Tong YX, Chen L, She J. Mining frequent itemsets in correlated uncertain databases. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 30(4): 696–712 July 2015. DOI 10.1007/s11390-015-1555-9

Mining Frequent Itemsets in Correlated Uncertain Databases

Yong-Xin Tong[1] (童咏昕), *Member, CCF, ACM, IEEE*, Lei Chen[2,*] (陈 雷), *Member, ACM, IEEE* and Jieying She[2] (佘洁莹), *Student Member, IEEE*

[1] *State Key Laboratory of Software Development Environment, School of Computer Science and Engineering Beihang University, Beijing 100191, China*

[2] *Department of Computer Science and Engineering, The Hong Kong University of Science and Technology Hong Kong, China*

E-mail: yxtong@buaa.edu.cn; {leichen, jshe}@cse.ust.hk

Received January 30, 2015; revised April 9, 2015.

Abstract Recently, with the growing popularity of Internet of Things (IoT) and pervasive computing, a large amount of uncertain data, e.g., RFID data, sensor data, real-time video data, has been collected. As one of the most fundamental issues of uncertain data mining, uncertain frequent pattern mining has attracted much attention in database and data mining communities. Although there have been some solutions for uncertain frequent pattern mining, most of them assume that the data is independent, which is not true in most real-world scenarios. Therefore, current methods that are based on the independent assumption may generate inaccurate results for correlated uncertain data. In this paper, we focus on the problem of mining frequent itemsets over correlated uncertain data, where correlation can exist in any pair of uncertain data objects (transactions). We propose a novel probabilistic model, called Correlated Frequent Probability model (CFP model) to represent the probability distribution of support in a given correlated uncertain dataset. Based on the distribution of support derived from the CFP model, we observe that some probabilistic frequent itemsets are only frequent in several transactions with high positive correlation. In particular, the itemsets, which are global probabilistic frequent, have more significance in eliminating the influence of the existing noise and correlation in data. In order to reduce redundant frequent itemsets, we further propose a new type of patterns, called global probabilistic frequent itemsets, to identify itemsets that are always frequent in each group of transactions if the whole correlated uncertain database is divided into disjoint groups based on their correlation. To speed up the mining process, we also design a dynamic programming solution, as well as two pruning and bounding techniques. Extensive experiments on both real and synthetic datasets verify the effectiveness and efficiency of the proposed model and algorithms.

Keywords correlation, uncertain data, probabilistic frequent itemset

1 Introduction

In recent years, with the widespread usage of pervasive computing and collection of uncertain data in daily life, e.g., real-time video data[1], moving object search[2-4], geo-positioning services (GPS) data[5], RFID data[6], sensor data[7], mining and managing uncertain data attracts much attention of data mining and database researchers. Moreover, due to its usefulness, discovering frequent itemsets is also well studied in the uncertain environment[8-20]. Even though many efficient algorithms of uncertain frequent itemset mining have been proposed, all of them assume that different transactions are independent. Unfortunately, this assumption does not hold in many real-world scenarios. In the following, we first introduce two representative real-world scenarios where correlation plays a key role.

Scenario 1 (*Mining Correlated Sensor Data*). Sensor data are often uncertain due to noise and trans-

Regular Paper

Special Section on Data Management and Data Mining

This work is partially supported by the Hong Kong RGC Project under Grant No. N_HKUST637/13, the National Basic Research 973 Program of China under Grant No. 2014CB340303, the National Natural Science Foundation of China under Grant Nos. 61328202 and 61300031, Microsoft Research Asia Gift Grant, Google Faculty Award 2013, and Microsoft Research Asia Fellowship 2012.

[*] Corresponding Author

mission errors[7]. For instance, in a real scenario of sensor networks, due to noise, the data collected from different sensors often have big differences even though these sensors locate nearby. Thus, the correlation of locations should be considered. In particular, when the collected sensor data are massive, finding the frequent combinations (itemsets) of attribute values (such as values of temperature, humidity, and concentration of carbon dioxide) will help us to discover some underlying properties of the data.

Scenario 2 (*Mining Correlated RFID Data*). Due to the limitation of hardware and the protocols of RFID, e.g., ALOHA[21], the raw data from RFID systems are usually incomplete and inaccurate. Moreover, correlations often exist because of signal interference of different RFID readers and response conflicts of multiple tags. Consider a scenario which has multiple RFID readers in a range, where the broadcasting signals from different RFID readers interfere with each other. Thus, the data collected from the RFID readers are correlated. If we want to know which items appear together with others frequently, the influence of correlation in the uncertain data should not be ignored.

According to the above application scenarios, mining frequent itemsets has to consider the correlation existing in uncertain data. We will show an example to clarify the difference between mining frequent itemsets under the independent assumption and that under the correlated assumption.

Example 1 (*Motivation*). Fig.1 shows a sensor network which consists of six sensors to monitor the environment of the building, such as collecting temperature or humidity values. Table 1 is the correlated uncertain data generated from the sensor network. Each transaction corresponds to a record collected from a sensor and it includes several possible values of different data types, e.g., temperature or humidity.

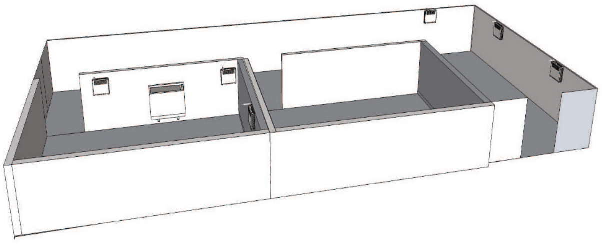


Fig.1. Sensor-based indoor monitoring system.

Table 1. Correlated Uncertain Database

TID	Transactions
T_1	$a(0.85), b(0.8), c(0.75), d(0.1), e(0.87), g(0.5)$
T_2	$a(0.9), b(0.7), e(0.6), f(0.05), g(0.45)$
T_3	$c(0.9), b(0.8), e(0.8)$
T_4	$a(0.9), b(0.1), c(0.6), d(0.3), f(0.2)$
T_5	$a(0.5), b(0.2), c(0.8), d(0.8), e(0.6)$
T_6	$a(0.7), b(0.1), d(0.7), e(0.8), f(0.1)$

Note: TID: transaction identification.

In this example, we find that three sensors locate in the room, and the other three sensors are deployed in the corridor. Also, there is an air-conditioner in the room. It is obvious that the collected data are correlated due to their spatial closeness. After cleaning the raw data, we can obtain the uncertain database shown in Table 1, where each probability in bracket denotes the likelihood that the corresponding value appears. Some pairs of sensors have correlation based on their spatial information, and each corresponding pair of transactions with correlation is assigned a non-zero Pearson's correlation coefficient[1] to evaluate the correlation between the two transactions. In this paper, we only consider linear correlation, which is actually very common in real world[22]. Thus, the correlation between any pair of transactions is represented via a correlation coefficient. For example, if the correlation coefficient between T_1 and T_2 is given as 0.7, we can generate the joint distribution of the appearance of item $\{g\}$ in T_1 and T_2 according to the correlation coefficient and Table 1. Note that $\neg T_1$ means the item $\{g\}$ does not appear in T_1, and $\neg T_2$ means that $\{g\}$ does not appear in T_2. Thus, $\Pr\{\{g\}$ appears in $T_1 \cap T_2\} = 0.7\sqrt{0.5 \times 0.5 \times 0.45 \times 0.55} + 0.5 \times 0.45 \approx 0.4$.

Similar to previous studies on mining frequent itemsets over independent uncertain data, the support of an itemset is still considered as a random variable in correlated uncertain data. However, the distribution of the support of an itemset does not follow the simple Poisson binomial distribution[18]. According to Table 2, we can find that each probability within the bracket is actually the marginal probability that the corresponding item appears in the current transaction. For example, $\Pr\{\{g\}$ appears in $T_1\} = \Pr\{\{g\}$ appears in $T_1 \cap T_2\} + \Pr\{\{g\}$ appears in $T_1 \cap \neg T_2\} = 0.4 + 0.1 = 0.5$. In particular, under the independent assumption, we will believe $\Pr\{\{g\}$ appears in $T_1 \cap T_2\} = 0.5 \times 0.45 = 0.225$. However, actually $\Pr\{\{g\}$ appears in $T_1 \cap T_2\} = 0.4$ due to correlation. Thus,

[1] $\rho_{xy} = \frac{p(xy)-p(x)p(y)}{\sqrt{p(x)(1-p(x))p(y)(1-p(y))}}$.

previous approaches under the independent assumption overestimate or underestimate the distribution of support and are not suitable for correlated uncertain data.

Table 2. Joint Distribution of Existing Probabilities of Item g

$\{g\}$	T_1	$\neg T_1$
T_2	0.4	0.05
$\neg T_2$	0.1	0.45

The above example well motivates us to work on mining frequent itemsets over correlated uncertain data. However, the problem is not trivial, as there are mainly three challenges.

Challenge 1. How to design an effective model, integrating the correlation and the probability distribution of support of an itemset? For mining frequent itemsets in correlated uncertain data, the core problem is to model the probability distribution of support of an itemset, which is the sum of a series of correlated Bernoulli random variables. As we mentioned above, it is impossible that the support of an itemset still follows the Poisson binomial distribution due to the existence of complex correlation. Thus, a direct idea is whether the existing correlation models, i.e., probabilistic graphical models, can be used. Although probabilistic graphical models utilize the factorization to reduce the complex joint distribution of random variables into the product of conditional independent sub-random variables, it is not suitable to model the probability distribution of support since support is the summation, not the product, of the sub-random variables. In other words, the complexity of inference is still exponential even if the probabilistic graphical models are used to represent the distribution of support.

Challenge 2. What types of itemsets are more significant in correlated uncertain data? Suppose the probability distribution of support in correlated uncertain data is obtained, the existing concept of probabilistic frequent itemset over independent uncertain data can be re-used on the new probability distribution. However, when the thresholds are low, a large number of redundant itemsets will be generated. Through experimental observations, we find that some itemsets are frequent in a few correlated transactions rather than the whole uncertain database. We call these frequent itemsets as *local* frequent itemsets, and others as *global* frequent itemsets. Based on these definitions, our follow-up question is which itemsets are more significant considering the relationship between local and global frequent itemsets.

Challenge 3. How to solve the problem of mining frequent itemsets over uncertain data efficiently? Efficiency is always one of the most crucial criteria of pattern mining algorithms in both deterministic data and uncertain data. Especially, efficiency and scalability are still the main challenges in the complex environment. Due to the lack of properties of Possion binomial distribution that exist under the independent assumption, existing efficient algorithms of mining probabilistic frequent itemsets do not work in correlated scenarios. A naive solution is to directly enumerate all possible worlds to compute the probability distribution of support in correlated uncertain data. However, its complexity is exponential.

In this paper, we address the above challenges and make the following contributions.

- We propose a novel probabilistic model, called Correlated Frequent Probability model (CFP model), to exactly represent the distribution of support of an itemset over correlated uncertain data.
- Due to the inherent correlation, we can divide the whole correlated uncertain database into different groups where transactions in each group are only correlated to each other. In order to find significant patterns, we propose a new type of interesting pattern, called global frequent itemset, which is not only frequent in each correlated group but also frequent among different groups.
- Based on the CFP model, we design an efficient dynamic programming algorithm together with two pruning and bounding methods, which can be used to find correlated frequent itemsets and global frequent itemsets efficiently.

The rest of the paper is organized as follows. Preliminaries and our problem formulation are introduced in Section 2. In Section 3, we present our novel model, CFP model, for capturing the correlated frequent probability of an itemset. Based on this model, several efficient algorithms, which aim to find correlated frequent itemsets and global frequent itemsets, and effective pruning strategies are proposed in Section 4. Experimental studies on both real and synthetic datasets are reported in Section 5. We review existing studies in Section 6 and conclude this paper in Section 7.

2 Problem Formulation

In this section, we review the preliminaries about uncertain frequent itemset mining in Subsection 2.1. Then, the definitions of correlated probabilistic frequent itemset and global probabilistic frequent itemset are introduced in Subsections 2.2 and 2.3, respectively.

2.1 Preliminaries

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of distinct items. We name a non-empty subset, X, of I as an itemset. For brevity, we use $X = x_1 \ldots x_n$ to denote itemset $X = \{x_1, \ldots, x_n\}$. X is an l-itemset if it has l items. Given a correlated uncertain database (CUD) including N transactions, each transaction is denoted as a tuple (tid, Y), where tid is the transaction identifier, and $Y = \{y_1(p_1), \ldots, y_m(p_m)\}$. Y contains m units. Each unit has an item y_i and a marginal probability p_i, denoting the likelihood of item y_i appearing in the tid-th transaction. $\Psi = \{\rho_{1,2}, \ldots, \rho_{N-1,N}\}$ is the set of correlation coefficients where the correlation of each pair of transactions is indicated by a Pearson's correlation coefficient to show their correlation. We assume the correlation is not the second or a higher order interaction among N transactions in this work. In other words, the N transactions only have linear correlation. Note that $\rho_{i,j} = 0$ if the i-th and the j-th transactions are independent, otherwise, $0 < \rho_{i,j} \leqslant 1$.

Definition 1 (Support and Correlated Distribution of Support). *Given a correlated uncertain database CUD, and an itemset X, the support of X, denoted as $sup(X)$, is a random variable, which represents the possible count that X appears in CUD. The correlated distribution of support of X is the probability mass function of $sup(X)$ in CUD, i.e.,* $\Pr\{sup(X) = k\}$, $k \in [0, maxv]$, *where $maxv$ is the maximum possible value of $sup(X)$.*

2.2 Correlated Probabilistic Frequent Itemset

In this subsection, we define the concepts of correlated frequent probability and correlated probabilistic frequent itemset, and then formulate the problem of mining correlated frequent itemsets.

Definition 2 (Correlated Frequent Probability). *Given a correlated uncertain database CUD which includes N transactions, a minimum support ratio minsup, and an itemset X, X's correlated frequent probability, denoted as $Pr_{cor}(X)$, is shown as follows:*

$$\begin{aligned} Pr_{cor}(X) &= \Pr\{sup(X) \geqslant N \times minsup\} \\ &= \sum_{k=N\times minsup}^{maxv} \Pr\{sup(x) = k\}, \end{aligned}$$

where $sup(X)$ follows the correlated distribution of $sup(X)$ in CUD.

Definition 3 (Correlated Probabilistic Frequent Itemsets). *Given a correlated uncertain database CUD including N transactions, a minimum support ratio minsup, and a probabilistic threshold pft, an itemset X is a correlated probabilistic frequent itemset if X's correlated frequent probability is greater than pft,*

$$Pr_{cor}(X) = \Pr\{sup(X) \geqslant N \times minsup\} > pft.$$

Table 3 summarizes the symbols we use. We further explain our above definitions and clarify the difference between mining frequent itemsets in independent uncertain data and that in correlated uncertain data via the following example.

Table 3. Summary of Notations

Notation	Description
CUD	Correlated uncertain database
T_i	The i-th transaction in CUD
ρ_{ij}	Correlated coefficient between the i-th and the j-th transactions
$minsup$	Specific minimum support threshold
pft	Specific probabilistic frequent threshold
$sup(X)$	Support count of an itemset X
$esup(X)$	Expectation of support of X
$Pr_{ind}(X)$	Independent frequent probability of X
$Pr_{cor}(X)$	Correlated frequent probability of X

Example 2 (*Independent Distribution of Support vs Correlated Distribution of Support*). Given the correlated uncertain database in Table 1, we can get two different distributions of $sup(g)$ under the independent and the correlated assumptions respectively in Table 4. If $minsup = 0.5$, $pft = 0.3$, according to Definition 4, we can compute the correlated frequent probability of $\{g\}$, which is $Pr_{cor}(g) = Pr_{cor}\{sup(2) = 2\} = 0.4 > 0.3$. Thus, $\{g\}$ is a frequent itemset. On the other hand, under the independent assumption, the independent frequent probability of $\{g\}$ is 0.225. Therefore, $\{g\}$ is not a frequent itemset. In fact, the correlation in data influences the probability distribution of support.

Table 4. Independent/Correlated Probability Distribution of $sup(g)$

$sup(g)$	Independent Probability Distribution	Correlated Probability Distribution
0	0.275	0.45
1	0.500	0.15
2	0.225	0.40

Based on Definition 3, we formulate our problem as follows.

Problem Statement 1 (Mining Probabilistic Frequent Itemsets over Correlated Uncertain Databases). *Given a correlated uncertain database CUD including N transactions, a minimum support ratio minsup, and a probabilistic threshold pft, this problem is to find all correlated probabilistic frequent itemsets.*

2.3 Global Probabilistic Frequent Itemsets

As we discussed in Subsection 2.2, the correlated distribution of support can completely capture the effect of correlation so that it can remove the inaccurate measurement of independent distribution of support. Unfortunately, similar to traditional frequent itemset mining problems, the definition of correlated probabilistic frequent itemsets may lead to a lot of redundant frequent itemsets when *minsup* and pft are low. In such case, it is hard for a user to understand the result of frequent itemsets directly. Therefore, a small set of itemsets should be refined from all the correlated probabilistic frequent itemsets so that these itemsets are more significant in correlated uncertain data. Based on the experimental results generated from real applications, we find that some itemsets only frequently appear in a few transactions with high correlation to each other even though they are correlated probabilistic frequent.

Recall example 1, in the indoor sensor network, the three sensors, T_1, T_2, and T_3, are located nearby in a room having an air-conditioner, and the other three sensors are deployed in the corridor. Although the two groups of sensors are used to monitor the temperature in a building, collected temperatures are likely to have a big gap due to the air-conditioner. If we get a frequent itemset about the temperature only within a group, the frequent itemset may be meaningless since its correlated frequent probability is likely enhanced by the correlation of the sensors in the group. Thus, a reasonable intuition is that a true frequent itemset should be frequent among all the groups. In other words, under the correlated scenario, a frequent itemset is preferred if it is globally frequent in each group, which includes some correlated transactions. Clearly, global frequent itemsets help eliminate the influence of local correlation. Another important problem is how to find these groups. With the correlated coefficients of transactions, we can partition the whole database into some disjoint groups, called correlated groups, where transactions in each group should have higher correlation. Moreover, different groups should be lowly correlated or independent. Based on the correlated groups, the global probabilistic frequent itemset is defined as follows.

Definition 4 (Global Probabilistic Frequent Itemset). *Given a correlated uncertain database CUD, which is partitioned into disjoint groups of transactions, a minimum support ratio minsup, and a probabilistic frequent threshold pft, an itemset X is a global probabilistic frequent itemset if X is a correlated probabilistic frequent itemset in each group.*

Note that the definition of global probabilistic frequent itemset can also be easily extended to be more flexible if a parameter, δ, is introduced to measure the global degree. In other words, an itemset is relaxed global probabilistic frequent if the itemset is a correlated probabilistic frequent itemset in at least δ correlated groups. That is, it does not need to be a correlated probabilistic frequent itemset in every group, but only in at least δ groups of them. Since the extension is straightforward and does not affect the overall structure of the problem and the devised algorithms, we continue to keep our definition of global probabilistic frequent itemset in each group in the following problem statement and other sections.

Problem Statement 2 (Mining Global Frequent Itemsets over Correlated Uncertain Databases). *Given a correlated uncertain database CUD, partitioned into correlated groups, a minimum support ratio minsup, and a probabilistic threshold pft, this problem is to find all global probabilistic frequent itemsets.*

3 CFP Model

In this section, we introduce the novel Correlated Frequent Probability (CFP) model. Firstly, we review several properties of support and linear correlation. Then, we show the distribution of support in two correlated transactions via the Pearson's correlation coefficient. Finally, by extending the case in two transactions, the model of the distribution of support in N correlated transactions is proposed and proven.

Support is the sum of a series of correlated random variables, each of which follows the Bernoulli distribution. Each Bernoulli random variable corresponds to a random event that the given itemset appears in the corresponding transaction. According to our linear correlation assumption, n Bernoulli random variables must satisfy the following lemma.

Lemma 1 (Property of Linear Correlation[22]). *Given a random variable B, which is the sum of n correlated Bernoulli random variables b_i, n Bernoulli random variables have no second order or higher order* *correlation if and only if,*

$$\frac{\Pr(b_1 = v_1, \dots, b_n = v_n)}{\Pr(b_1 = v_1) \times \cdots \times \Pr(b_n = v_n)} = \sum_{1 \leqslant i \leqslant j \leqslant n} \frac{\Pr(b_i = v_i, b_j = v_j)}{\Pr(b_i = v_i)\Pr(b_j = v_j)} - \frac{n(n+1)}{2} + 1,$$

where $v_k = 0$ *or* 1 *when* $k \in [1, n]$.

As shown in Section 1, existing popular models for correlated uncertain data are not suitable for calculating the probability distribution of support of an itemset. Thus, we propose a novel approach to model the probability distribution of support in correlated uncertain data. In order to explain the intuition of our model, we consider the simplest case, where the correlated uncertain database only contains two correlated transactions with a Pearson's correlation coefficient. The probability distribution of support can be represented by the recursive formula in the following theorem.

Theorem 1 (Probability Distribution of $sup(X)$ in Two Correlated Transactions). *Given an itemset* X, *two transactions,* T_1 *and* T_2, *where* X *appears with probabilities* p_1 *and* p_2, *respectively, and a correlated coefficient* $\rho_{1,2}$, *the probability distribution of* $sup_2(X)$ *is,*

$$\begin{aligned} &\Pr\{sup_2(X) = k\} \\ &= p_2 \times \Pr\{sup_1(X) = k-1\} + \\ &\quad (1-p_2) \times \Pr\{sup_1(X) = k\} + \\ &\quad \rho_{1,2}\sqrt{p_1p_2(1-p_1)(1-p_2)}C_{2,k,1}, \end{aligned}$$

where $sup_2(X) = T_1(X) + T_2(X)$ *is a random variable, which is the sum of two Bernoulli random variables* $T_1(X)$ *and* $T_2(X)$. *Additionally,* k *can only be* 0, 1, *or* 2, *so*

$$C_{2,k,1} = \begin{cases} 0, & if \;\; k < 0 \; or \; k > 2, \\ 1, & if \;\; k = 0, \\ -2, & if \;\; k = 1, \\ 1, & if \;\; k = 2. \end{cases}$$

Proof. According to the joint probability distribution of $sup_2(X)$, and the definitions of covariance and correlated coefficient, we have

$$\begin{aligned} &\Pr\{sup_2(X) = 0\} \\ &= (1-p_1)(1-p_2) + \rho_{1,2}\sqrt{p_1p_2(1-p_1)(1-p_2)} \\ &= p_2\Pr\{sup_1(X) = -1\} + \\ &\quad (1-p_2)\Pr\{sup_1(X) = 0\} + \\ &\quad \rho_{1,2}\sqrt{p_1p_2(1-p_1)(1-p_2)}, \end{aligned}$$

$$\begin{aligned} &\Pr\{sup_2(X) = 1\} \\ &= p_2(1-p_1) + p_1(1-p_2) - \\ &\quad 2\rho_{1,2}\sqrt{p_1p_2(1-p_1)(1-p_2)} \\ &= p_2\Pr\{sup_1(X) = 0\} + (1-p_2)\Pr\{sup_1(X) = 1\} - \\ &\quad 2\rho_{1,2}\sqrt{p_1p_2(1-p_1)(1-p_2)}, \\ &\Pr\{sup_2(X) = 2\} \\ &= p_1p_2 + \rho_{1,2}\sqrt{p_1p_2(1-p_1)(1-p_2)} \\ &= p_2\Pr\{sup_1(X) = 1\} + (1-p_2)\Pr\{sup_1(X) = 2\} + \\ &\quad \rho_{1,2}\sqrt{p_1p_2(1-p_1)(1-p_2)}. \end{aligned}$$

Therefore, we can induce the form of $C_{2,k,1}$ as above. □

According to Theorem 1, we can further induce a general representation for the probability distribution of $sup_n(X)$ which includes n transactions in Theorem 2 as follows.

Theorem 2 (Probability Distribution of $sup(X)$ in n Correlated Transactions). *Given an itemset* X, *a correlated uncertain database which includes* n *transactions,* $T_1, \dots, T_n$, *where* X *likely appears with probabilities* $p_1, \dots, p_n$, *respectively, and a set of correlated coefficients* $\{\rho_{1,2}, \dots, \rho_{n-1,n}\}$, *the probability distribution of* $sup_n(X)$ *is,*

$$\begin{aligned} &\Pr\{sup_n(X) = k\} \\ &= p_n\Pr\{sup_{n-1}(X) = k-1\} + (1-p_n) \times \\ &\quad \Pr\{sup_{n-1}(X) = k\} + \\ &\quad \sum_{j=1}^{n-1} \sqrt{p_np_j(1-p_n)(1-p_j)}\rho_{j,k}C_{n,k,j}, \end{aligned} \tag{1}$$

where $sup_n(X) = \sum_{i=1}^{n} T_i(X)$ *is a random variable, which is the sum of* n *Bernoulli random variables* $T_i(X)$, $i \in [1, n]$, *and* k *can be an arbitrary integer from* 1 *to* n, *thus we have*

$$\begin{aligned} &C_{n,k,j} \\ &= \begin{cases} 0, \quad if \;\; k < 0 \; or \; k > n, \\ C_{2,0,1} = C_{2,2,1} = 1, \;\; and \;\; C_{2,1,1} = -2, \\ \quad if \;\; k = 2, j = 1, \\ p_{n-2}C_{n-1,k-1,j-1} + (1-p_{n-2})C_{n-1,k,j-1}, \\ \quad if \;\; n > 2, j = k-1, \\ p_{n-1}C_{n-1,k-1,j} + (1-p_{n-1})C_{n-1,k,j}, \\ \quad if \;\; n > 2, j = 1, \dots, k-2. \end{cases} \end{aligned}$$

Proof. The theorem can be proven by mathematical induction. Firstly, we have already proven that the recursive form is correct when $n = 2$ by Theorem 1. Thus, the initial condition holds. In addition, we extend Theorem 1 to the case where $n = 3$, and the following recursive relationship can be deduced.

$$\begin{aligned}&\Pr\{sup_3(X) = k\}\\ =\ &p_3\Pr\{sup_2(X) = k - 1\} +\\ &(1 - p_3)\Pr\{sup_2(X) = k\} +\\ &\sum_{j=1}^{2}\sqrt{p_1p_2(1 - p_1)(1 - p_2)}\rho_{j,k}C_{3,k,j},\end{aligned}$$

where $sup_3(X) = \sum_{i=1}^{3} T_i(X)$ is a random variable, which is the sum of three random variables $T_i(X)$ following Bernoulli distribution, $i \in [1, 3]$. Furthermore, k can only be 0, 1, or 2. Thus,

$$\begin{aligned}&C_{3,k,j}\\ =\ &\begin{cases}0, & \text{if } \ k < 0 \text{ or } k > 3,\\ p_2C_{2,k-1,1} + (1 - p_2)C_{2,k,1}, & \text{if } j = 1,\\ p_1C_{2,k-1,1} + (1 - p_1)C_{2,k,1}, & \text{if } j = 2.\end{cases}\end{aligned}$$

Then, we assume the case of $sup_{n-1}(X)$ holds and further deduce the case of $sup_n(X)$. Since $sup_{n-1}(X)$ is right, we can obtain the following formula.

$$\begin{aligned}&\Pr\{F_n(X) = k\}\\ =\ &p_n\Pr\{F_{n-1}(X) = k - 1\} +\\ &(1 - p_n)\Pr\{F_{n-1}(X) = k\} +\\ &\sum_{j=1}^{n-1}\sqrt{p_np_j(1 - p_n)(1 - p_j)}\rho_{j,k} \times\\ &\Bigg(\sum_{C(n,k=0,j=0)}\prod_{i=1,i\neq j}^{k-1} b_i - \sum_{C(n,k=0,j=1)}\prod_{i=1,i\neq j}^{k-1} b_i -\\ &\sum_{C(n,k=1,j=0)}\prod_{i=1,i\neq j}^{k-1} b_i + \sum_{C(n,k=1,j=1)}\prod_{i=1,i\neq j}^{k-1} b_i\Bigg).\end{aligned}$$

Therefore, Theorem 2 holds. □

To sum up, we propose a novel model, the CFP model, to represent the probability distribution of support of an itemset in this section. We assume the discussed correlation is linear since it is true in most real-world cases, e.g., the distance of sensors mainly determines the correlations of each other. Thus, we utilize the correlated coefficients to measure the correlations of any two transactions and further infer the recursive form (in (1)) to obtain the probability distribution.

4 Mining Algorithms

In this section, we introduce several algorithms and pruning-and-bounding techniques for mining correlated probabilistic frequent itemsets and global probabilistic frequent itemsets, respectively.

4.1 Mining Correlated Probabilistic Frequent Itemsets

According to the CFP model, the probability distribution of support of an itemset is represented as in Theorem 2. To determine whether an itemset is a correlated probabilistic frequent itemset, we need to compute the correlated frequent probability of the itemset. A naive solution is to directly apply (1) to compute the probability recursively. Unfortunately, the naive solution is infeasible since the computational complexity of the recursive process is $O(2^N)$, where N is the size of the correlated uncertain database. In this subsection, we propose an efficient dynamic-programming-based algorithm to compute the correlated frequent probability exactly. Then, we design two pruning methods to speed up the whole mining process in Subsection 4.1.2. Finally, we introduce an Apriori framework to find all the correlated probabilistic frequent itemsets.

4.1.1 Dynamic Programming-Based Exact Algorithm

Based on (1), we can design a 3-dimensional dynamic programming algorithm to compute the frequent probability efficiently. The pseudo-code is shown in Algorithm 1.

Algorithm 1. Dynamic-Programming Algorithm (DP)
Input: a correlated uncertain database *CUD* including N transactions, an itemset X, a minimum support ratio *minsup*
Output: a correlated frequent probability of X, $Pr_{cor}(X)$

```
Scan $CUD$, pull out probability $p_1, \dots, p_n$ for $X$
Initialize $HPD_X \leftarrow \{1 - p_1, p_1, 0, \dots, 0\}$
for $i \leftarrow 2$ to $|CUD|$ do
    $PD_X \leftarrow \{0, \dots, 0\}$
    if $i = 2$ then
        $M \leftarrow 0$; $M_{0,1} \leftarrow 1$; $M_{1,1} \leftarrow -2$; $M_{2,1} \leftarrow 1$
    else
        $HM \leftarrow M$
        for $k \leftarrow 0$ to $i$ do
            for $j \leftarrow 1$ to $k - 2$ do
                $M_{k,j} \leftarrow p_{i-2} \times HM_{k-1,j-1} + (1 - p_{i-2}) \times HM_{k,j-1}$
            $M_{k,k-1} \leftarrow p_{i-1} \times HM_{k-1,j} + (1 - p_{i-1}) \times HM_{k,j}$
    for $k \leftarrow 0$ to $i$ do
        $PD_X[k] \leftarrow p_i \times HPD_X[k-1] + (1 - p_i) \times HPD_X[k]$
        for $j \leftarrow 1$ to $i - 1$ do
            $PD_X[k] \leftarrow PD_X[k] + \rho_{j,i}\sqrt{p_ip_j(1 - p_i)(1 - p_j)}M_{k,j}$
$Pr_{cor}(X) \leftarrow \sum_{m=|CUD| \times minsup}^{|CUD|} PD_X[m]$;
return $Pr_{cor}(X)$;
```

In Algorithm 1, the algorithm firstly initializes the probability distribution of $sup(X)$ as a vector according to the initial conditions of Theorem 2 in line 2. In lines 3∼15, the algorithm iteratively calculates the distribution of $sup(X)$ in the first i transactions. Note that PD_X and HPD_X are used to store the distribution of $sup(X)$ in the current iteration and that in the previous one, respectively. Similarly, M and HM denote the $C_{n,k,j}$ when n is fixed at i and $i-1$, respectively. Then during iteration, we compute $C_{n,k,j}$ of Theorem 2 in lines 4∼12, and then calculate PD_X in lines 13∼15. Finally, the correlated frequent probability of X, $Pr_{cor}(X)$, is accumulated in line 16.

Computational Complexity Analysis. According to Algorithm 1, we can know that the time and the space complexities are $O(N^3)$ and $O(N^2)$, respectively, where N is the size of the correlated uncertain database.

4.1.2 Pruning and Bounding Techniques

Based on Algorithm 1, we can exactly calculate the probability distribution of support and the correlated frequent probability of any itemset with $O(N^3)$ computational cost. We need effective pruning techniques to speed up the mining process. In this subsection, we prove that the anti-monotonic pruning method still works in the correlated uncertain data and propose a tight upper bound of the correlated frequent probability.

Lemma 2 (Anti-Monotonic Pruning). *Given a correlated uncertain database CUD and an itemset X, $\forall Y \supseteq X$ will not be a correlated probabilistic frequent itemset if X is a correlated probabilistic infrequent itemset.*

Proof. According to the definition of correlated frequent probability, we know that it is actually the sum of the probabilities whose support is greater than *minsup*. Moreover, by (1), each probability of each value of *support* depends on the corresponding marginal probability. However, the marginal probability of any superset of X cannot be greater than that of X. Thus, the correlated frequent probability of any superset of X must be lower than that of X, and hence the lemma holds. □

Lemma 3 (Upper Bound of $Pr_{cor}(X)$). *Given a correlated uncertain database CUD, a minimum support ratio minsup, and an itemset X, the correlated frequent probability of X satisfies the following relationship,*

$$Pr_{cor}(X) \leqslant \begin{cases} \frac{esup(X)}{N\times minsup}, & \exists\rho_{j,k} > 0, \\ 2^{-esup(X)\times\theta}, & \\ \quad \forall\rho_{j,k} \leqslant 0 \ and\ 2\times esup(X)-1 \leqslant \theta, & \\ e^{-esup(X)\times\theta^2/4}, & \\ \quad \forall\rho_{j,k} \leqslant 0 \ and\ 0 \leqslant \theta \leqslant 2\times esup(X)-1, & \end{cases}$$

where $esup(X)$ is the expectation of $sup(X)$, $\theta = \frac{N\times minsup - esup(X)}{esup(X)}$.

Proof. Based on Theorem 2, we know that the probability distribution of $sup_n(X)$ is,

$$\begin{aligned} &\Pr\{sup_n(X)=k\} \\ &= p_n\Pr\{sup_{n-1}(X)=k-1\} + (1-p_n)\times \\ &\quad \Pr\{sup_{n-1}(X)=k\} + \\ &\quad \sum_{j=1}^{n-1}\sqrt{p_np_j(1-p_n)(1-p_j)}\rho_{j,k}C_{n,k,j}. \end{aligned}$$

We can divide the above formula into two parts, where the first part is

$$\begin{aligned} &\Pr\{sup_n(X)=k\} \\ &= p_n\Pr\{sup_{n-1}(X)=k-1\} + (1-p_n)\times \\ &\quad \Pr\{sup_{n-1}(X)=k\}, \end{aligned}$$

which is a simple recursive formula. And the second part is

$$\sum_{j=1}^{n-1}\sqrt{p_np_j(1-p_n)(1-p_j)}\rho_{j,k}C_{n,k,j},$$

where $-1 \leqslant \rho_{j,k} \leqslant 1$.

In Lemma 3, we try to get the upper bound of $Pr_{cor}(X)$, which is actually

$$Pr_{cor}(X) = \sum_{k=minsup}^{n}\Pr\{sup_n(X)=k\}.$$

We analyze (1) according to the following two different cases.

Case 1 ($\exists\rho_{j,k} > 0$). When there is at least a $\rho_{j,k} > 0$, according to Markov inequality, we can obtain the following upper bound of $Pr_{cor}(X)$,

$$Pr_{cor}(X) \leqslant \frac{esup(X)}{N\times minsup},$$

where $esup(X)$ is the expectation of support of X.

Case 2 ($\forall \rho_{j,k} \leqslant 0$). Since all $\rho_{j,k} \leqslant 0$, we know that, for each k,

$$\begin{aligned} &\Pr\{sup_n(X) = k\} \\ \leqslant\ & p_n \Pr\{sup_{n-1}(X) = k - 1\} + (1 - p_n) \times \\ &\Pr\{sup_{n-1}(X) = k\}. \end{aligned}$$

Thus, we can obtain,

$$\begin{aligned} Pr_{cor}(X) \leqslant \sum_{k=minsup}^{n} & p_n \Pr\{sup_{n-1}(X) = k - 1\} + \\ & (1 - p_n) \times \Pr\{sup_{n-1}(X) = k\}. \end{aligned}$$

According to Chernoff inequality, we can obtain the following upper bound of $Pr_{cor}(X)$,

$$Pr_{cor}(X) \leqslant \begin{cases} 2^{-esup(X)\times\theta}, \\ \quad 2 \times esup(X) - 1 \leqslant \theta, \\ e^{-esup(X)\times\theta^2/4}, \\ \quad 0 \leqslant \theta \leqslant 2 \times esup(X) - 1, \end{cases}$$

where $\theta = \frac{N \times minsup - esup(X) - 1}{esup(X)}$.

Combining the above two cases, Lemma 3 holds. □

To sum up, the aforementioned lemmas indicate that we can prune infrequent itemsets effectively. And we apply the pruning technique in a general framework that will be presented in the next subsection.

4.1.3 *Apriori-Style Framework*

In this subsection, we give a general Apriori-style algorithm framework in Algorithm 2, which seamlessly integrates the aforementioned dynamic-programming-based efficient algorithm and the pruning-and-bounding method to discover all correlated probabilistic frequent itemsets.

In Algorithm 2, the algorithm initially fills C_1 with distinct items in line 1. In lines 4∼7, the algorithm determines which k-itemsets are correlated probabilistic frequent. In particular, the upper-bound pruning method is used to filter infrequent itemsets in line 5. For the itemsets which cannot be pruned in line 5, we have to compute the correlated frequent probabilities of these itemsets in line 6. After obtaining all correlated probabilistic frequent k-itemsets, we can generate the $(k + 1)$-size candidate set in line 8. Finally, the results are returned in line 11.

Correctness of CApriori Algorithm. Algorithm 2 is extended from the Apriori algorithm[23]. The main differences are the correlated frequent probability computation and pruning methods. According to Lemma 2 and Lemma 3, we can guarantee that the two pruning methods are safe. Algorithm 1 is also correct. Therefore, there is no false positive or false negative frequent itemset. Thus, Algorithm 2 is correct.

Algorithm 2. Correlated Apriori Algorithm Framework (CApriori)

Input: a correlated uncertain database CUD, a minimum support ratio $minsup$, and a probabilistic threshold pft

Output: a result set of all correlated probabilistic frequent itemsets

```
1  $C_1 \leftarrow$ {All distinct items in $CUD$}
2  $k = 1; j = 0$
3  while ($|C_k| \neq 0$) do
4      for each $X \in C_k$ do
5          if $UpperBound(X) \geqslant pft$ then
6              if $DP(X, CUD, minsup) \geqslant pft$ then
7                  $F_k.insert(X)$;
8      $C_{k+1} \leftarrow GenerateCandidates(F_k)$;
9      $F \leftarrow F \cup F_k$;
10     $k \leftarrow k + 1$;
11 return $F$;
```

4.2 Mining Global Probabilistic Frequent Itemsets

According to the definition of the global frequent itemsets, we next present the algorithm to find all the global frequent itemsets. Recall the motivation example in Fig.1, where we can divide the six sensors into two groups, $group_1 = \{T_1, T_2, T_3\}$ and $group_2 = \{T_4, T_5, T_6\}$, obviously. For item $\{b\}$ (which means that the temperature is cold), we present the distributions of $sup\{b\}$ in $group_1$, $group_2$ and the whole database in Fig.2. If $minsup = \frac{1}{3}, pft = 0.5$, the item $\{b\}$ is correlated frequent in $group_1$ and in the whole database, but not in $group_2$. Thus, $\{b\}$ is not a global frequent item in the database. As we mentioned in example 1, the sensors in $group_1$ report low temperature due to the air-conditioner, but we cannot say that the temperature of the building is low. This also indicates that defining and discovering global probabilistic frequent itemsets is necessary.

To capture the global frequent itemsets, our first task is to partition the database into some disjoint correlated groups. Various approaches can be used to do the partition, such as K-means, hierarchical clustering and so on. However, it is impossible to define a distance between any pair of the transactions in our database, since if two transactions are not correlated

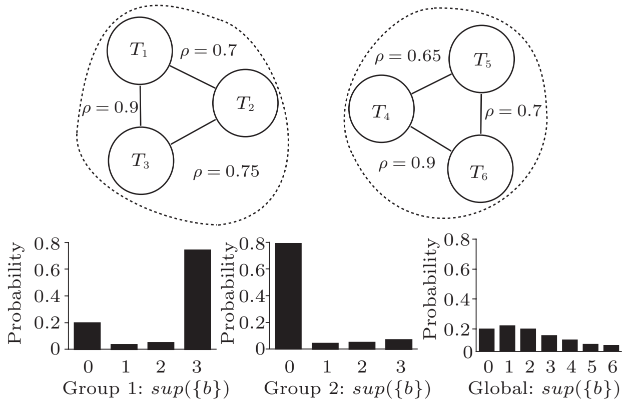


Fig.2. Partitioning CUD into correlated groups.

(e.g., independent), their distance is infinite. Thus, the distance-based partitioning or clustering approaches are not suitable for our problem. To avoid the problem of distance measurement, we construct a graph in which the nodes represent the transactions and the weighted edges denote the correlation coefficients, while uncorrelated transactions are unconnected. Then, we conduct graph clustering methods to discover correlated groups. Since we do not know how many correlated groups exist in the database, we cannot use the methods where the number of clusters must be specified. Thus, we apply the MCL algorithm[24], which is a graph clustering method and does not need to set the number of clusters, to generate all the correlated groups.

After partitioning, if we conduct the mining algorithm separately on these groups, much redundant computation will occur since some itemsets might be infrequent in some groups while frequent in others. To avoid redundant computation, we modify the Apriori framework so that these groups can share their pruning results. The pseudo-code is shown in Algorithm 3.

In Algorithm 3, the algorithm finds distinct items in line 1. Before going to the mining stage, we load the correlated groups which are generated by the offline MCL Algorithm in line 2. In lines 5∼8, we discover the correlated frequent k-itemsets in each partition with the shared candidate set. Then, the algorithm generates the global frequent k-itemsets by removing the itemsets that are not correlated frequent in at least one partition in lines 9∼13. This algorithm prevents redundant computation because in line 14, we generate the shared set of $(k+1)$-candidates only with global frequent k-itemsets, so that we avoid the problem stated above.

To clarify how Algorithm 3 works, we present a running example as shown in Fig.3.

Example 3 (*Grouped Apriori Algorithm* (*GApriori*)). Given the correlated uncertain database in Table 1, and all correlated coefficients shown in Fig.2, $minsup = \frac{1}{3}$, $pft = 0.5$, we can perform Algorithm 3 to discover the global frequent itemsets. Notice that $\{d\}$ is not frequent in group 1 and $\{b\}$ is not frequent in group 2, they are removed from F_1 and will not be used to generate C_2. The same cases go for $\{ac\}$ and $\{ce\}$ in the second step.

Algorithm 3. Grouped Apriori Algorithm Framework (GApriori)
Input: a correlated uncertain database CUD, a minimum support ratio $minsup$, and a probabilistic threshold pft
Output: a result set of all correlated probabilistic frequent itemsets

```
1  $C_1 \leftarrow$ {All distinct items in $CUD$};
2  Load the set of correlated groups of $CUD$;
3  $k = 1; j = 0$;
4  while ($|C_k| \neq 0$) do
5      for each partition $p$ do
6          for each $X \in C_k$ do
7              if ($UpperBound(X) \geqslant pft$ & $DC(X, p, minsup)$
                   $\geqslant pft$) then
8                  $F_k^p.insert(X)$;
9      $F_k \leftarrow F_k^{p_1}$; //$p_1$ means the first partition
10     for each partition $p$ other than $p_1$ do
11         for each $X \in F_k$ do
12             if $X \notin F_k^p$ then
13                 Remove $X$ from $F_k$;
14     $C_{k+1} \leftarrow GenerateCandidates(F_k)$;
15     $k \leftarrow k + 1$;
16 return $F$;
```

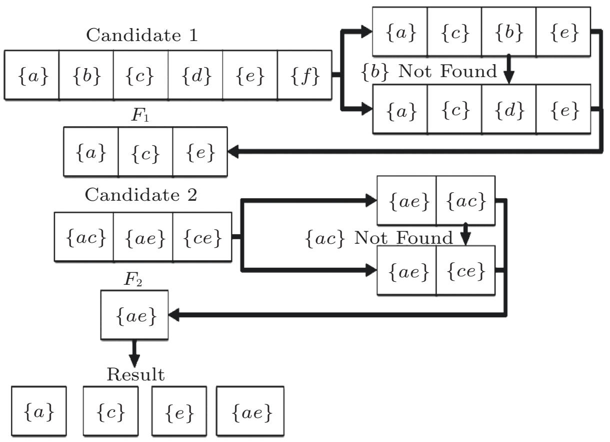


Fig.3. Example of grouped Apriori algorithm.

5 Experimental Study

In this section, we report the experimental results on tests of the efficiency of the proposed algorithms and the quality of global probabilistic frequent itemsets. In order to conduct a fair comparison, all the experiments are performed on an Intel® Core$^{\text{TM}}$ i7 3.40 GHz PC with 4 GB main memory, running on Microsoft Windows 7. Moreover, all the algorithms were implemented and compiled using Microsoft's Visual C++ 2010.

In order to test our proposed method, we use a real correlated uncertain dataset and three classical deterministic benchmarks from FIMI repository②. For the real dataset, it comes from a real sensor network monitoring project[25]. The dataset is generated from 97 sensors and contains 19 275 transactions. Each transaction records the monitored sample from one sensor, where a single item represents the possible values of a monitored event associated with a confidence. Moreover, the correlation between any pair of sensors is determined by a function of spatial information. In this dataset, 86% pairs of sensors, each of which corresponds to one transaction, have non-zero correlation. For each deterministic benchmark data, each item is assigned a probability that follows the Gaussian distribution. Assigning probability to deterministic databases to generate uncertain data is widely accepted by previous related studies[11-13,16,19]. In addition, the three benchmarks include a dense dataset, Accident, a sparse dataset, Kosarak, and a very large synthetic dataset T25I15D320k, which is used to test the scalability of the proposed approaches[11]. The characteristics and the default parameters of the datasets are shown in Table 5. For the Accident dataset, in order to retain the dense property in uncertain environment, we let the probabilities of items have high mean (0.8) and low variance (0.1). In particular, 50% of the transactions in the Accident dataset are selected randomly, and each pair of them is assigned a non-zero correlation coefficient, which is a real number generated in the interval of 0 to 1 following the uniform distribution. For the Kosarak and the T25I15D320k datasets, due to their sparse property, we set the probabilities with low mean (0.5) and high variance (0.5), 20% of the transactions are randomly chosen, and each pair of them is assigned a non-zero correlation coefficient with the same aforementioned method.

5.1 Time Efficiency Test

In this subsection, we verify the efficiency of the proposed algorithms and the pruning strategies. We compare two correlated probabilistic frequent itemset mining algorithms: CApriori and CApriori-NoPrune which does not include the pruning methods in Lemma 2 and Lemma 3, and two global probabilistic frequent itemset mining algorithms: GApriori and GApriori-NoPrune, which also does not have the two pruning strategies.

Varying minsup. Figs.4(a)~4(c) show the running time of the four competitive algorithms w.r.t. *minsup* in the Real, Accident and Kosarak datasets, respectively. When *minsup* decreases, we observe that the running time of all the algorithms goes up. In addition, GApriori is always the fastest algorithm, CApriori-NoPrune is the slowest one, and GApriori-NoPrune is faster than CApriori.

It is reasonable because GApriori and GApriori-NoPrune aim to find all the global probabilistic frequent itemsets rather than correlated probabilistic frequent itemsets. GApriori-NoPrune is slower than GApriori since it does not apply the upper-bound-based pruning. Moreover, the result that CApriori always outperforms CApriori-NoPrune makes sense as well. Since CApriori-NoPrune does not apply the aforementioned pruning, it has to spend $O(N^3)$ computational cost to check each itemset. However, CApriori filters out most infrequent itemsets with only $O(N)$ time cost. Another interesting observation is that all the algorithms spend more time under the same *minsup* in the Accident dataset than that in the Kosarak dataset. This result also makes

Table 5. Characteristics and Default Parameters of Datasets

Dataset	Number of Trans.	Number of Items	Avg. Length	*minsup*	*pft*	Correlation Coefficient (%)
Real data	19 275	126	17.0	0.5	0.9	86
Accident	340 183	468	33.8	0.6	0.9	50
Kosarak	990 002	41 270	8.1	0.5	0.9	20
T25I15D320k	320 000	994	25.0	0.5	0.9	20

② Frequent itemset mining implementations repository. http://fimi.us.ac.be, May 2015.

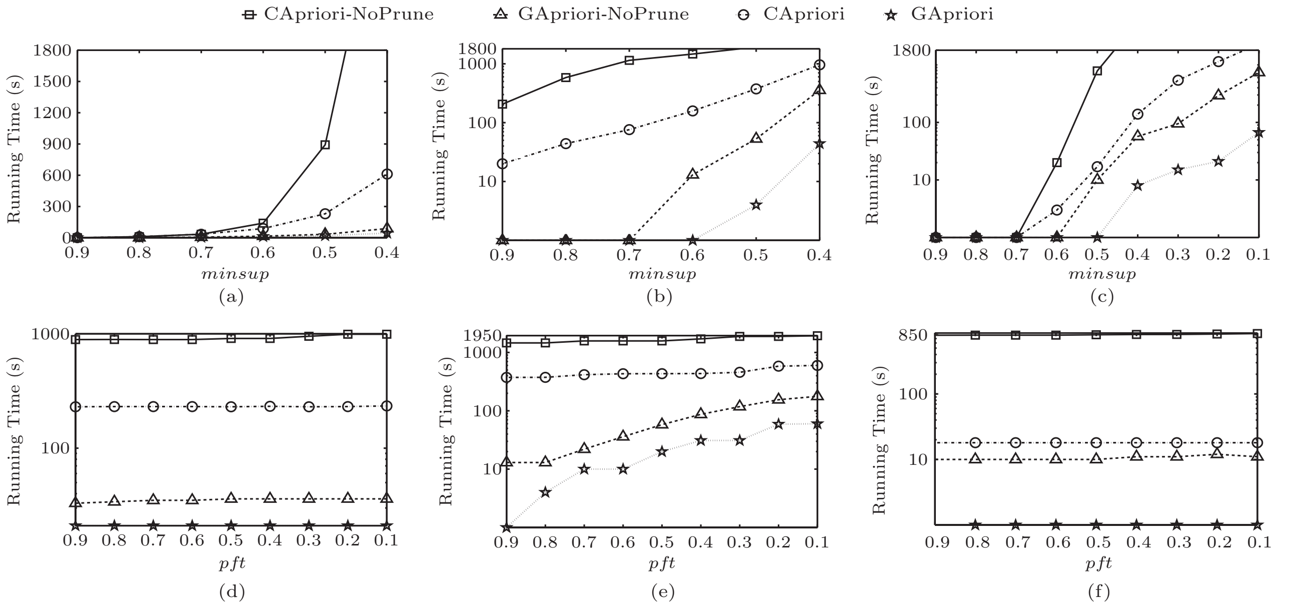


Fig.4. Test of running time. (a) Real: *minsup* vs time ($pft = 0.9$). (b) Accident: *minsup* vs time ($pft = 0.9$). (c) Kosarak: *minsup* vs time ($pft = 0.9$). (d) Real: pft vs time ($minsup = 0.5$). (e) Accident: pft vs time ($minsup = 0.6$). (f) Kosarak: pft vs time ($minsup = 0.5$).

sense because the assigned probabilities and the number of non-zero correlation coefficients in Accident are greater than those of Kosarak.

Varying pft. Figs.4(d)∼4(f) report the running time w.r.t. pft. We can find that GApriori is still the fastest algorithm in most of time. Different from the results w.r.t *minsup*, we observe that, by varying pft, the fluctuation of the running time is relative stable. Thus, pft does not have significant impact on the running time of the four proposed algorithms. This is reasonable because most of the probabilities of the correlated frequent itemsets are 1.

5.2 Pruning Power Test

To better verify the effectiveness of our proposed pruning technique, in this subsection, we report the pruning ratio of the upper-bound-based pruning in Figs.5(a)∼5(f).

Varying minsup. Figs.5(a)∼5(c) show the pruning ratio w.r.t. *minsup* in the Real dataset, Accident and Kosarak datasets, respectively. The pruning ratio in the Accident dataset is smaller than those in the Real and Kosarak datasets. The smaller pruning ratio indicates that the computation saved in the Accident dataset is smaller than that in the other two datasets. Moreover, the less pruning ratio makes sense because the Accident dataset is set with high mean and low variance for each item, and it is assigned non-zero correlation coefficients to 50% of the transactions as well. Therefore, the upper bound-based pruning in Lemma 3 becomes weaker since the expectations of the support of correlated infrequent itemsets in the Accident dataset are generally larger than those in the other two datasets.

Varying pft. Figs.5(d)∼5(f) report the pruning ratio w.r.t. pft. Different from the results w.r.t. *minsup*, we observe that the pruning ratio is stable by varying pft. The results confirm again that pft does not have significant impact on the efficiency of the algorithms.

5.3 Memory Cost Test

In this subsection, we report the results of memory cost of the four algorithms under different parameter settings.

Varying minsup. According to Fig.6(a), the memory costs of GApriori and GApriori-NoPrune are less than those of CApriori and CApriori-NoPrune, especially when *minsup* is low. This is reasonable since the numbers of infrequent candidates stored in both GApriori and GApriori-NoPrune are much smaller than those of the other algorithms. In addition, we can observe that the sharp change of the memory usage curve of CApriori-NoPrune is earlier than that of CApriori because there are few frequent itemsets when *minsup* is high and most of the infrequent itemsets are filtered out by the upper-bound-based pruning of CApriori.

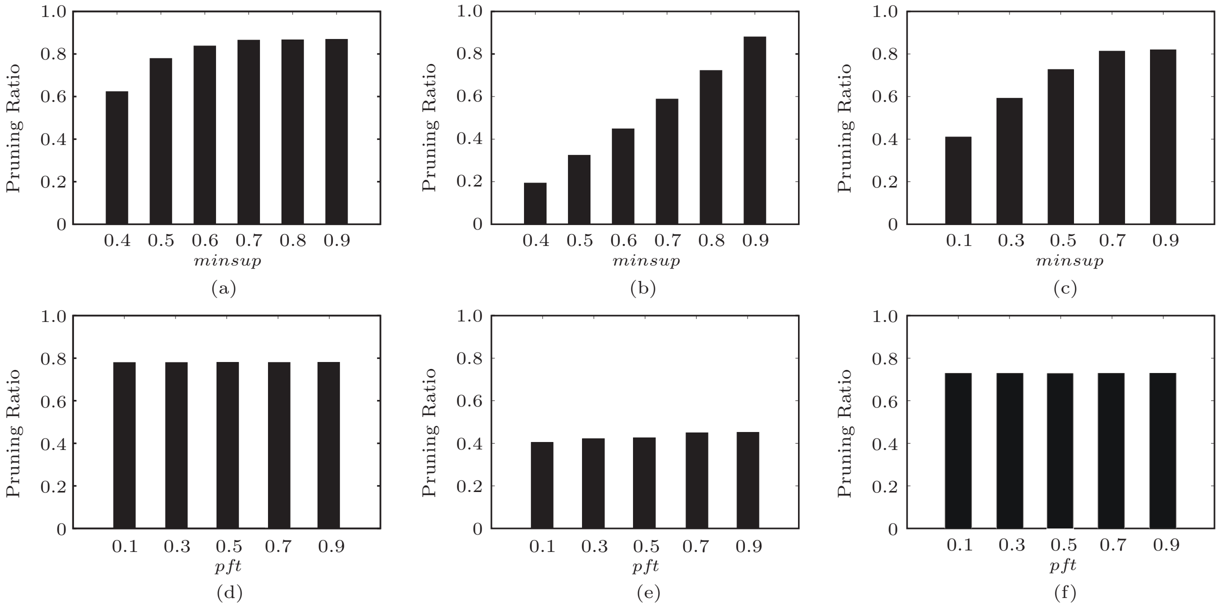


Fig.5. Test of pruning effect. (a) Real: *minsup* vs pruning ($pft = 0.9$). (b) Accident: *minsup* vs pruning ($pft = 0.9$). (c) Kosarak: *minsup* vs pruning ($pft = 0.9$). (d) Real: pft vs pruning ($minsup = 0.5$). (e) Accident: pft vs pruning ($minsup = 0.6$). (f) Kosarak: pft vs pruning ($minsup = 0.5$).

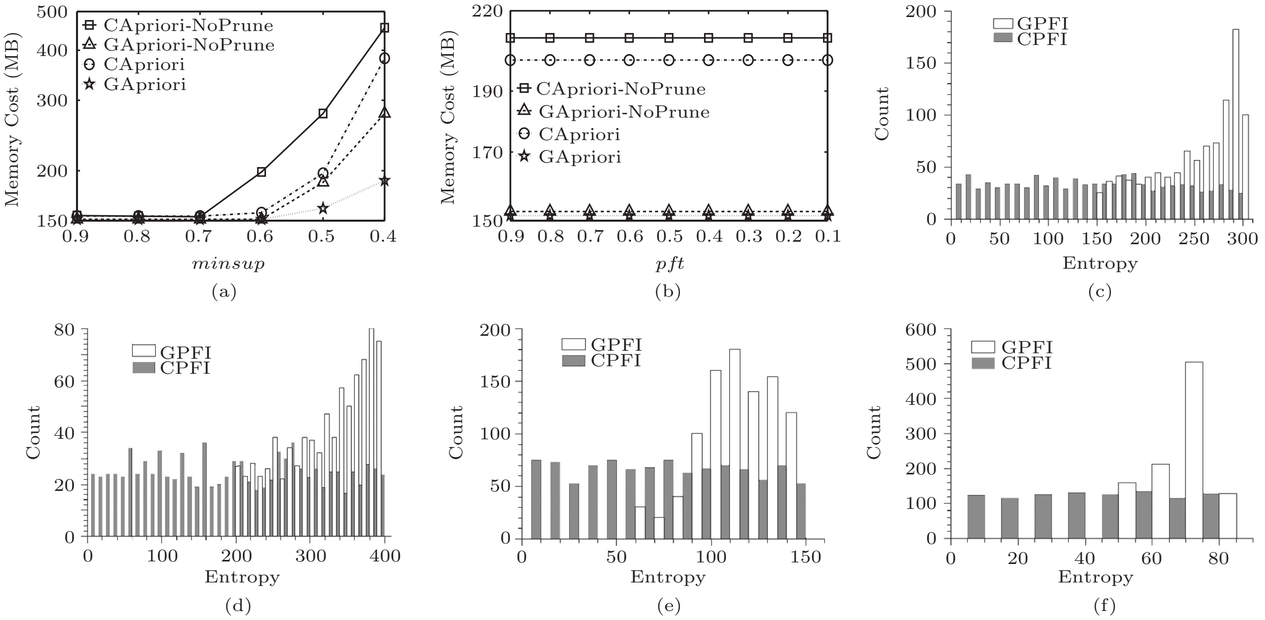


Fig.6. Test of memory cost and entropy-based quality of global frequent itemsets. (a) Accident: *minsup* vs memory ($pft = 0.9$). (b) Accident: pft vs memory ($minsup = 0.5$). (c) Accident: $minsup = 0.7$ and $pft = 0.9$. (d) Accident: $minsup = 0.4$ and $pft = 0.5$. (e) Kosarak: $minsup = 0.5$ and $pft = 0.9$. (f) Kosarak: $minsup = 0.1$ and $pft = 0.5$.

Varying pft. Fig.6(b) shows the memory cost w.r.t. pft. We observe that the memory usages of both GApriori and GApriori-NoPrune are smaller than those of the other two algorithms. Furthermore, we also observe that, by varying pft, the fluctuation of the memory cost is still stable. Thus, pft does not have significant impact on the memory cost of the four algorithms.

5.4 Quality of Global Frequent Itemsets

In this subsection, we analyze the quality of global frequent itemsets. We sample 1 000 itemsets from the results of CApriori and those of GApriori, respectively.

For each sample, we first compute the joint entropy of the correlated distribution of support in all the transactions in its corresponding dataset, and then we do frequency counting to get a counting distribution of entropy values for each algorithm. For example, the counting distribution of entropy values in the Kosarak dataset when $minsup = 0.1$ and $pft = 0.5$ is presented in Fig.6(f). We can observe that there are roughly 500 itemsets, whose entropy values are in the interval of 70 to 75, in the set sampled from the results of the GApriori algorithm. The black bars in Fig.6(f) report the counting distribution of entropy values based on the sampled results of the GApriori algorithm. Similarly, the counting distribution of the entropy values for results sampled from CApriori is shown in the gray bars in Fig.6(f).

Figs.6(c)∼6(f) show that the sampled results of global probabilistic frequent itemsets generally have a higher average entropy value, which indicates that: 1) the counting distribution of the entropy values for the sampled results of correlated probabilistic frequent itemsets is in overall more uniform than that of global probabilistic frequent itemsets. This is because the condition for correlated probabilistic frequent itemsets is not so strict as that for global probabilistic frequent itemsets, and thus all kinds of cases w.r.t. the probability distribution of correlated probabilistic frequent itemsets are possible. Therefore, there could be many different entropy values for the correlated probabilistic frequent itemsets. 2) The entropy values for the sampled results of global probabilistic frequent itemsets are usually higher because the global probabilistic frequent itemsets are less influenced by a local cluster of highly correlated transactions, which confirms the intuition of global frequent itemsets.

5.5 Scalability Test

In this subsection, we report the scalability of our proposed algorithms. In Fig.7(a), when increasing the number of transactions in the T25I15D320k dataset from 20 k to 320 k, we observe that the running time of the four algorithms almost increases linearly. However, the slopes of the four curves are different. The slopes of GApriori and GApriori-NoPrune are smaller than those of CApriori and CApriori-NoPrune. This result is reasonable because GApriori and GApriori-NoPrune aim to discover global frequent itemsets, and the other algorithms are to find correlated frequent itemsets. Additionally, Fig.7(b) reports the memory usages of the four algorithms, which demonstrate their linearity w.r.t. the number of transactions. The slopes of the memory usage curves of GApriori and GApriori-NoPrune are still smaller than those of other algorithms since GApriori and GApriori use the MCL algorithm to partition the whole database and only store the correlated groups with smaller sizes.

5.6 Mutual Information Test

Although we focus on addressing the linear correlation of correlated uncertain data in this paper, we also try to briefly evaluate other correlation measurements, such as the mutual information[3]. We select the aforementioned Accident dataset and 50% of the transactions in the Accident dataset are selected randomly, and each pair of them is assigned a non-zero normalized mutual information value, which is a real number generated in the interval of 0 to 1 following the uniform distribution. Fig.7(c) shows the running time of our proposed algorithm, CApriori, and the mutual-information-based algorithm, which utilizes the Apri-

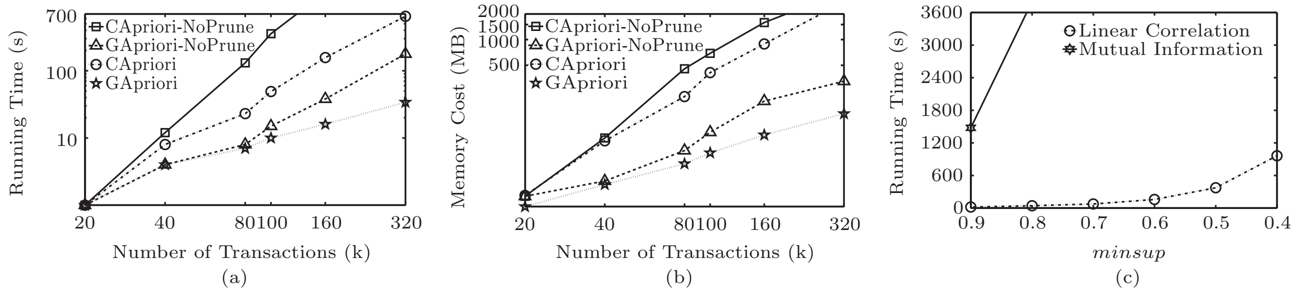


Fig.7. Test of scalability and mutual information-based correlated probabilistic frequent itemsets. (a) Scalability in Accident (running time). (b) Scalability in Accident (memory cost). (c) Linear correlation vs mutual information.

[3] $I(X;Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log(\frac{p(x,y)}{p(x)p(y)})$.

ori mining framework but calculates the correlated frequent probability based on mutual information. When *minsup* decreases, we observe that CApriori is significantly faster than the mutual-information-based algorithm. That is because that there is no dynamic programming scheme of calculating the correlated frequent probability based on the mutual information, and we have to enumerate all possible worlds to compute the correlated frequent probability based on the mutual information. Therefore, our model can efficiently address the linear correlation in correlated uncertain data.

6 Related Work

In this section, we review the related work in three categories, mining frequent itemsets in deterministic data and those in uncertain data, and techniques of mining and managing correlated uncertain data.

6.1 Deterministic Mining Frequent Itemsets

Since Agrawal and Srikant first proposed the concept of mining frequent itemsets (or called mining large itemset)[23], many efficient algorithms about mining frequent itemsets have been designed, such as FP-growth[26], Eclat[27], and so on. However, mining the complete set of frequent itemsets produces a lot of redundant itemsets because of the well-known downward closure property. To solve such problem, some alternative approaches were proposed instead of mining complete frequent itemsets, such as mining frequent closed itemsets[28], mining frequent maximal itemsets[29], mining non-derivable frequent itemsets[30], and so on. These alternative methods can be grouped into two categories: the lossless compression-based methods and the lossy compression-based methods. Among the aforementioned approaches, mining frequent closed itemsets and mining non-derivable itemsets are lossless compression-based methods. The others belong to the lossy compression-based methods. Moreover, even though there are a lot of studies on mining frequent itemsets, all these approaches do not concern the correlated property of uncertain data.

6.2 Uncertain Mining Frequent Itemsets

The second category of researches related to our work is mining frequent itemsets over uncertain data. Different from the deterministic case, the definition of a frequent itemset over uncertain data has two types of semantic explanations: expected support-based frequent itemset[8,11] and probabilistic frequent itemset[12], both of which consider the support of an itemset as a discrete random variable. However, the two definitions use different probabilistic methods on the random variable to define the frequent itemset over uncertain data. In the definition of the expected support-based frequent itemset, the expectation of the support of an itemset is defined as the measurement, called as the expected support of this itemset[8-9,11,15]. In the definition of probabilistic frequent itemset[12,16,19], the probability that an itemset appears at least the minimum support (*minsup*) times is defined as the measurement, called as the frequent probability of an itemset.

Although there are related researches of mining frequent itemsets over uncertain data, all of them are built over the independent assumption, which means that each transaction is independent from other transactions. In other words, none of the existing work about mining frequent itemsets over uncertain data considers correlation. Moreover, correlation is an important and universal property in real-world uncertain data, and thus the real application of mining frequent itemset over uncertain data cannot ignore the effect of correlation. Therefore, to the best of our knowledge, this work is the first one of mining probabilistic frequent itemsets based on the intrinsic correlation in uncertain data.

6.3 Mining and Managing Correlated Uncertain Data

In addition, managing and mining correlated uncertain (or probabilistic) data has attracted much attention from the database and the data mining communities. Sen *et al.* first proposed a general framework to reduce a query processing in probabilistic databases based on possible world semantics to the corresponding probabilistic inference problems in probabilistic graphical models (PGMs)[31-32]. Furthermore, the work in [33] modelled temporally correlated probabilistic streams by a graphical model. Besides the issues of query processing over correlated uncertain data, a few studies on clustering correlated uncertain data have been proposed in recent years. Olteanu and van Schaik[34] proposed a clustering algorithm to handle a symbolic representation of correlated probabilistic events. In addition, a junction-tree-based index, called INDSEP, was proposed to store the joint probabilities for the variables under/among the nodes for selection on uncertain data[35]. Recently, Gu *et al.* addressed the problem of clustering correlated probabilistic graphs[36].

Although there are related studies of querying and mining correlated uncertain data, most of them were proven as #P-hard problems and were solved by approximation algorithms. Different from aforementioned related studies, our work focuses on linear correlation and develops a polynomial-time solution to calculate the correlated frequent probability for each itemset and avoid the exponential enumeration computation for the corresponding possible worlds. Hence, to the best of our knowledge, this work is the first work of mining probabilistic frequent itemsets based on the linear correlated uncertain data.

7 Conclusions

In this paper, we studied the problem of mining frequent itemsets over correlated uncertain data, where there may be correlation between any pair of uncertain objects (tuples). For capturing the correlation in the given uncertain data, we proposed a novel probabilistic model, called Correlated Frequent Probability model (CFP model), which can represent the probability distribution of support of any itemset. Moreover, based on the experimental observations, we discovered that some probabilistic frequent itemsets are only frequent in several transactions with high positive correlation rather than in the whole database. In order to eliminate the redundant frequent itemsets and the noisy influence in uncertain data, a new type of itemset, called global frequent itemsets, was defined to identify itemsets. In addition, to enhance the efficiency of the mining process, we designed an Apriori-style framework which seamlessly integrates a dynamic-programming-based efficient algorithm with two pruning and bounding techniques. In particular, a scheduler-based efficient algorithm was also developed for mining global frequent itemsets. Extensive experiments on both real and synthetic datasets verified the effectiveness and efficiency of the proposed model and algorithms.

References

[1] Böhm C, Gruber M, Kunath P, Pryakhin A, Schubert M. ProVer: Probabilistic video retrieval using the Gauss-tree. In *Proc. the 23rd ICDE*, April 2007, pp.1521-1522.

[2] Chen L, Ng R T. On the marriage of Lp-norms and edit distance. In *Proc. the 30th VLDB*, August 31-September 3, 2004, pp.792-803.

[3] Chen L, Özsu M T, Oria V. Robust and fast similarity search for moving object trajectories. In *Proc. ACM SIGMOD*, June 2005, pp.491-502.

[4] Cheng R, Kalashnikov D V, Prabhakar S. Querying imprecise data in moving object environments. *IEEE Trans. Knowl. Data Eng.*, 2004, 16(9): 1112-1127.

[5] Deshpande A, Guestrin C, Madden S, Hellerstein J M, Hong W. Model-driven data acquisition in sensor networks. In *Proc. the 30th VLDB*, August 31-September 3, 2004, pp.588-599.

[6] Kodialam M S, Nandagopal T. Fast and reliable estimation schemes in RFID systems. In *Proc. the 12th MOBICOM*, September 2006, pp.322-333.

[7] Liu Y, Liu K, Li M. Passive diagnosis for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 2010, 18(4): 1132-1144.

[8] Chui C K, Kao B, Hung E. Mining frequent itemsets from uncertain data. In *Proc. the 11th PAKDD*, May 2007, pp.47-58.

[9] Chui C K, Kao B. A decremental approach for mining frequent itemsets from uncertain data. In *Proc. the 12th PAKDD*, May 2008, pp.64-75.

[10] Calders T, Garboni C, Goethals B. Efficient pattern mining of uncertain data with sampling. In *Proc. the 14th PAKDD*, June 2010, pp.480-487.

[11] Aggarwal C C, Li Y, Wang J, Wang J. Frequent pattern mining with uncertain data. In *Proc. the 15th SIGKDD*, June 28–July 1, 2009, pp.29-38.

[12] Bernecker T, Kriegel H P, Renz M, Verhein F, Züfle A. Probabilistic frequent itemset mining in uncertain databases. In *Proc. the 15th SIGKDD*, June 28–July 1, 2009, pp.119-128.

[13] Calders T, Garboni C, Goethals B. Approximation of frequentness probability of itemsets in uncertain data. In *Proc. the 10th ICDM*, December 2010, pp.749-754.

[14] Gao C, Wang J. Direct mining of discriminative patterns for classifying uncertain data. In *Proc. the 16th SIGKDD*, July 2010, pp.861-870.

[15] Leung C K S, Mateo M A F, Brajczuk D A. A tree-based approach for frequent pattern mining from uncertain data. In *Proc. the 12th PAKDD*, May 2008, pp.653-661.

[16] Sun L, Cheng R, Cheung D W, Cheng J. Mining uncertain data with probabilistic guarantees. In *Proc. the 16th SIGKDD*, July 2010, pp.273-282.

[17] Tong Y, Chen L, Ding B. Discovering threshold-based frequent closed itemsets over probabilistic data. In *Proc. the 28th ICDE*, April 2012, pp.270-281.

[18] Tong Y, Chen L, Cheng Y, Yu P S. Mining frequent itemsets over uncertain databases. *PVLDB*, 2014, 5(11): 1650-1661.

[19] Wang L, Cheng R, Lee S D, Cheung D W. Accelerating probabilistic frequent itemset mining: A model-based approach. In *Proc. the 19th CIKM*, October 2010, pp.429-438.

[20] Zhang Q, Li F, Yi K. Finding frequent items in probabilistic data. In *Proc. ACM SIGMOD*, June 2008, pp.819-832.

[21] Schoute F. Dynamic frame length ALOHA. *IEEE Trans. Communications*, 1983, 31(4): 565-568.

[22] Lancaster H O. The Chi-Squared Distribution. New York, USA: Wiley, 1969.

[23] Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases. In *Proc. the 20th VLDB*, September 1994, pp.487-499.

[24] van Dongen S M. Graph clustering by flow simulation [Ph.D. Thesis]. University of Utrecht, 2000.

[25] Mo L, He Y, Liu Y, Zhao J, Tang S, Li X Y, Dai G. Canopy closure estimates with GreenOrbs: Sustainable sensing in the forest. In *Proc. the 7th SenSys*, November 2009, pp.99-112.

[26] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation. In *Proc. ACM SIGMOD*, May 2000, pp.1-12.

[27] Zaki M J. Scalable algorithms for association mining. *IEEE Trans. Knowl. Data Eng.*, 2000, 12(3): 372-390.

[28] Pasquier N, Bastide Y, Taouil R, Lakhal L. Discovering frequent closed itemsets for association rules. In *Proc. the 7th ICDT*, January 1999, pp.398-416.

[29] Bayardo R J. Efficiently mining long patterns from databases. In *Proc. ACM SIGMOD*, June 1998, pp.85-93.

[30] Calders T, Goethals B. Mining all non-derivable frequent itemsets. In *Proc. the 6th PKDD*, August 2002, pp.74-85.

[31] Sen P, Deshpande A. Representing and querying correlated tuples in probabilistic databases. In *Proc. the 23rd ICDE*, April 2007, pp.596-605.

[32] Sen P, Deshpande A, Getoor L. Exploiting shared correlations in probabilistic databases. *PVLDB*, 2008, 1(1): 809-820.

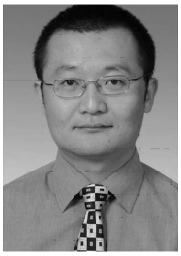
[33] Kanagal B, Deshpande A. Efficient query evaluation over temporally correlated probabilistic streams. In *Proc. the 25th ICDE*, March 29–April 2, 2009, pp.1315-1318.

[34] Olteanu D, van Schaik S J. Dagger: Clustering correlated uncertain data (to predict asset failure In energy networks). In *Proc. the 18th SIGKDD*, August 2012, pp.1504-1507.

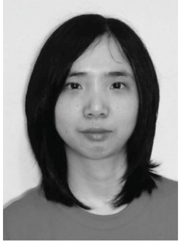
[35] Kanagal B, Deshpande A. Indexing correlated probabilistic databases. In *Proc. ACM SIGMOD*, June 2009, pp.455-468.

[36] Gu Y, Gao C, Cong G, Yu G. Effective and efficient clustering methods for correlated probabilistic graphs. *IEEE Trans. Knowl. Data Eng.*, 2014, 26(5): 1117-1130.

Yong-Xin Tong received his Ph.D. degree in computer science and engineering from the Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2014. He is currently an associate professor in the School of Computer Science and Engineering, Beihang University, Beijing. Before that, he served as a research assistant professor and a postdoctoral fellow at HKUST. He is a member of CCF, ACM, and IEEE. His research interests include crowdsourcing, uncertain data mining and management, and social network analysis.



Lei Chen received his B.S. degree in computer science and engineering from Tianjin University, Tianjin, in 1994, M.A. degree from Asian Institute of Technology, Bangkok, Thailand, in 1997, and Ph.D. degree in computer science from the University of Waterloo, Canada, in 2005. He is currently an associate professor in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. So far, he published over 200 conference and journal papers. He got the Best Paper Awards in DASFAA 2009 and 2010. He is PC Track chairs for SIGMOD 2014, VLDB 2014, ICDE 2012, CIKM 2012, SIGMM 2011. He has served as PC members for SIGMOD, VLDB, ICDE, SIGMM, and WWW. Currently, Prof. Chen is an associate editor-in-chief for IEEE Transactions on Knowledge and Data Engineering and serves on the editorial board of Distributed and Parallel Databases. He is a member of the VLDB endowment committee and the chairman of ACM SIGMOD China Chapter. His research interests include crowdsourcing over social media, social media analysis, probabilistic and uncertain databases, and privacy-preserved data publishing.



Jieying She is currently a Ph.D. student at the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology, Hong Kong. Her major research interest is managing event-based social networks. She is a student member of IEEE.