Technical Report CSD-TR No. 040015

Mining Association Rules with Non-uniform Privacy Concerns

Yi Xia Yirong Yang Yun Chi Richard R. Muntz
Computer Science Department, University of California
Los Angeles, CA 90095
{*xiayi,yyr,ychi,muntz*}*@cs.ucla.edu*

Abstract

Privacy concerns have become an important issue in data mining. A popular way to preserve privacy is to randomize the dataset to be mined in a systematic way and mine the randomized dataset instead. On the other hand, people usually have different privacy concerns for different attributes in data. E.g., in survey data, the sensitivity of questions varies. Appropriate use of this information can lead to more accurate data mining results. However, this information has not been fully utilized by many privacy preserving association rule mining algorithms.

In this paper, we generalize the privacy preserving association rule mining problem by allowing different attributes to have different levels of privacy, that is, using different randomization factors for values of different attributes in the randomization process. We also propose an efficient algorithm RE (Recursive Estimation) to estimate the support of itemsets under this framework. Both theoretical and empirical results show that the use of non-uniform randomization factors improves the accuracy of the support estimates, compared to the use of one single conservative randomization factor.

1 Motivation

The importance of privacy preservation in data mining has been recognized recently. Privacy preserving data mining algorithms aim at discovering accurate knowledge/patterns while avoiding actual access to sensitive individual information in data. To achieve this, a common approach is to randomize/distort the real dataset, so that the true value for a particular instance can not be inferred from its randomized counterpart with probabilities better than a pre-defined threshold, and the data mining algorithms are performed on the randomized/distorted dataset instead([2], [5], [8], [4]).

In this paper, we continue the study of privacy preserving association rule mining problem. A problem with many existing algorithms is that they treat all data attributes identically. That is, all values are randomized/distorted to the same extent in the randomization process. This is usually not ideal. In fact, data values could be of different sensitivity. For example, in a survey dataset, values of different attributes

are of different importance to people. Information such as gender and age is usually not as sensitive as income or GPA. Furthermore, different people may treat the sensitivity of the same attribute values differently. So it's not necessary to have all the values to be protected at the same level.

On the other side, privacy and accuracy are a pair of contradictory measures. The increase of privacy will incur the decrease of accuracy. By allowing people to provide more accurate information on attributes that are less sensitive to them, we may gain an improvement in the quality of the mining results, compared to the approach that forces values of different attributes to be protected equally.

In this paper, we focus on the scenario where values of different attributes can have different randomization levels, while values of the same attribute always have the same randomization level. As we will see in the following sections, the introduction of non-uniform privacy levels will significantly increase the complexity of the existing data mining algorithms, and the existing mining algorithms become either impractical or at least time and space inefficient.

Our contributions in this paper are:

1) We propose a general framework for privacy preserving association rule mining that allows attributes to be randomized using different randomization factors, based on their privacy levels.

2) We develop an efficient algorithm RE(Recursive Estimation) to mine frequent itemsets under this framework.

3) We theoretically prove that the use of non-uniform randomization factors can lead to more accurate mining results than the use of one unique conservative randomization factor. Empirical experiment results also verified our claim.

The remainder of the paper is organized as follows: Section 2 discusses the related work; Section 3 formally defines the problem of privacy preserving association rule mining under non-uniform randomization factors. A direct extension of the MASK algorithm is described in Section 4. Section 5 elaborates on our proposed algorithm RE. The bias and variance of the support estimator in the RE algorithm are discussed in Section 6. Experimental results are given in Section 7 and the paper concludes with a short summary in Section 8.

2 Related Work

The privacy preserving association rule mining problem has been discussed in [5], [6], [8]. In [5], the way to achieve privacy is to hide the items of a transaction among a large set of fake items. Its algorithm treats all items with equal privacy importance. There is no direct extension of this algorithm to items with different privacy concerns. [6] focuses on the development of a new formulation for privacy breaches and the adaption of randomization operators in [5] to the new formulation of privacy breaches.

[8] uses a relatively simple way to randomize binary valued datasets. Each value in the real dataset is preserved with probability p and flipped with probability $1 - p$. An algorithm MASK is proposed to mine association rules from the randomized dataset. Again, MASK assumes that values of all items be randomized using the same randomization factor. The direct extension of MASK to data with different privacy concerns is possible, but is not time and space efficient.

Recently, a generalized version of the MASK algorithm — EMASK is proposed in [3]. EMASK allows different randomization factors (i.e., probability of flipping) for value 0s versus 1s. This flexibility provides a means to control the size of the randomized dataset and reduce the cost of the data mining algorithm. Compared to EMASK, we generalize the MASK algorithm in another direction. That is, we allow different items to use different randomization factors. As we will briefly mention in Section 5, these two directions of generalization can be smoothly combined in our RE algorithm.

[10] talks about privacy preserving association rules mining from data vertically separated in two parties that do no trust each other. This scenario is different from ours in the sense that in our scenario, each person manages his/her own data and only reveals a randomized version of his/her data to a third party.

The privacy concerns have been considered in other data mining algorithms, such as, decision trees([2],[4]) and collaborative filtering([7]). These are out of the scope of this paper which focuses on association rule mining.

3 Problem Definition

Let $\mathcal{D}$ be a binary dataset over a set of items Ω. Each row in $\mathcal{D}$ represents a transaction. The value 1 in $\mathcal{D}$ indicates the presence of the corresponding item in a transaction; value 0 otherwise. An association rule[1] in $\mathcal{D}$ is an implication of the form $\mathcal{I} \Rightarrow \mathcal{J}$, where $\mathcal{I}, \mathcal{J}$ are subsets of Ω, and $\mathcal{I} \cap \mathcal{J} = \emptyset$. It has the meaning that the occurrence of itemset $\mathcal{I}$ in a transaction implies (to some degree) the occurrence of itemset $\mathcal{J}$ in the same transaction. The most important step in association rule mining is to discover frequent itemsets. An itemset is *frequent* if it occurs in a sufficient number of transactions. The number of transactions containing this itemset is called the *support* of the itemset. In this paper, we focus on the discovery of frequent itemsets.

To protect the individual specific values in $\mathcal{D}$, we apply a randomization process on $\mathcal{D}$, and output a randomized version $\mathcal{D}'$ of $\mathcal{D}$. The task of the privacy preserving association rule mining algorithm is to discover from the randomized dataset $\mathcal{D}'$ association rules that are likely to be true in the real dataset $\mathcal{D}$.

Definition 1 *Let v be the value of item X in a tuple of the real dataset $\mathcal{D}$. In the randomization process, we use a probability p_X to generate v's counterpart v' in the randomized dataset $\mathcal{D}'$, such that v' is equal to v with probability p_X and $1 - v$ with probability $1 - p_X$. p_X is called the* **randomization factor** *for item X. The higher the randomization factor p_X is, the more likely the original value v is preserved in $\mathcal{D}'$.* □

For simplicity, we denote $1 - p_X$ by $\overline{p_X}$ in the remainder of the paper. Following the above randomization process, it is straightforward that using randomization factor p_X for an item X has equivalent effect as that using randomization factor $\overline{p_X}$. So in the following discussion, we assume that p_X is always larger than 0.5.

As we mentioned before, people usually have different privacy concerns on different items. In our randomization process, we allow different randomization factors to be used for different items, depending on their privacy concerns. The questions are:

- How should we select appropriate randomization factors for different items that best fit people's privacy concerns? That is, neither should the randomization factors violate people's privacy concerns, nor should they be too conservative.
- How should we adjust existing association rule mining algorithms to deal with different randomization factors?

- Can we really benefit from the consideration of different randomization factors?

In this paper, we will focus on the second and third question while leaving the first one open.

4 Extending MASK for Non-uniform Randomization Factors

Let $\mathcal{D}$ be the true dataset and $\mathcal{D}'$ be a randomized version of $\mathcal{D}$. Let $\mathcal{I}$ be an itemset with K distinct items, $\mathcal{I} \subseteq \Omega$. The support of $\mathcal{I}$ is denoted by $S_{\mathcal{I}}$. To make the following discussion easier, we impose an order on the items in $\mathcal{I}$, and establish a bijection between a subset f of $\mathcal{I}$ and a K-bit binary number $\mathcal{N}$. The most significant bit of $\mathcal{N}$ takes value 1 if f contains the first item in $\mathcal{I}$; and 0 otherwise. So are for the remaining bits of $\mathcal{N}$. We will use f to refer to either the subset f itself or its corresponding binary number.

Itemset $\mathcal{I}$ may occur fully or partially in a tuple. We use the form "$f, \mathcal{I}\backslash f$" to indicate that only the subset f of $\mathcal{I}$ occurs in a tuple, but not its complement $\mathcal{I}\backslash f$. Here, the symbol "$\backslash$" represents the "set minus" operation. To avoid ambiguity, we assume that "set minus $\backslash$" has higher priority than other set operations, such as "set intersection $\cap$" and "set union $\cup$" in this paper.

Let $C_{f,\mathcal{I}\backslash f}$ be the number of tuples in $\mathcal{D}$ that only contain the subset f of $\mathcal{I}$. It's easy to see that $S_{\mathcal{I}} = C_{\mathcal{I},\emptyset}$. When $\mathcal{I}$ in the context is fixed, we will use C_f as a simplification of $C_{f,\mathcal{I}\backslash f}$. For the randomized dataset $\mathcal{D}'$, we define $S'_{\mathcal{I}}$, $C'_{f,\mathcal{I}\backslash f}$ and C'_f similarly.

Definition 2 *Let i, j be two subsets of itemset $\mathcal{I}$ and X be an item in $\mathcal{I}$. If both i and j contain X or neither of them contains X, then i and j are* **consistent** *on X. Let ω be another subset of $\mathcal{I}$. If i and j are consistent on every item of ω, then i and j are* **consistent** *on ω.* □

Let $\overrightarrow{C}$ be a vector composed of the counters $\{C_{f,\mathcal{I}\backslash f} | f \subseteq \mathcal{I}\}$. That is, $\overrightarrow{C} = [C_\emptyset \cdots C_f \cdots C_{\mathcal{I}}]^T$; similarly $\overrightarrow{C'} = [C'_\emptyset \cdots C'_f \cdots C'_{\mathcal{I}}]^T$. Let $R = \{p_X | X \in \mathcal{I}\}$ be the set of randomization factors for items in $\mathcal{I}$. According to the MASK algorithm, $\overrightarrow{C}$ and the expectation of $\overrightarrow{C'}$ have the following probabilistic relationship:

$$E[\overrightarrow{C'}] = \mathbb{P}\overrightarrow{C}. \tag{1}$$

Here, $\mathbb{P}$ is a $2^K \times 2^K$ symmetric matrix, and for $i, j \subseteq \mathcal{I}$,

$$\mathbb{P}(i,j) = \prod_{X \in \mathcal{I}} \mathcal{F}_X(p_X), \text{ with } \mathcal{F}_X(p_X) = \begin{cases} p_X, & \text{if } i \text{ is consistent with } j \text{ on X;} \\ \overline{p_X}, & \text{otherwise.} \end{cases} \tag{2}$$

We call $\mathbb{P}$ the **transition matrix** for $\mathcal{I}$ under the set of randomization factors R. It is important to note that $\mathbb{P}$ is invertible if none of the randomization factors is 0.5. (See Appendix A for details.)

The MASK algorithm approximates $E[\overrightarrow{C'}]$ in equation (1) by $\overrightarrow{C'}$ and uses the solution of this equation as the estimate for $\overrightarrow{C}$. That is,

$$\overrightarrow{C'} = \mathbb{P}\overrightarrow{C^{MASK}}. \tag{3}$$

Here, $\overrightarrow{C^{MASK}}$ represents the MASK algorithm's estimation for $\overrightarrow{C}$. And $C^{MASK}_{\mathcal{I},\emptyset}$ in $\overrightarrow{C^{MASK}}$ is an estimate for $S_{\mathcal{I}}$.

Take the support estimation process for itemset $\mathcal{I} = \{ABC\}$ as an example. The MASK algorithm scans $\mathcal{D}'$ to get all values in $\{C'_f | f \subseteq \{ABC\}\}$. Thus a total of 2^3 counters are accumulated. According to formula (3), the following 2^3 equations can be constructed:

$$\begin{pmatrix} C'_\emptyset \\ C'_A \\ \cdots \\ C'_{ABC} \end{pmatrix} = \mathbb{P} \times \begin{pmatrix} C_\emptyset^{MASK} \\ C_A^{MASK} \\ \cdots \\ C_{ABC}^{MASK} \end{pmatrix}, \quad \text{with } \mathbb{P} = \begin{pmatrix} p_A p_B p_C & \overline{p_A} p_B p_C & \cdots & \overline{p_A p_B p_C} \\ \overline{p_A} p_B p_C & p_A p_B p_C & \cdots & p_A \overline{p_B p_C} \\ \cdots & \cdots & \cdots & \cdots \\ \overline{p_A p_B p_C} & p_A \overline{p_B p_C} & \cdots & p_A p_B p_C \end{pmatrix}.$$

If the randomization factors p_A, p_B, p_C are not equal, the size of the above matrix equation cannot be reduced from exponential to linear, as when $p_A = p_B = p_C$([8]). So a direct application of the MASK algorithm to non-uniform randomization factors needs the following time and space expensive operations:

- For each itemset of size K, it needs to construct the corresponding $2^K \times 2^K$ transition matrix $\mathbb{P}$;
- For each itemset of size K, it needs to solve 2^K equations.

5 the RE Algorithm

By carefully studying the direct extension of the MASK algorithm for non-uniform randomization factors, we find that it does a lot of redundant work. We propose a new algorithm RE(Recursive Estimation) that removes the redundancy and greatly improves the time and space efficiency.

Let $S_\mathcal{I}$ be the true support of itemset $\mathcal{I}$ in $\mathcal{D}$, and $S'_\mathcal{I}$ be the support of $\mathcal{I}$ in $\mathcal{D}'$. The RE algorithm defines an estimate of $S_\mathcal{I}$ recursively as follows:

$$\begin{cases} S_\emptyset^{RE} = S'_\emptyset = |\mathcal{D}'| = |\mathcal{D}|; \\ S_\mathcal{I}^{RE} = \frac{S'_\mathcal{I} - \sum_{f \subset \mathcal{I}} \left\{ S_f^{RE} * \prod_{X \in f} (p_X - \overline{p_X}) * \prod_{X \in \mathcal{I} \setminus f} \overline{p_X} \right\}}{\prod_{X \in \mathcal{I}} (p_X - \overline{p_X})}. \end{cases} \tag{4}$$

In this formula, the support estimate $S_\mathcal{I}^{RE}$ is derived based on the support estimates of all its subsets $\{S_f^{RE} | f \subset \mathcal{I}\}$. If the mining process is conducted in a level-wise fashion (from low level to high), then at level K, the support estimates for each K-itemset's subsets are known, and the support estimate for the K-itemset can be directly computed.

In the following, we discuss some properties of the RE algorithm, and demonstrate how the RE algorithm achieves efficiency.

Let $\overrightarrow{S}$ be the vector containing the supports of $\mathcal{I}$ and all its subsets in $\mathcal{D}$, and $\overrightarrow{S'}$ the counterpart of $\overrightarrow{S}$ for $\mathcal{D}'$. Also let $\overrightarrow{S^{RE}}$ be algorithm RE's estimate for $\overrightarrow{S}$. According to the Principle of Inclusion/Exclusion in set theory, for any $i \subseteq \mathcal{I}$, we have

$$C_{i,\mathcal{I}\setminus i} = \sum_{i \subseteq j \subseteq \mathcal{I}} (-1)^{|j|-|i|} S_j, \quad \text{and} \quad S_i = \sum_{i \subseteq j \subseteq \mathcal{I}} C_{j,\mathcal{I}\setminus j}.$$

Using matrix representation, the following equations hold:

$$\overrightarrow{C} = \mathbb{T} \overrightarrow{S}; \quad \text{and} \quad \overrightarrow{S} = \mathbb{T}^{-1} \overrightarrow{C}, \tag{5}$$

with

$$\mathbb{T}(i,j) = \begin{cases} (-1)^{|j|-|i|}, & \text{if } i \subseteq j; \\ 0, & \text{otherwise,} \end{cases} \quad \text{and} \quad \mathbb{T}^{-1}(i,j) = \begin{cases} 1, & \text{if } i \subseteq j; \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Similarly,

$$\overrightarrow{C'} = \mathbb{T}\overrightarrow{S'}; \quad \text{and} \quad \overrightarrow{S'} = \mathbb{T}^{-1}\overrightarrow{C'}. \tag{7}$$

Proposition 1 *Let $\overrightarrow{S^{RE}}$ be algorithm RE's estimate for $\overrightarrow{S}$, then the following equation holds:*

$$\overrightarrow{S'} = \mathbb{T}^{-1}\mathbb{P}\mathbb{T}\overrightarrow{S^{RE}}. \tag{8}$$

Proof. Let $\mathbb{M} = \mathbb{P}\mathbb{T}$, from the definition of $\mathbb{P}$, $\mathbb{T}$ in formula (2) and (6), we get

$$\mathbb{M}(i,j) = \prod_{X \in i \cap j} (p_X - \overline{p_X}) \prod_{X \in i \cap \mathcal{I} \setminus j} \overline{p_X} \prod_{X \in \mathcal{I} \setminus i \cap j} (\overline{p_X} - p_X) \prod_{X \in \mathcal{I} \setminus i \cap \mathcal{I} \setminus j} p_X. \tag{9}$$

Let $\mathbb{N} = \mathbb{T}^{-1}\mathbb{M} = \mathbb{T}^{-1}\mathbb{P}\mathbb{T}$, then

$$\mathbb{N}(i,j) = \begin{cases} \prod_{X \in j}(p_X - \overline{p_X}) \prod_{X \in i \setminus j} \overline{p_X}, & \text{if } j \subseteq i; \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

A more detailed development for formula (9) (10) can be found in Appendix B.

From the definition of $S_{\mathcal{I}}^{RE}$ in formula (4), we get

$$S'_i = \sum_{j \subseteq i} \{ S_j^{RE} * \prod_{X \in j} (p_X - \overline{p_X}) * \prod_{X \in i \setminus j} \overline{p_X} \}.$$

Its matrix representation is exactly $\overrightarrow{S'} = \mathbb{N}\overrightarrow{S^{RE}}$, so the equation $\overrightarrow{S'} = \mathbb{T}^{-1}\mathbb{P}\mathbb{T}\overrightarrow{S^{RE}}$ holds. ∎

Theorem 1. *The support of itemset $\mathcal{I}$ estimated by the RE algorithm is the same as that by the MASK algorithm.*

Proof. Combining formula (3), (7) and (8), we get

$$\left.\begin{array}{rcl} \overrightarrow{C'} & = & \mathbb{P}\overrightarrow{C^{MASK}} \\ \overrightarrow{C'} & = & \mathbb{T}\overrightarrow{S'} \\ \overrightarrow{S'} & = & \mathbb{T}^{-1}\mathbb{P}\mathbb{T}\overrightarrow{S^{RE}} \end{array}\right\} \Rightarrow \overrightarrow{C^{MASK}} = \mathbb{T}\overrightarrow{S^{RE}}.$$

And according to the definition of $\mathbb{T}$, we get $C_{\mathcal{I},\emptyset}^{MASK} = S_{\mathcal{I}}^{RE}$. ∎

Theorem 1 and Proposition 1 provide a clue concerning how the RE algorithm can achieve efficiency. Basically, Theorem 1 says that estimating the support for itemset $\mathcal{I}$ from $\overrightarrow{C'} = \mathbb{P}\overrightarrow{C^{MASK}}$ is equivalent to estimating it from $\overrightarrow{S'} = \mathbb{N}\overrightarrow{S^{RE}}$. If we denote the matrix equation $\overrightarrow{S'} = \mathbb{N}\overrightarrow{S^{RE}}$ for an itemset f by $\mathbb{E}_f$, then computing the support estimates for itemset $\mathcal{I}$ and all its subsets in the MASK algorithm is equivalent to solving the 2^K matrix equations $\{\mathbb{E}_f | f \subseteq \mathcal{I}\}$, one for each subset. On the other hand, Proposition 1 implies

that the matrix equation $\mathbb{E}_f$ for a subset f of $\mathcal{I}$ is just a part of $\mathbb{E}_\mathcal{I}$. For example, for $f = \{A\}$ and $\mathcal{I} = \{AB\}$, $\mathbb{E}_f$ corresponds to the first two equations in $\mathbb{E}_\mathcal{I}$. That is,

$$\mathbb{E}_f : \left\{ \begin{array}{rcl} S'_\emptyset & = & S^{RE}_\emptyset \\ S'_A & = & \overline{p_A} S^{RE}_\emptyset + (p_A - \overline{p_A}) S^{RE}_A \end{array} \right. , \text{ and}$$

$$\mathbb{E}_\mathcal{I} : \left\{ \begin{array}{rcl} S'_\emptyset & = & S^{RE}_\emptyset \\ S'_A & = & \overline{p_A} S^{RE}_\emptyset + (p_A - \overline{p_A}) S^{RE}_A \\ S'_B & = & \overline{p_B} S^{RE}_\emptyset + (p_B - \overline{p_B}) S^{RE}_B \\ S'_{AB} & = & \overline{p_A}\,\overline{p_B} S^{RE}_\emptyset + (p_A - \overline{p_A})\overline{p_B} S^{RE}_A + \overline{p_A}(p_B - \overline{p_B}) S^{RE}_B + (p_A - \overline{p_A})(p_B - \overline{p_B}) S^{RE}_{AB} \end{array} \right. .$$

When solving the matrix equation for $\mathbb{E}_\mathcal{I}$, the part related to f is computed redundantly. In fact, if we view $\mathbb{E}_\mathcal{I}$ as 2^K equations, then only the equation for $S'_\mathcal{I}$ in $\mathbb{E}_\mathcal{I}$ has never appeared in any subset f's matrix equation $\mathbb{E}_f$. Imagine the redundancy in computing the support estimates for $\mathcal{I}$ and all its subsets. What the MASK algorithm does is equivalent to solving a total of $\sum_{f \subseteq \mathcal{I}} 2^{|f|} = 3^K$ equations [1], while what we really need is just 2^K equations, that's exactly what the RE algorithm does —— a reduction from 3^K to 2^K.

In a summary, with the RE algorithm, we no longer need to construct a $2^K \times 2^K$ transition matrix $\mathbb{P}$ for each itemset, and we can compute the support estimate for each itemset using just one formula instead of solving 2^K equations.

The RE algorithm can be further generalized by integrating [3]'s idea that different values of an item can have different randomization factors. The formula for the support estimate under this scenario can be developed similarly. That is, if p_X is the randomization factor for value 1 of X, and q_X is the randomization factor for value 0 of X, then the support estimate is:

$$\left\{ \begin{array}{l} S^{RE}_\emptyset = S'_\emptyset = |\mathcal{D}'| = |\mathcal{D}|; \\ S^{RE}_\mathcal{I} = \frac{S'_\mathcal{I} - \sum_{f \subset \mathcal{I}} \left\{ S^{RE}_f * \prod_{X \in f} (p_X - \overline{q_X}) * \prod_{X \in \mathcal{I} \setminus f} \overline{q_X} \right\}}{\prod_{X \in \mathcal{I}} (p_X - \overline{q_X})}. \end{array} \right.$$

It is obvious that this generalization does not increase the complexity of our algorithm.

6 Bias and Variance for the Support Estimator $S^{RE}_\mathcal{I}$

6.1 Bias

By taking expectation on both side of equation (8), we get

$$E[\overrightarrow{S'}] = \mathbb{T}^{-1}\mathbb{P}\mathbb{T}E[\overrightarrow{S^{RE}}]. \tag{11}$$

On the other hand,

$$\left. \begin{array}{rcl} E[\overrightarrow{C'}] & = & \mathbb{P}\overrightarrow{C} \\ \overrightarrow{C'} & = & \mathbb{T}\overrightarrow{S'} \\ \overrightarrow{C} & = & \mathbb{T}\overrightarrow{S} \end{array} \right\} \Rightarrow E[\overrightarrow{S'}] = \mathbb{T}^{-1}\mathbb{P}\mathbb{T}\overrightarrow{S}. \tag{12}$$

Combining equation (11) and (12), we get $E[\overrightarrow{S^{RE}}] = \overrightarrow{S}$. So $S^{RE}_\mathcal{I}$ is an unbiased estimator of $S_\mathcal{I}$.

[1] $\sum_{f \subseteq \mathcal{I}} 2^{|f|} = \sum_{j=|f|=0}^{K} \binom{K}{j} * 2^j * 1^{K-j} = (2+1)^K = 3^K$.

6.2 Variance

Since the support estimates by algorithm RE and MASK are the same, we can get the variance of $S_{\mathcal{I}}^{RE}$ by computing the variance of $C_{\mathcal{I},\emptyset}^{MASK}$. We use a method similar to [5] to compute the variance.

First, we compute the covariance of $\overrightarrow{C'}$. According to the randomization process, $\overrightarrow{C'}$ is a sum of 2^K independent vectors where each vector is decided by a multinomial distribution. [2] So,

$$\text{For any } i,j \subseteq \mathcal{I}, \quad Cov(C'_i, C'_j) = \sum_{l \subseteq \mathcal{I}} C_l[\mathbb{P}(i,l)\delta_{i=j} - \mathbb{P}(i,l)\mathbb{P}(j,l)]. \tag{13}$$

Suppose $\overrightarrow{\mathbb{P}_l}$ is the l-th column of $\mathbb{P}$, that is, $\overrightarrow{\mathbb{P}_l} = [\mathbb{P}(\emptyset,l), \ldots, \mathbb{P}(\mathcal{I},l)]^T$, and $\mathbb{D}_l = diag(\overrightarrow{\mathbb{P}_l}) - \overrightarrow{\mathbb{P}_l}\overrightarrow{\mathbb{P}_l}^T$, where $diag(\overrightarrow{\mathbb{P}_l})$ is a diagonal matrix with its (i,i)-th element equal to the i-th element of $\overrightarrow{\mathbb{P}_l}$. Then

$$Cov[\overrightarrow{C'}] = Cov([C'_\emptyset, \ldots, C'_{\mathcal{I}}]^T) = \sum_{l \subseteq \mathcal{I}} C_l \mathbb{D}_l. \tag{14}$$

Secondly, we compute the variance of $S_{\mathcal{I}}^{RE}$, which is also the variance of $C_{\mathcal{I},\emptyset}^{MASK}$. Since $\overrightarrow{C^{MASK}} = \mathbb{P}^{-1}\overrightarrow{C'}$, we get

$$Cov[\overrightarrow{C^{MASK}}] = \mathbb{P}^{-1}Cov[\overrightarrow{C'}](\mathbb{P}^{-1})^T = \sum_{l \subseteq \mathcal{I}} C_l \mathbb{P}^{-1}\mathbb{D}_l(\mathbb{P}^{-1})^T.$$

Theorem 2. *Suppose $\mathcal{D}'$ and $\mathcal{D}'_1$ are randomized datasets generated from $\mathcal{D}$ using randomization factors $R = \{p_X | X \in \mathcal{I}\}$ and $R' = \{p'_X | X \in \mathcal{I}\}$ respectively. R' is the same as R except for one item A. That is,*

$$p'_X \begin{cases} = & p_X, \quad \text{if } X \in \mathcal{I} \text{ and } X \neq A; \\ > & p_X, \quad \text{if } X = A. \end{cases}$$

Then the variance for $C_{\mathcal{I},\emptyset}^{MASK}$ from $\mathcal{D}'_1$ is no larger than that from $\mathcal{D}'$. ∎

Proof. Let $\mathbb{P}, \mathbb{V}$ and $\mathbb{P}', \mathbb{V}'$ be the transition matrix for $\mathcal{I}$ and covariance matrix for $\overrightarrow{C^{MASK}}$ under the randomization factors R and R' respectively. We first prove that Δ defined in the following is a semi-positive definite matrix.

$$\begin{aligned}
\Delta &= \mathbb{P}\mathbb{P}'\big[\mathbb{V} - \mathbb{V}'\big]\mathbb{P}'^T\mathbb{P}^T && (15)\\
&= \sum_{l \subseteq \mathcal{I}} C_l \mathbb{P}'\mathbb{D}_l\mathbb{P}'^T - \sum_{l \subseteq \mathcal{I}} C_l \mathbb{P}\mathbb{D}'_l\mathbb{P}^T && (16)\\
&= \sum_{l \subseteq \mathcal{I}} C_l \big\{\mathbb{P}'\big[diag(\overrightarrow{\mathbb{P}_l}) - \overrightarrow{\mathbb{P}_l}\overrightarrow{\mathbb{P}_l}^T\big]\mathbb{P}' - \mathbb{P}\big[diag(\overrightarrow{\mathbb{P}'_l}) - \overrightarrow{\mathbb{P}'_l}\overrightarrow{\mathbb{P}'_l}^T\big]\mathbb{P}\big\} \\
&= \sum_{l \subseteq \mathcal{I}} C_l \big\{\big[\mathbb{P}' \times diag(\overrightarrow{\mathbb{P}_l}) \times \mathbb{P}' - \mathbb{P} \times diag(\overrightarrow{\mathbb{P}'_l}) \times \mathbb{P}\big] - \big[\mathbb{P}'\overrightarrow{\mathbb{P}_l}\overrightarrow{\mathbb{P}_l}^T\mathbb{P}' - \mathbb{P}\overrightarrow{\mathbb{P}'_l}\overrightarrow{\mathbb{P}'_l}^T\mathbb{P}\big]\big\} && (17)\\
&= \sum_{l \subseteq \mathcal{I}} C_l \big\{\mathbb{P}' \times diag(\overrightarrow{\mathbb{P}_l}) \times \mathbb{P}' - \mathbb{P} \times diag(\overrightarrow{\mathbb{P}'_l}) \times \mathbb{P}\big\}. && (18)
\end{aligned}$$

[2] We can view the true dataset $\mathcal{D}$ as composed of 2^K blocks. Tuples in the same block contain the same subset of the itemset $\mathcal{I}$. For example, block $B_{l,\mathcal{I}\backslash l}$ is composed of tuples that support and only support the subset l of $\mathcal{I}$ and no items in $\mathcal{I}\backslash l$. According the randomization process, each block will contribute an amount to each element of $\overrightarrow{C'}$ following a multinomial distribution. The parameters of this multinomial distribution is exactly the l-th column in $\mathbb{P}$, that is, $\overrightarrow{\mathbb{P}_l} = [\mathbb{P}(\emptyset,l), \ldots, \mathbb{P}(\mathcal{I},l)]^T$.

The equality between (15) and (16) is based on the fact that

$$\mathbb{P}\mathbb{P}' = \mathbb{P}'\mathbb{P}, \tag{19}$$

and the equation between (17) and (18) is based on the fact that

$$\mathbb{P}'\overrightarrow{\mathbb{P}_l}\overrightarrow{\mathbb{P}_l}^T\mathbb{P}' = \mathbb{P}\overrightarrow{\mathbb{P}'_l}\overrightarrow{\mathbb{P}'_l}^T\mathbb{P}. \tag{20}$$

Appendix C and D give the proof for equation (19) and (20).

Proposition 2 *Let* $\Phi_l = \mathbb{P}' \times diag(\overrightarrow{\mathbb{P}_l}) \times \mathbb{P}' - \mathbb{P} \times diag(\overrightarrow{\mathbb{P}'_l}) \times \mathbb{P}$, *then* Φ_l *is semi-positive definite for* $l \subseteq \mathcal{I}$.

Appendix E gives the proof for Proposition 2.

Since $\Delta = \sum_{l \subseteq \mathcal{I}} C_l \Phi_l$, where C_l is non-negative, Δ is semi-positive definite.

A sufficient and essential condition for a semi-positive definite matrix is that it can be represented as the product of a matrix and its transpose. Suppose $\Delta = \mathbb{U}\mathbb{U}^T$, then

$$\begin{aligned} \mathbb{V} - \mathbb{V}' &= \mathbb{P}'^{-1}\mathbb{P}^{-1}\Delta(\mathbb{P}^T)^{-1}(\mathbb{P}'^T)^{-1} \\ &= \left[\mathbb{P}'^{-1}\mathbb{P}^{-1}\mathbb{U}\right]\left[\mathbb{P}'^{-1}\mathbb{P}^{-1}\mathbb{U}\right]^T. \end{aligned}$$

So $\mathbb{V} - \mathbb{V}'$ is also semi-positive definite. Since the diagonal elements of a semi-positive definite matrix are non-negative, the variance for $C^{MASK}_{\mathcal{I},\emptyset}$ from $\mathcal{D}'$ is larger than or equal to that from $\mathcal{D}'_1$. ∎

Theorem 2 shows that using different randomization factors for different items can reduce variance, compared to that using the most conservative randomization factor for all items. To gain some sense about how non-uniform randomization factors affect the variance of the support estimates, we did a small experiment: Let the true supports for itemset $\{XYZ\}$ and all its subsets be $\overrightarrow{S} = [10000, 2668, 3463, 957, 3489, 887, 1285, 328]^T$. Under one conservative randomization factor 0.7 for all items, the variance of the support estimate for $\{XYZ\}$ is $8.662e4$. However, if we raise the randomization factors for YZ to a higher value 0.9, then the variance of the support estimate for $\{XYZ\}$ is reduced to $5.382e3$.

7 Experimental Results

In the previous section, we have theoretically proved that increasing the randomization factors for non-sensitive items can lead to the reduction of variance for support estimates. Intuitively, the smaller the variance is, the better the mining result will be. In this section, we conduct experiments on both synthetic and empirical datasets using our RE algorithm to verify this idea.

The synthetic dataset is generated by the IBM Almaden generator [1] with parameters T=10, I=4, D=1M, and N=1K. [3] The empirical dataset we use is Microsoft's Anonymous Web Data available at UC Irvine's KDD archive [4]. This dataset contains $37,711$ web user records. Each record is a set of areas at the Microsoft

[3] The parameter's notation follows the naming convention in [1]: T represents the average tuple size; I represents the average size of the maximal potentially large itemsets; D represents the number of tuples; and N represents the number of distinct items.

[4] http://kdd.ics.uci.edu/databases/msweb/msweb.html

web site visited by a user in one week's time frame. Totally, there are 294 distinct areas. To get a relatively large dataset, we duplicate the dataset by a factor of 3, which results in an increase of the number of records to 113, 133.

To evaluate the quality of the mining results, we use the following three measures:

- FP: the ratio of the number of False Positive frequent itemsets over the number of true frequent itemsets. A false positive frequent itemset is one that is not actually frequent but mistakenly identified as frequent.
- FN: the ratio of the number of False Negative frequent itemsets over the number of true frequent itemsets. A false negative frequent itemset is one that is actually frequent but is not identified as frequent.
- DEV: the average deviation of the estimated support value from its true value among the correctly identified frequent itemsets. The following formula is used to compute the deviation of our support estimate for itemset $\mathcal{I}$ from its true support $S_{\mathcal{I}}$:

$$DEV_{\mathcal{I}} = \frac{|S_{\mathcal{I}}^{RE} - S_{\mathcal{I}}|}{S_{\mathcal{I}}}.$$

For both the synthetic dataset and the Microsoft Web Data dataset, the following three settings of randomization factors (Table 1) are used to randomize the datasets. Setting $S1$ simulates the scenario where all

Table 1: the settings of randomization factors for items in a dataset

$S1$	
p_X	percentage
0.7	1

$S2$	
p_X	percentage
0.7	0.25
0.8	0.5
0.9	0.25

$S3$	
p_X	percentage
0.7	0.1
0.8	0.1
0.9	0.8

the items have to use the same conservative randomization factor (in this case, 0.7) due to the limitation of the data mining algorithms, even though only a small fraction of the items require high privacy; Setting $S2$ raises part of the items' randomization factors to 0.8 and 0.9; Setting $S3$ corresponds to the scenario where people's privacy concerns are fully utilized. That is, only the sensitive items are randomized using conservative randomization factors. The items that are not sensitive will have as high randomization factors as possible.

We first perform our RE algorithm on the synthetic dataset under the three randomization factor settings. The support threshold is set to 0.25% of the total number of tuples. Figure 1 provides the measures FP, FN, and DEV under the three settings. We can see that as a larger fraction of randomization factors are raised, the number of False Positives and False Negatives, as well as the average estimation error decrease continuously. The errors under Setting $S3$ are significantly less than that under Setting $S1$. Table 2 gives more detailed comparison for itemsets at different levels. Here, LEVEL indicates the size of an itemset, that is, the number of items in the itemset. TRUE represents the true number of frequent itemsets. FPs, FNs are the absolute number of False Positives and False Negatives; and DEV is the average estimation error.

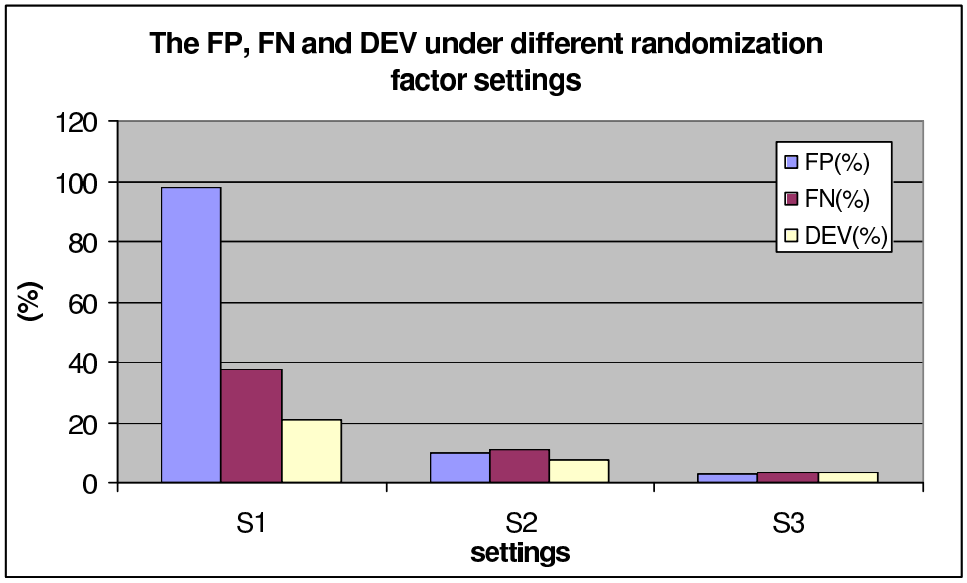

Figure 1: The quality of the mining results for the synthetic dataset under different randomization factor settings

In Table 2, the number of False Positives or False Negatives under Setting $S1, S2$ may be less than that under Setting $S3$ at some levels. This is because the randomization factors are assigned to items independently under different Settings. The fraction of items that have randomization factor 0.7 in Setting $S3$, for example, may come from the items in Setting $S2$ that have randomization factor 0.8. Also, the RE algorithm is performed on different randomized datasets generated independently for each of the three randomization factor settings. Despite of these small variations, the overall trend of decrease for all three measures from Setting $S1$ to $S2$ to $S3$ is still apparent.

For the experiment on the Microsoft Web Data dataset, we set the support threshold to 0.6% of the total number of records. Figure 2 provides the measures FP, FN and DEV under the three settings of randomization factors, while Table 3 gives the detailed comparison at each level. Basically the result shows similar patterns as that in the synthetic dataset.

8 Conclusion and Future Direction

This paper is based on the motivation that people usually have different privacy concerns for different attributes in data, and taking advantage of this to allow some attributes to be reported more accurately may lead to improvements in the quality of data mining results. In this paper, we theoretically proved the feasibility of this idea in association rule mining. By allowing different attributes to have different randomization factors, we can get more accurate estimation of itemsets' support values. We proposed an efficient algorithm called RE that significantly reduces the complexity in the association rule mining algorithms for non-uniform randomization factors.

Table 2: Detailed mining results for the synthetic dataset under different randomization factor settings

LEVEL	TRUE	S1			S2			S3		
		FPs	FNs	DEV(%)	FPs	FNs	DEV(%)	FPs	FNs	DEV(%)
1	630	30	20	9.14	6	38	9.97	6	7	4.83
2	2938	8750	476	20.48	901	276	9.04	244	128	4.57
3	2786	1158	870	22.11	65	294	7.13	40	74	2.82
4	2088	0	1086	25.67	5	282	5.63	4	46	2.10
5	1135	0	847	27.03	2	163	5.17	3	45	2.16
6	442	0	405	32.84	0	57	5.38	0	33	2.43
7	112	0	112	N/A	0	9	5.73	0	16	2.91
8	17	0	17	N/A	0	1	6.31	0	6	3.93
9	1	0	1	N/A	0	0	0.50	0	1	N/A
All	10149	9938	3834	21.07	979	1120	7.28	297	356	3.21

Table 3: Detailed mining results for Microsoft Web data

LEVEL	TRUE	S1			S2			S3		
		FPs	FNs	DEV(%)	FPs	FNs	DEV(%)	FPs	FNs	DEV(%)
1	78	39	12	13.60	11	5	10.06	10	6	7.99
2	154	431	30	26.98	50	19	12.08	27	12	6.50
3	103	142	30	35.18	18	10	10.70	6	2	6.34
4	39	14	26	67.31	10	15	9.06	5	7	4.92
All	374	626	98	27.85	89	49	11.01	48	27	6.62

In the future, we will study how the randomization factors should be selected to best fit people's different privacy concerns. We also plan to extend our study to other data mining tasks, such as privacy preserving decision tree mining.

Acknowledgment This work was supported in part by NSF grants IIS-0086116, ANI-0085773 and EAR-9817773.

References

[1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 12–15 1994.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proc. of the ACM SIGMOD Conference on Management of Data*, pages 439–450. ACM Press, May 2000.

[3] S. Agrawal, V. Krishnan, and J. Haritsa. On addressing efficiency concerns in privacy-preserving mining. In *Proc. of 9th Intl. Conf. on Database Systems for Advanced Applications (DASFAA)*, pages 113–124, 2004.

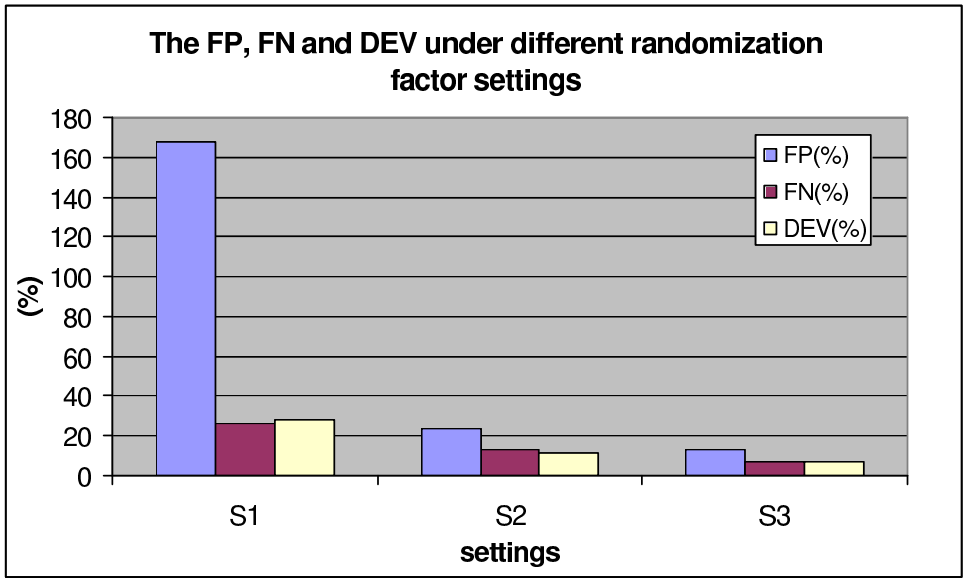

Figure 2: The quality of the mining results for Microsoft Web data under different randomization factor settings

[4] Wenliang Du and Zhijun Zhan. Using randomized response techniques for privacy-preserving data mining. In *Proc. of 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, 2003.

[5] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *Proc. of 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, 2002.

[6] Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proc. of the 22th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 211–222. ACM Press, 2003.

[7] H. Polat and W. Du. Privacy-Preserving Collaborative Filtering using Randomized Perturbation Techniques. In *Proc. of the 3th IEEE International Conference on Data Mining (ICDM)*, Melbourne, FL, November 2003.

[8] Shariq Rizvi and Jayant R. Haritsa. Maintaining data privacy in association rule mining. In *Proc. 28th Int. Conf. Very Large Data Bases, VLDB*, 2002.

[9] Gheorghe Silberberg and Szilard Pafka. A sufficient condition for the positive definiteness of the covariance matrix of a multivariate GARCH model. Technical Report CEU-Economics WP7/2001, Central European University, Economics Department, 2001.

[10] J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *Proc. of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

Appendix

A Proof that the transition matrix is invertible if no randomization factor is 0.5

This can be proved inductively. For an empty set, its transition matrix is $\big(1\big)$, so it is invertible. Suppose $\mathbb{P}'$ is the transition matrix for an itemset $\mathcal{I}$ with K items and $\mathbb{P}'$ is invertible. If we add one more item X to $\mathcal{I}$, then the corresponding transition matrix $\mathbb{P}$ can be represented as a block matrix with the following form:

$$\begin{aligned}\mathbb{P} &= \begin{pmatrix} \mathbb{P}' * p_X & \mathbb{P}' * \overline{p_X} \\ \mathbb{P}' * \overline{p_X} & \mathbb{P}' * p_X \end{pmatrix} \\ &= \begin{pmatrix} \mathbb{P}' * p_X & 0 \\ \mathbb{P}' * \overline{p_X} & \mathbb{I} \end{pmatrix} \times \begin{pmatrix} \mathbb{I} & \mathbb{I} * \frac{\overline{p_X}}{p_X} \\ 0 & \mathbb{P}' * p_X - \mathbb{P}' * \frac{\overline{p_X}^2}{p_X} \end{pmatrix}.\end{aligned}$$

Here, $\mathbb{I}$ is a $2^K \times 2^K$ identity matrix and p_X is the randomization factor for item X, $p_X \neq 0.5$. So the determinant of $\mathbb{P}$ can be computed as follows:

$$\begin{aligned} det(\mathbb{P}) &= det(\mathbb{P}' * p_X) * det(\mathbb{P}' * p_X - \mathbb{P}' * \frac{\overline{p_X}^2}{p_X}) \\ &= det(\mathbb{P}' * p_X) * det(\mathbb{P}' * \frac{(p_X + \overline{p_X})(p_X - \overline{p_X})}{p_X}) \\ &= det(\mathbb{P}') * det(\mathbb{P}') * (p_X - \overline{p_X})^{2^K}. \end{aligned}$$

Since $p_X \neq 0.5$, we get $p_X - \overline{p_X} \neq 0$. Also, from our assumption, $det(\mathbb{P}') \neq 0$, so $det(\mathbb{P}) \neq 0$, i.e., $\mathbb{P}$ is invertible.

B Proof for formula (9) and (10)

Proposition 3 *Given that i, l, u are subsets of itemset $\mathcal{I}$ and $l \subseteq u$, The following equation holds:*

$$\sum_{l \subseteq k \subseteq u} \{ \prod_{X \in i \cap k \backslash l} \mathcal{A}_X \prod_{X \in i \cap u \backslash k} \mathcal{B}_X \prod_{X \in \mathcal{I} \backslash i \cap k \backslash l} \mathcal{F}_X \prod_{X \in \mathcal{I} \backslash i \cap u \backslash k} \mathcal{G}_X \} = \prod_{X \in i \cap u \backslash l} (\mathcal{A}_X + \mathcal{B}_X) \prod_{X \in \mathcal{I} \backslash i \cap u \backslash l} (\mathcal{F}_X + \mathcal{G}_X). \quad (21)$$

Here, $\mathcal{A}_X, \mathcal{B}_X, \mathcal{F}_X$ and $\mathcal{G}_X$ are functions related to item X. □

Proof. We inductively prove the equation over u. Apparently the equation holds for $u = l$. Suppose it holds for $u' \supset l$. That is,

$$\sum_{l \subseteq k \subseteq u'} \{ \prod_{X \in i \cap k \backslash l} \mathcal{A}_X \prod_{X \in i \cap u' \backslash k} \mathcal{B}_X \prod_{X \in \mathcal{I} \backslash i \cap k \backslash l} \mathcal{F}_X \prod_{X \in \mathcal{I} \backslash i \cap u' \backslash k} \mathcal{G}_X \} = \prod_{X \in i \cap u' \backslash l} (\mathcal{A}_X + \mathcal{B}_X) \prod_{X \in \mathcal{I} \backslash i \cap u' \backslash l} (\mathcal{F}_X + \mathcal{G}_X).$$

Then for an item $A \notin u'$ such that $u = u' \cup \{A\}$, we can divide all itemsets k that are supersets of l and subsets of u into two groups, according to whether they contain A or not: $\{k | l \subseteq k \subseteq u'\}$ and $\big\{k | k = \{A\} \cup m \text{ and } l \subseteq m \subseteq u'\big\}$. So

- If $A \in i$, then

$$\begin{aligned}
&\sum_{l\subseteq k\subseteq u}\{\prod_{X\in i\cap k\backslash l}\mathcal{A}_X\prod_{X\in i\cap u\backslash k}\mathcal{B}_X\prod_{X\in\mathcal{I}\backslash i\cap k\backslash l}\mathcal{F}_X\prod_{X\in\mathcal{I}\backslash i\cap u\backslash k}\mathcal{G}_X\}\\
=&\sum_{l\subseteq k\subseteq u'}\{\prod_{X\in i\cap k\backslash l}\mathcal{A}_X\prod_{X\in i\cap u'\backslash k}\mathcal{B}_X*\mathcal{B}_A*\prod_{X\in\mathcal{I}\backslash i\cap k\backslash l}\mathcal{F}_X\prod_{X\in\mathcal{I}\backslash i\cap u'\backslash k}\mathcal{G}_X\}\\
+&\sum_{l\subseteq m\subseteq u'}\{\prod_{X\in i\cap m\backslash l}\mathcal{A}_X*\mathcal{A}_A*\prod_{X\in i\cap u'\backslash m}\mathcal{B}_X\prod_{X\in\mathcal{I}\backslash i\cap m\backslash l}\mathcal{F}_X\prod_{X\in\mathcal{I}\backslash i\cap u'\backslash m}\mathcal{G}_X\}\\
=&\ \mathcal{B}_A\prod_{X\in i\cap u'\backslash l}(\mathcal{A}_X+\mathcal{B}_X)\prod_{X\in\mathcal{I}\backslash i\cap u'\backslash l}(\mathcal{F}_X+\mathcal{G}_X)+\mathcal{A}_A\prod_{X\in i\cap u'\backslash l}(\mathcal{A}_X+\mathcal{B}_X)\prod_{X\in\mathcal{I}\backslash i\cap u'\backslash l}(\mathcal{F}_X+\mathcal{G}_X)\\
=&\prod_{X\in i\cap u\backslash l}(\mathcal{A}_X+\mathcal{B}_X)\prod_{X\in\mathcal{I}\backslash i\cap u\backslash l}(\mathcal{F}_X+\mathcal{G}_X);
\end{aligned}$$

- If $A \in \mathcal{I}\backslash i$, then

$$\begin{aligned}
&\sum_{l\subseteq k\subseteq u}\{\prod_{X\in i\cap k\backslash l}\mathcal{A}_X\prod_{X\in i\cap u\backslash k}\mathcal{B}_X\prod_{X\in\mathcal{I}\backslash i\cap k\backslash l}\mathcal{F}_X\prod_{X\in\mathcal{I}\backslash i\cap u\backslash k}\mathcal{G}_X\}\\
=&\sum_{l\subseteq k\subseteq u'}\{\prod_{X\in i\cap k\backslash l}\mathcal{A}_X\prod_{X\in i\cap u'\backslash k}\mathcal{B}_X\prod_{X\in\mathcal{I}\backslash i\cap k\backslash l}\mathcal{F}_X\prod_{X\in\mathcal{I}\backslash i\cap u'\backslash k}\mathcal{G}_X*\mathcal{G}_A\}\\
+&\sum_{l\subseteq m\subseteq u'}\{\prod_{X\in i\cap m\backslash l}\mathcal{A}_X\prod_{X\in i\cap u'\backslash m}\mathcal{B}_X\prod_{X\in\mathcal{I}\backslash i\cap m\backslash l}\mathcal{F}_X*\mathcal{F}_A*\prod_{X\in\mathcal{I}\backslash i\cap u'\backslash m}\mathcal{G}_X\}\\
=&\ \mathcal{G}_A\prod_{X\in i\cap u'\backslash l}(\mathcal{A}_X+\mathcal{B}_X)\prod_{X\in\mathcal{I}\backslash i\cap u'\backslash l}(\mathcal{F}_X+\mathcal{G}_X)+\mathcal{F}_A\prod_{X\in i\cap u'\backslash l}(\mathcal{A}_X+\mathcal{B}_X)\prod_{X\in\mathcal{I}\backslash i\cap u'\backslash l}(\mathcal{F}_X+\mathcal{G}_X)\\
=&\prod_{X\in i\cap u\backslash l}(\mathcal{A}_X+\mathcal{B}_X)\prod_{X\in\mathcal{I}\backslash i\cap u\backslash l}(\mathcal{F}_X+\mathcal{G}_X).
\end{aligned}$$

■

For the (i,j)-th element of $\mathbb{M}$, we have

$$\begin{aligned}
\mathbb{M}(i,j) &= \sum_{k\subseteq\mathcal{I}}\mathbb{P}(i,k)\mathbb{T}(k,j)=\sum_{k\subseteq j}\mathbb{P}(i,k)\mathbb{T}(k,j)\\
&= \sum_{k\subseteq j}[(-1)^{|j|-|k|}\prod_{X\in i\cap k}p_X\prod_{X\in i\cap\mathcal{I}\backslash k}\overline{p_X}\prod_{X\in\mathcal{I}\backslash i\cap k}\overline{p_X}\prod_{X\in\mathcal{I}\backslash i\cap\mathcal{I}\backslash k}p_X]\\
&= \prod_{X\in i\cap\mathcal{I}\backslash j}\overline{p_X}\prod_{X\in\mathcal{I}\backslash i\cap\mathcal{I}\backslash j}p_X\sum_{k\subseteq j}\{\prod_{X\in i\cap k}p_X\prod_{X\in i\cap j\backslash k}(-\overline{p_X})\prod_{X\in\mathcal{I}\backslash i\cap k}\overline{p_X}\prod_{X\in\mathcal{I}\backslash i\cap j\backslash k}(-p_X)\} \quad (22)\\
&= \prod_{X\in i\cap\mathcal{I}\backslash j}\overline{p_X}\prod_{X\in\mathcal{I}\backslash i\cap\mathcal{I}\backslash j}p_X\prod_{X\in i\cap j}(p_X-\overline{p_X})\prod_{X\in\mathcal{I}\backslash i\cap j}(\overline{p_X}-p_X). \quad (23)
\end{aligned}$$

The equation between (22) and (23) is based on Proposition 3.

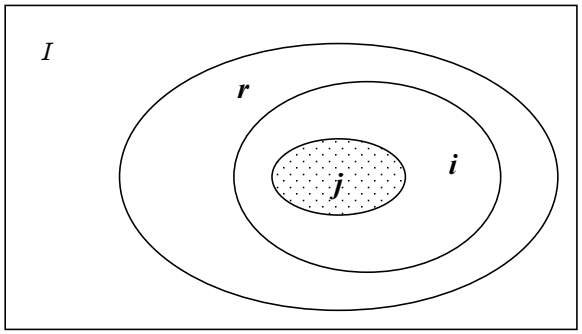

Figure 3: $j \subseteq i$

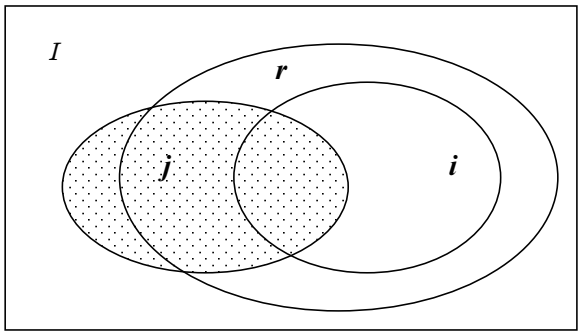

Figure 4: $j \nsubseteq i$

For the (i, j)-th element of $\mathbb{N}$, we have

$$\begin{aligned}
&\mathbb{N}(i,j) = \sum_{r\subseteq\mathcal{I}} \mathbb{T}^{-1}(i,r)\mathbb{M}(r,j) = \sum_{i\subseteq r\subseteq\mathcal{I}} \mathbb{M}(r,j) \\
= &\sum_{i\subseteq r\subseteq\mathcal{I}} \{ \prod_{X\in r\cap\mathcal{I}\setminus j} \overline{p_X} \prod_{X\in\mathcal{I}\setminus r\cap\mathcal{I}\setminus j} p_X \prod_{X\in r\cap j} (p_X - \overline{p_X}) \prod_{X\in\mathcal{I}\setminus r\cap j} (\overline{p_X} - p_X)\} \\
= &\prod_{X\in i\cap\mathcal{I}\setminus j} \overline{p_X} \prod_{X\in i\cap j} (p_X - \overline{p_X}) \sum_{i\subseteq r\subseteq\mathcal{I}} \{ \prod_{X\in r\setminus i\cap\mathcal{I}\setminus j} \overline{p_X} \prod_{X\in\mathcal{I}\setminus r\cap\mathcal{I}\setminus j} p_X \prod_{X\in r\setminus i\cap j} (p_X - \overline{p_X}) \prod_{X\in\mathcal{I}\setminus r\cap j} (\overline{p_X} - p_X)\} \quad (24) \\
= &\prod_{X\in i\cap\mathcal{I}\setminus j} \overline{p_X} \prod_{X\in i\cap j} (p_X - \overline{p_X}) \prod_{X\in\mathcal{I}\setminus i\cap\mathcal{I}\setminus j} (\overline{p_X} + p_X) \prod_{X\in\mathcal{I}\setminus i\cap j} (p_X - \overline{p_X} + \overline{p_X} - p_X) \quad (25) \\
= &\prod_{X\in i\cap\mathcal{I}\setminus j} \overline{p_X} \prod_{X\in i\cap j} (p_X - \overline{p_X}) \prod_{X\in\mathcal{I}\setminus i\cap j} (p_X - \overline{p_X} + \overline{p_X} - p_X).
\end{aligned}$$

The equation between (24) and (25) is based on Proposition 3. In the above formula, if $j \subseteq i$, then $\mathcal{I}\setminus i \cap j = \emptyset$, and $\mathbb{N}(i,j) = \prod_{X\in i\cap\mathcal{I}\setminus j} \overline{p_X} \prod_{X\in i\cap j}(p_X - \overline{p_X}) = \prod_{X\in i\setminus j} \overline{p_X} \prod_{X\in j}(p_X - \overline{p_X})$; otherwise, $\mathcal{I}\setminus i \cap j \neq \emptyset$, $\prod_{X\in\mathcal{I}\setminus i\cap j}(p_X - \overline{p_X} + \overline{p_X} - p_X) = 0$, and $\mathbb{N}(i,j) = 0$. The relationship among the subsets l, i, j and $\mathcal{I}$ under the two situations is demonstrated in Figure 3 and 4.

C Proof for equation (19)

We need to prove that $\sum_{k\subseteq\mathcal{I}} \mathbb{P}(i,k)\mathbb{P}'(k,j) = \sum_{k\subseteq\mathcal{I}} \mathbb{P}'(i,k)\mathbb{P}(k,j)$ for any $i, j \subseteq \mathcal{I}$.

According to formula (2), $\mathbb{P}(i,k)$ is a product of functions about every item's randomization factor. Based on the condition in Theorem 2, A is the only item whose randomization factor is different in $\mathbb{P}$ and $\mathbb{P}'$. If i and j are consistent on A, then $\mathbb{P}'(i,k)\mathbb{P}(k,j) = \mathbb{P}(i,k)\mathbb{P}'(k,j)$. If i and j are inconsistent on A, then $\mathbb{P}'(i,k)\mathbb{P}(k,j) = \mathbb{P}(i,m)\mathbb{P}'(m,j)$, where m is a subset of $\mathcal{I}$, and m is consistent with k on every item except A. In both cases, we can get

$$\sum_{k\subseteq\mathcal{I}} \mathbb{P}'(i,k)\mathbb{P}(k,j) = \sum_{k\subseteq\mathcal{I}} \mathbb{P}(i,k)\mathbb{P}'(k,j). \quad (26)$$

So $\mathbb{P}\mathbb{P}' = \mathbb{P}'\mathbb{P}$.

D Proof for equation (20)

$$\mathbb{P}\mathbb{P}' = \mathbb{P}'\mathbb{P} \quad \Longrightarrow \quad \mathbb{P}\overrightarrow{\mathbb{P}'_l} = \mathbb{P}'\overrightarrow{\mathbb{P}_l} \quad \Longrightarrow \quad \mathbb{P}\overrightarrow{\mathbb{P}'_l}\overrightarrow{\mathbb{P}'_l}^T\mathbb{P} = \left[\mathbb{P}\overrightarrow{\mathbb{P}'_l}\right]\left[\mathbb{P}\overrightarrow{\mathbb{P}'_l}\right]^T = \left[\mathbb{P}'\overrightarrow{\mathbb{P}_l}\right]\left[\mathbb{P}'\overrightarrow{\mathbb{P}_l}\right]^T = \mathbb{P}'\overrightarrow{\mathbb{P}_l}\overrightarrow{\mathbb{P}_l}^T\mathbb{P}'.$$

E Proof for Proposition 2

Theorem 2 assumes that A is the only item in $\mathcal{I}$ whose randomization factor is changed from p_A in $\mathbb{P}$ to p'_A in $\mathbb{P}'$, and $p'_A > p_A$. For any other item X, $p'_X = p_X$. Without the loss of generality, assume A is the last item in the ordered list of items of $\mathcal{I}$. According to the rule of matrix multiplication, the $(i,j)-$th element of $\mathbb{P} \times diag(\overrightarrow{\mathbb{P}'_l}) \times \mathbb{P}$ is $\sum_{k\subseteq\mathcal{I}} \mathbb{P}(i,k)\mathbb{P}'(k,l)\mathbb{P}(k,j)$. Using formula (2), this element can be expanded and reorganized as follows:

$$\sum_{k\subseteq\mathcal{I}} \mathbb{P}(i,k)\mathbb{P}'(k,l)\mathbb{P}(k,j) = \mathbb{H}'_A(i,j) \times \prod_{X\in\mathcal{I}, X\neq A} \mathbb{H}'_X(i,j), \text{ with}$$

$$\mathbb{H}'_A(i,j) = \begin{cases} p_A^2 p'_A + (1-p_A)^2(1-p'_A), & \text{if } i,j,l \text{ are consistent on } A; \\ p_A(1-p_A)p'_A + p_A(1-p_A)(1-p'_A), & \text{if } l \text{ is consistent with either } i \text{ or } j \text{ on } A; \\ p_A^2(1-p'_A) + (1-p_A)^2 p'_A, & \text{if } i,j \text{ are consistent on } A, \text{ but not } l. \end{cases}$$

and

$$\mathbb{H}'_X(i,j) = \begin{cases} p_X^3 + (1-p_X)^3, & \text{if } i,j,l \text{ are consistent on } X; \\ p_X^2(1-p_X) + (1-p_X)^2 p_X, & \text{otherwise.} \end{cases}$$

Similarly, the $(i,j)-$th element of $\mathbb{P}' \times diag(\overrightarrow{\mathbb{P}_l}) \times \mathbb{P}'$ can be represented as

$$\sum_{k\subseteq\mathcal{I}} \mathbb{P}'(i,k)\mathbb{P}(k,l)\mathbb{P}'(k,j) = \mathbb{H}_A(i,j) \times \prod_{X\in\mathcal{I}, X\neq A} \mathbb{H}_X(i,j), \text{ with}$$

$$\mathbb{H}_A(i,j) = \begin{cases} p'^2_A p_A + (1-p'_A)^2(1-p_A), & \text{if } i,j,l \text{ are consistent on } A; \\ p'_A(1-p'_A)p_A + p'_A(1-p'_A)(1-p_A), & \text{if } l \text{ is consistent with either } i \text{ or } j \text{ on } A; \\ p'^2_A(1-p_A) + (1-p'_A)^2 p_A, & \text{if } i,j \text{ are consistent on } A, \text{ but not } l. \end{cases}$$

and

$$\mathbb{H}_X(i,j) = \mathbb{H}'_X(i,j).$$

From above, we can view $\Phi_l = \mathbb{P}' \times diag(\overrightarrow{\mathbb{P}_l}) \times \mathbb{P}' - \mathbb{P} \times diag(\overrightarrow{\mathbb{P}'_l}) \times \mathbb{P}$ as an element-wise multiplication of K matrices. Each of the K matrices corresponds to an item in $\mathcal{I}$. The matrix corresponding to item X is $\mathbb{H}_X$ if $X \neq A$, and $\mathbb{H}_A - \mathbb{H}'_A$ otherwise.

For item A, it is easy to conclude that

$$\mathbb{H}_A(i,j) - \mathbb{H}'_A(i,j) = \begin{cases} (p'_A - p_A)(p'_A + p_A - 1) > 0, & \text{if } i,j \text{ are consistent on } A; \\ -(p'_A - p_A)(p'_A + p_A - 1) < 0, & \text{otherwise.} \end{cases}$$

Since A is the last item in $\mathcal{I}$ and it corresponds to the least significant bit in an itemset's binary number representation, $\mathbb{H}_A - \mathbb{H}'_A$ can be represented as a scalar multiplication of a constant $(p'_A - p_A)(p'_A + p_A - 1)$ and a $2^K \times 2^K$ circulant matrix as follows:

$$\mathbb{H}_A - \mathbb{H}'_A = (p'_A - p_A)(p'_A + p_A - 1) * \begin{pmatrix} 1 & -1 & 1 & \dots & -1 \\ -1 & 1 & -1 & \dots & 1 \\ 1 & -1 & 1 & \dots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 1 & -1 & \dots & 1 \end{pmatrix}.$$

The above circulant matrix's eigenvalues are 0 and 2^K, so $\mathbb{H}_A - \mathbb{H}'_A$ is semi-positive definite.

For item $X \neq A$, we reorder the items in $\mathcal{I}$ so that X becomes the first item. This reordering operation corresponds to a series of simultaneous exchanges of rows and columns in $\mathbb{H}_X$, and it leads to a matrix of the following form:

$$\begin{cases} \mathbb{Q}\mathbb{H}_X\mathbb{Q}^T = \begin{pmatrix} \mathbb{U} & \mathbb{V} \\ \mathbb{V} & \mathbb{V} \end{pmatrix} = \begin{pmatrix} \mathbb{U} - \mathbb{V} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} \mathbb{V} & \mathbb{V} \\ \mathbb{V} & \mathbb{V} \end{pmatrix}, & \text{if } X \notin l; \\ \mathbb{Q}\mathbb{H}_X\mathbb{Q}^T = \begin{pmatrix} \mathbb{V} & \mathbb{V} \\ \mathbb{V} & \mathbb{U} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & \mathbb{U} - \mathbb{V} \end{pmatrix} + \begin{pmatrix} \mathbb{V} & \mathbb{V} \\ \mathbb{V} & \mathbb{V} \end{pmatrix}, & \text{if } X \in l. \end{cases}$$

Here $\mathbb{Q}$ represents a series of row exchanges; $\mathbb{U}$ is a $2^{K-1} \times 2^{K-1}$ constant matrix with $\mathbb{U}(i,j) = p_X^3 + (1-p_X)^3$; and $\mathbb{V}$ is a $2^{K-1} \times 2^{K-1}$ constant matrix with $\mathbb{V}(i,j) = p_X^2(1-p_X) + (1-p_X)^2 p_X$. Since $\mathbb{U}(i,j) - \mathbb{V}(i,j) = [p_X^3 + (1-p_X)^3] - [p_X^2(1-p_X) + (1-p_X)^2 p_X] = (2p_X - 1)^2 \geq 0$, $\mathbb{U} - \mathbb{V}$ and $\mathbb{V}$ are semi-positive definite. So is $\mathbb{H}_X$.

Lemma 1. *The element-wise product of two symmetric semi-positive definite square matrices is a symmetric semi-positive definite matrix.[9]* ∎

According to Lemma 1, $\Phi_l = \mathbb{P}' \times diag(\overrightarrow{\mathbb{P}_l}) \times \mathbb{P}' - \mathbb{P} \times diag(\overrightarrow{\mathbb{P}'_l}) \times \mathbb{P}$ is semi-positive definite.