2002)
9. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proc. VLDB, pp. 487–499 (1994)
10. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proc. ACM SIGMOD, pp. 1–12 (2000)
11. Leung, C.K.-S., Brajczuk, D.A.: Efficient algorithms for mining constrained frequent patterns from uncertain data. In: Proc., pp. 9–18 (2009)
12. Zaki, M.J., Parthasarathy, S., Lin, W., Ogihara, M.: Evaluation of sampling for data mining of association rules. In: Proc. 3rd KDD, pp. 283–286 (1997)
13. Toivonen, H.: Sampling large databases for association rules. In: Proc. 22nd VLDB, pp. 134–145 (1996)