Technical Report: Mining Probabilistic Frequent Closed Itemsets in Uncertain Databases*

Peiyi Tang
Department of Computer Science
University of Arkansas at Little Rock
2801 S. University Ave., Little Rock, AR 72204
pxtang@ualr.edu

Erich A. Peterson
Department of Computer Science
University of Arkansas at Little Rock
2801 S. University Ave., Little Rock, AR 72204
contact@erichpeterson.com

February 21, 2011

Abstract

This paper defines probabilistic support and probabilistic frequent closed itemsets in uncertain databases for the first time. It also proposes a probabilistic frequent closed itemset mining (PFCIM) algorithm to mine probabilistic frequent closed itemsets from uncertain databases.

*This work was supported in part by the National Science Foundation under Grant CRI CNS-0855248, Grant EPS-0701890, Grant EPS-0918970, Grant MRI CNS-0619069, and OISE-0729792.

1 Introduction

Broadly speaking, renewed interest in the field of mining in uncertain databases has been motivated by the advent of applications which lend themselves to that area. More specifically, some modern applications are known to produce incomplete or noisy data; one salient example being a sensor network [9, 12]. Further, privacy-preserving data mining applications [6, 10] in particular have a need for frequent itemset mining algorithms that operate within an uncertain data context.

Starting with the work of Agrawal et al. [2], mining frequent itemsets has been extensively (and sometimes seemingly exhaustively) studied; that research, however, has focused on so-called certain databases (i.e., where each transaction and the items it contains is known for sure). This contrasts sharply with research into mining for frequent itemsets in databases that contain transactions or items that have existential probabilities, which has been studied in [11, 4, 5, 3, 1, 8]. Put another way: "All previous studies ... assume a data model under which transactions capture doubtless facts about the items that are contained in each transaction." [5] Thus, in these so-called *uncertain databases*, one can not be certain about whether an itemset is frequent or not, and as to the makeup of a particular database. All we can can do is estimate if an itemset is frequent or not within a certain confidence.

There have been works on mining frequent itemsets from uncertain databases. Chui at et al. [5, 4] uses the expected support of an itemset from an uncertain database to define whether it is frequent or not. Itemsets are considered frequent if its expected support exceeds the minimum support threshold *minsup*. As indicated by [3], a frequent itemset based the expected support cannot express how close the estimate is that it is frequent. Bernecker et al. [3] defined the support distribution of an itemset and used it to define the frequentness probability as the sum of the probabilities of the support equal to or above the minimum support *minsup*. Thus, the frequent itemsets with conference threshold τ are the itemsets whose frequentness probabilities exceeds τ.

In traditional (certain database) data mining, the frequent *closed* itemsets are widely seen as being a more compact and lossless representation of frequent itemsets. A frequent itemset is closed if it is the Galois closure of itself or if it has no super itemset with the same support.

Hitherto, no researchers have proposed a method for mining frequent closed itemsets in uncertain databases. In fact, no researchers have defined

the probabilistic support of an itemset accurately and closed itemsets in uncertain databases. In this paper, we use maximum frequentness to define the probabilistic support of an itemset and use the probabilistic support to define closed itemsets in uncertain databases for the first time. We also proposed an algorithm (PFCIM) to mine probabilistic frequent closed itemsets in uncertain databases, by extending the probabilistic frequent itemset mining (PFIM) algorithm from [3].

The rest of this paper is laid out as follows: Section 2 will disseminate some necessary concepts and notations for the full understanding of later material. In Section 3, we define the probabilistic support of an itemset based on maximum frequentness and then probabilistic frequent closed itemsets in uncertain databases. Section 4 describes the probabilistic frequent closed itemset mining (`PFCIM`) algorithm. Section 5 provides an experimental evaluation of the algorithm; finally, Section 6 concludes the paper.

2 Preliminaries

2.1 Uncertain Data Model

The uncertain data model used in this paper assumes the presence of a set of items $I = \{x_1, x_2, \ldots, x_m\}$ and a set of transactions $T = \{t_1, t_2, \ldots, t_n\}$. Each item $x \in I$ has an accompanying *existential probability* of being in transactions t_j, denoted as $P(x \in t_j) \in [0, 1]$. The item x with $0 < P(x \in t_j) < 1$ is called an uncertain item in t_j. The zero existential probability of item x in t_j, $P(x \in t_j) = 0$, simply means that x does not exist in t_j. Item x with $P(x \in t_j) = 1$ is contained in t_j with full certainty.

An uncertain database T over itemset I can be represented by a $n \times m$ matrix M, where $M_{j,i}$ is the existential probability of i-th item x_i in j-th transaction t_j, $M_{j,i} = P(x_i \in t_j)$. If all the existential probabilities in M are either 1 or 0, the database degenerates to a traditional certain database. Thus, a certain database can be regarded as a special case of uncertain database.

An example uncertain database is shown in Figure 1, where $I = \{1, 2, 3\}$ and there are three transactions $T = \{t_1, t_2, t_3\}$. In transaction t_j, item x and its existential probability $P(x \in t_j)$ in it is represented as the tuple (x, $P(x \in t_i)$). The items with zero existential probability are not shown.

In Figure 1, each transaction could uniquely identify a loyal customer,

TID	Itemset
t_1	(1, 1.0), (3, 0.99)
t_2	(2, 0.4), (3, 0.88)
t_3	(1, 0.9), (2, 0.2), (3, 0.95)

Figure 1: Uncertain Database

and each uncertain item (i.e., 1, 2, 3) could represent a particular store item and the probability of a customer purchasing that item. Alternatively, each TID could represent a patient and each item a disease and the probability of that patient being diagnosed with that disease.

An uncertain database defines a number of *possible world*s. "A *possible world* is a hypothetical state of the world that may be represented by an ordinary database with complete and certain information." [13] Since for each probability $P(x \in t_j)$, there exists a possible world that includes item x in transaction t_j and another that does not, there are total of $2^{|T|\cdot|I|} = 2^{n \cdot m}$ possible worlds. Assuming that the transactions in the uncertain database are independent (customers' purchase pattern are independent) and the existential probabilities of the items in each transaction are also independent (the probabilities of the items purchased by a customer are independent), the probability of a possible word w, denoted as $P(w)$, which is the joint probability of all its certain transactions, is

$$P(w) = \prod_{t \in T(w)} (\prod_{x \in t} P(x \in t') \cdot \prod_{x \notin t} (1 - P(x \in t'))) \tag{1}$$

where $T(w)$ is the set of certain transactions of world w, t a certain transaction in $T(w)$, t' the corresponding *uncertain* transaction in uncertain database T, and $P(x \in t')$ the existential probability of item x in the uncertain transaction t'. It can be proved that $\sum_{w \in W} P(w) = 1$, where W is the set of all $2^{|T|\cdot|I|}$ possible worlds. If the existential probabilities $P(x \in t_j)$ are either 1 or 1 for all $t_j \in T$ and item $x \in I$ (i.e. T is a certain database), the probability of that possible world equal to T is 1 and all the other possible worlds have zero probability.

2.2 Probabilistic Frequent Itemsets

The possible worlds and their probabilities are the foundation of reasoning about the support of itemsets in uncertain databases. Since each possible

world w and its set of transactions $T(w)$ are certain, the *support* of each itemset X in $T(w)$ is well-defined and is the number of transactions in $T(w)$ that contains X, denoted as $Sup_{T(w)}(X)$. Thus, the probability of the support of X being i, denoted as $P_i(X)$ in the original uncertain database is

$$P_i(X) = \sum_{w \in W, Sup_{T(w)}(X)=i} P(w) \tag{2}$$

where W is the set of possible worlds. Therefore, an uncertain database T defines a discrete probability distribution of the support of each itemset X, $P_i(X)$ $(i = 0, 1, \ldots, |T|)$, according to (2).

Bernecker et. al. [3] proved that support probability distribution $P_i(X)$ can be calculated without materializing all the possible worlds by

$$P_i(X) = \sum_{S \subseteq T, |S|=i} (\prod_{t \in S} P(X \subseteq t) \cdot \prod_{t \in T-S} (1 - P(X \subseteq t))) \tag{3}$$

where T is the original uncertain database and $P(X \subseteq t)$ is

$$P(X \subseteq t) = \prod_{x \in X} P(x \in t)$$

All the work on frequent itemset mining in uncertain databases prior to [3] used the expected support of the support distribution $P_i(X)$, $E(X) = \sum_{i=0}^{|T|} i \cdot P_i(X)$, to define frequent itemsets as being those whose expected support exceeds the minimum support *minsup*.

Instead of using the expected support, Bernecker et. al. [3] proposed to use *frequentness probability* to define *probabilistic frequent itemsets* with a certain confidence. The probability that the support of itemset X is at least i is $P_{\geq i}(X) = \sum_{k=i}^{|T|} P_k(X)$. Thus, $P_{\geq minsup}(X)$ is the probability that itemset X is frequent and is called the *frequentness probability* of X. In [3], the *probabilistic frequent itemsets with confidence* τ are the itemsets whose frequentness probability $P_{\geq minsup}(X)$ exceeds τ.

3 Probabilistic Frequent Closed Itemset Mining

Frequent itemset mining has two drawbacks: (1) there are often too many frequent itemsets to report and digest and (2) frequent itemsets mined do

not have information about their frequentness or support. Mining maximal frequent itemsets can solve the first problem, but not the second one. Only mining closed frequent itemsets solves both problems.

In the certain database mining, an itemset X is closed if and only if it is the Galois closure of itself, i.e. $X = c(X)$. Here c is the Galois closure operator defined as $c = f \circ g$, where $g : I \to T$ and $f : T \to I$ are the two functions defined as follows. Given itemset X, $g(X)$ is the set of transactions that contain X, i.e. $g(X) = \{t \in T \mid X \subseteq t\}$. Given a set of transactions $Y \subseteq T$, $f(Y)$ is the maximal itemset that are contained in all transactions in Y, i.e. $f(Y) = \{x \in I \mid \forall t \in Y, x \in t\}$. In other words, an itemset X is closed if and only if $f(g(X)) = X$.

The support of an itemset X, $Sup_T(X)$, in an certain database T is the number of transactions in T that contain X, i.e. $Sup_T(X) = |g(X)|$. We can prove that an itemset X is closed if and only if it does not have any proper super itemset with the same support, i.e. there is no itemset Y such that $X \subset Y$ and $Sup_T(X) = Sup_T(Y)$. Often, this theorem is used as a second alternative definition for closed itemsets.

3.1 Probabilistic Support of Itemsets

In uncertain database mining, we cannot use Galois closure to define a closed itemset, because the database T is uncertain and functions g and f are not defined. However, we may be able to use the second definition to define closed itemset. The challenge is that for an uncertain database T, the support of an itemset X does not have a specific value. It is rather a discrete random number with distribution $P_i(X)$ $(i = 0, \ldots, |T|)$ determined by (3). Bernecker et. al. [3] used frequentness probability $P_{\geq minsup}(X) = \sum_{k=minsup}^{|T|} P_k(X)$, the probability that the support of X is at least *minsup*, to define *probabilistic frequent itemsets* with certain conference τ to be the itemsets X such that $P_{\geq minsup}(X) \geq \tau$. But, the *probabilistic support* of an itemset with a certain confidence has never been defined, although the term of "probabilistic support" appeared in [3].

Note that $P_{\geq i}(X) = \sum_{k=i}^{|T|} P_k(X)$ is a non-increasing monotonous function of i, i.e. $P_{\geq j}(X) \leq P_{\geq i}(X)$ for $j > i$. In this paper, we define the probabilistic support of itemset X with confidence τ, denoted as $Sup_T(X, \tau)$, to be the largest i such that $P_{\geq i}(X) \geq \tau$. Formally,

Definition 1 *Given an itemset X, uncertain database T, and confidence threshold τ, the probabilistic support of X with confidence τ, denoted as $Sup_T(X, \tau)$, is defined as follows:*

$$Sup_T(X, \tau) = argmax_{i \in [0, |T|]}(P_{\geq i}(X) \geq \tau)$$

The probabilistic support $Sup_T(X, \tau)$ above is the largest threshold, above which we can say about the support of X in database T with confidence τ. In other words, it is the maximum frequentness of itemset X with confidence τ, and indicates how frequent—probabilistically—an itemset X is.

3.2 Probabilistic Frequent Closed Itemset

Bernecker et. al. [3] also proved that frequentness probability $P_{\geq minsup}(X) = \sum_{k=minsup}^{|T|} P_k(X)$ is anti-monotonic. That is, for any $Y \subseteq X$, and any i, $P_{\geq i}(X) \leq P_{\geq i}(Y)$ (Lemma 17 of [3]). Using the anti-monotonic property of $P_{\geq i}(X)$, We can prove the following anti-monotonic property of the probabilistic support $Sup_T(X, \tau)$ as follows:

Lemma 1 *For all itemsets $Y \subseteq X$ in an uncertain database T and any confidence τ, $Sup_T(X, \tau) \leq Sup_T(Y, \tau)$. In other words, the probabilistic support with the same confidence decreases as the itemset increases.*

Proof 1 *Suppose the contrary that $Sup_T(X, \tau) > Sup_T(Y, \tau)$. Let $Sup_T(X, \tau)$ and $Sup_T(Y, \tau)$ be k and j, respectively, and we have $k > j$. Since $Sup_T(X, \tau) = k$, we have $P_{\geq k}(X) \geq \tau$ according to Definition 1. Since $Y \subseteq X$, we have $P_{\geq k}(X) \leq P_{\geq k}(Y)$ according to the anti-monotonic property of frequentness probability (Lemma 17 of [3]). Thus, we have $P_{\geq k}(Y) \geq \tau$. Since $k > j$ and $P_{\geq k}(Y) \geq \tau$, j is not the largest i such that $P_{\geq i}(Y) \geq \tau$. Therefore, $Sup_T(Y, \tau)$ is not j according to Definition 1. We, thus, reached the contradiction and $Sup_T(X, \tau) \leq Sup_T(Y, \tau)$ must be true.*

Lemma 1 shows that the probabilistic support of itemsets of uncertain databases, defined in Definition 1, has the similar anti-monotonous property for the support of itemsets in certain databases. This property allows us to define probabilistic closed itemsets in uncertain databases, by following the second alternative definition of a closed itemset in certain databases.

Definition 2 *Given an uncertain database T and a confidence threshold τ, an itemset X is probabilistically closed with confidence τ if and only if there is no proper super itemset $Y \supset X$ that has the same probabilistic support with the same confidence τ as X, i.e. with $Sup_T(Y, \tau) = Sup_T(X, \tau)$.*

Just as not all itemsets are frequent probabilistically, not all probabilistic closed itemsets are frequent. We can define in our term that an itemset X is probabilistic frequent with respect to minimum support *minsup* and confidence τ if and only if its probabilistic support, $Sup_T(X, \tau)$, is at least *minsup*.

Definition 3 *Given an uncertain database T, a minimum support minsup between 0 and $|T|$, and a confidence threshold τ between 0 and 1, an itemset X is probabilistically frequent if and only its probabilistic support with confidence τ exceeds minsup, i.e. $Sup_T(X, \tau) \geq minsup$.*

In [3], frequent itemsets X are defined as those satisfying $P_{\geq minsup}(X) \geq \tau$. The following lemma shows that $P_{\geq minsup}(X) \geq \tau$ is actually equivalent to $Sup_T(X, \tau) \geq minsup$.

Lemma 2 *Given an uncertain database T, a minimum support minsup between 0 and $|T|$, a confidence threshold τ between 0 and 1, and an itemset X, $Sup_T(X, \tau) \geq minsup$ if and only if $P_{\geq minsup}(X) \geq \tau$.*

Proof 2 *($\Rightarrow$) Let $Sup_T(X, \tau) \geq minsup$ be j. If $Sup_T(X, \tau) \geq minsup$, then $j \geq minsup$ and also $P_{\geq j}(X) \geq \tau$ (Definition 1). We have $P_{\geq j}(X) = \sum_{k=j}^{|T|} P_k(X) \geq \tau$. $P_{\geq minsup}(X)$ can be divided as $P_{\geq minsup}(X) = \sum_{minsup}^{j} P_k(X) + \sum_{j}^{|T|} P_k(X)$ because $j \geq minsup$. Since $\sum_{minsup}^{j} P_k(X) \geq 0$, we have $P_{\geq minsup}(X) \geq \sum_{j}^{|T|} P_k(X) \geq \tau$.*
($\Leftarrow$) Let $P_{\geq minsup}(X) \geq \tau$ be true and $Sup_T(X, \tau)$ be j. Assume the contrary that $j < minsup$. Since $P_{\geq minsup}(X) \geq \tau$ and $j < minsup$, $Sup_T(X, \tau)$ cannot be j according to Definition 1. Thus, $j \geq minsup$.

Therefore, our definition of frequent itemsets by Definition 3 is equivalent to the one in [3]. But, our definition is more in line with the traditional frequent itemset definition for certain databases.

3.3 PFCIM Problem Definition

Now, the probabilistic frequent closed itemset can be defined as follows:

Definition 4 *Given an uncertain database T, a minimum support minsup between 0 and $|T|$, a confidence threshold τ between 0 and 1, X is a probabilistic frequent closed itemset with confidence τ if and only (1) there is no proper super itemset $Y \supset X$ such that $Sup_T(Y, \tau) = Sup_T(X, \tau)$ and (2) $Sup_T(X, \tau) \geq minsup$.*

The problem of probabilistic frequent closed itemsets mining (PFCIM) in uncertain databases can be defined as follows:

Definition 5 *Given an uncertain database T, the minimum support minsup between 0 and $|T|$, and confidence threshold τ between 0 and 1, the problem of probabilistic frequent closed itemset mining (PFCIM) is to find all the itemset X such that (1) there is no proper super itemset $Y \supset X$ such that $Sup_T(Y, \tau) = Sup_T(X, \tau)$ and (2) $Sup_T(X, \tau) \geq minsup$.*

4 Mining Algorithm

As shown previously, $P_{\geq i}(X)$ is defined as $\sum_{k=i}^{|T|} P_k(X)$ and $P_i(X)$ can be calculated using Equation (3).

Bernecker et al. [3] shows that $P_{\geq i}(X)$ can also be calculated as:

$$P_{\geq i}(X) = \sum_{S \subseteq T, |S| \geq i} (\prod_{t \in S} P(X \subseteq t) \cdot \prod_{t \in T - S} (1 - P(X \subseteq t))) \tag{4}$$

The complexity of computing $P_{\geq i}(X)$ is exponential with respect to the size of database $|T|$.

In [3], Bernecker et al. use a dynamic programming scheme to calculate $P_{\geq i}(X)$ in linear time $O(|T|)$ by calculating $P_{\geq i,j}(X)$, which is $P_{\geq i}(X)$ from the first j transactions of T only. In other words, $P_{\geq i}(X) = P_{\geq i,|T|}(X)$. The recursive equation for $P_{\geq i,j}(X)$ is

$$\begin{aligned} P_{\geq i,j}(X) &= P_{\geq i-1,j-1}(X) \cdot P(X \subseteq t_j) \\ &+ P_{\geq i,j-1}(X) \cdot (1 - P(X \subseteq t_j)) \end{aligned} \tag{5}$$

$$\text{where } P_{\geq 0,j} = 1 \ \forall . 0 \leq j \leq |T|, P_{\geq i,j} = 0 \ \forall . i > j$$

Figure 2(a) taken from [3] illustrates the aforementioned dynamic programming scheme used in calculating
$P_{\geq minsup,|T|}(X)$ for mining probabilistic frequent itemsets.

For mining probabilistic frequent closed itemsets, we need to find the probabilistic support $Sup_T(X,\tau)$. This means that after we find
$P_{\geq minsup,|T|}(X) \geq \tau$, we need to continue calculating $P_{\geq i,|T|}(X)$ for $i > minsup$ as long as
$P_{\geq i-1,|T|}(X) \geq \tau$, until it is less than τ. In Figure 2(b), we see an example where the computation is run until
$P_{\geq minsup+2,|T|}(X) < \tau$ is reached for the first time, which makes $Sup_T(X,\tau) = minsup + 1$.

Figure 3 shows our function `ProbSup()` for calculating the probabilistic support of an itemset using the dynamic programming scheme shown in Figure 2(b). An itemset X has two fields: 1) an integer $X.MS$ to hold the probabilistic support of the itemset, $Sup_{|T|}(X,\tau)$, and 2) an array $X.P[1\ldots|T|]$ of floats, where $X.P[j]$ is used to store $P(X \subseteq t_j)$.

Recall that $P(X \subseteq t) = \prod_{x \in X} P(x \in t)$. Therefore, we have

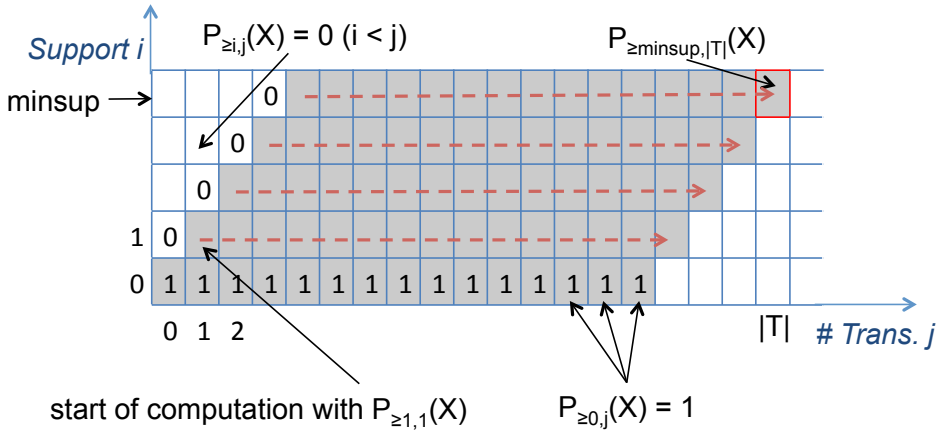
$$P((X \cup \{b\}) \subseteq t_j) = P(X \subseteq t_j) \cdot P(b \in t_j)$$

We explore this fact and store $P(X \subseteq t_j)$ in $X.P[j]$ so that when we need to calculate $P((X \cup \{b\})$, we simply retrieve $X.P[j]$ and multiply it with $P(b \in t_j)$ instead of calculating the product $\prod_{x \in X} P(x \in t)$ directly; this speeds up the computation. Thus, the dynamic programming Equation 5 can be re-formulated as:

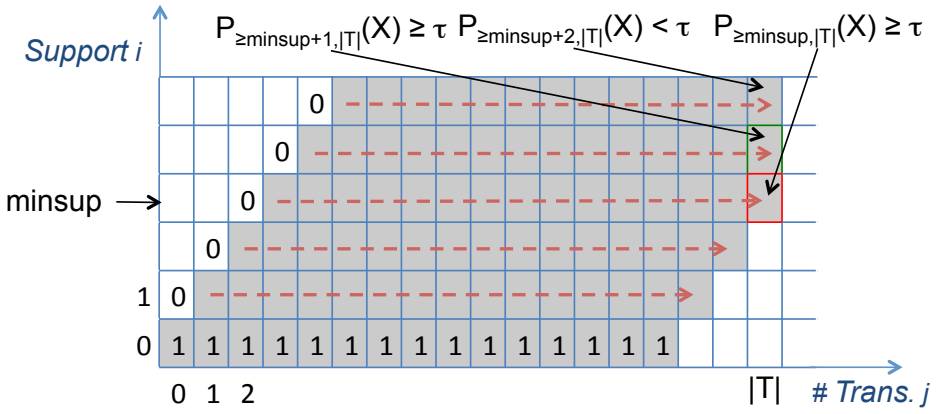
$$\begin{aligned} P_{\geq i,j}(X \cup \{b\}) &= P_{\geq i-1,j-1}(X \cup \{b\}) \cdot \\ &\quad P(X \subseteq t_j) \cdot P(b \in t_j) \\ &+ P_{\geq i,j-1}(X \cup \{b\}) \cdot \\ &\quad (1 - P(X \subseteq t_j) \cdot P(b \in t_j)) \end{aligned} \tag{6}$$

In Figure 3, the uncertain database is stored in a two-dimensional T, where the existential probability for item $x \in I$ in transaction $t_j \in T$, $P(x \in t_j)$, is stored in $T[j][x]$ (items are represented as natural numbers and thus used as an index directly).

The probabilistic support of itemset X is found by calling `ProbSup(X, tau, minsup, L)`, where `L` is a set that contains the itemset of $(|X|-1)$-prefix of X ($|X| > 1$). The `if` statement at line 2 of Figure 3, retrieves this



(a) Calculation of $P_{\geq minsup,|T|}(X)$ [3]



(b) Calculation of $Sup_T(X, \tau)$

Figure 2: Dynamic Programming Schemes

prefix using $prefix$ function (not shown) and assigns it to Y. The purpose is to have access to $Y.P[\ldots]$. The next two nested loops i and j are used to compute $P_{\geq i,j}(X)$ saved in $matrix[i][j]$. Take note that values of j that are greater than $|T| - minsup + i$ and $|T|$ need not be visited. Also note that for simplicity, the entire $matrix$ (possibly large) is shown, but when implementing, only rows i and $i-1$ are necessary—greatly reducing the memory burden. Lines 11–17 simply initialize the first row of the matrix in Figure 2(b) to all 1's or initialize a cell $j-1$ to 0 if $i = 0$. Next, x is the last item of itemset X—obtained by calling the $postfix$ function (not shown). x can be seen as being equivalent to b in Equation 6. At line 19 the whole of Equation 6 is performed to calculate $P_{\geq i,j}(X)$. Within this calculation, one can see the reuse of previous calculations of existential probability of an itemset occurring in transaction t_j, by way of referencing $Y.P[j]$. Next, at line 20, if $i = 1$ or $j = |T| - minsup + i$, then the probability of itemset X occurring in transaction t_j ($P(X \subseteq t_j)$) is saved for itemsets of length $k+1$. Finally, at line 24, if $matrix[i][j-1] < \tau$—meaning that the itemset will not be frequent and there is no need continuing computation—and $i \neq 0$, then $i-1$[1] is returned. Otherwise, the computation continues until this condition is meet, or execution flows out of all loops and line 28 is reached—at which point $i-1$ will be returned.

Our Probabilistic Frequent Closed Itemset Mining (PFCIM) algorithm uses an apriori-style breadth-first technique for its mining. That is, all probabilistic frequent closed itemsets of length 1 are discovered first, followed by those of length 2, and so on.

The closure checking (to determine if a frequent item is closed or not) is based on the property that a k-itemset (itemset of length k) is closed if none of its $k+1$-super-itemsets has the same support.

Property 1 *Given a probabilistic frequent itemset X of length k, if all $Y \supset X$ of length $k+1$ have a probability support not equal to X (i.e., $Sup_T(X,\tau) \neq Sup_T(Y,\tau)$), then X is a probabilistic frequent closed itemset.*

Proof 3 *Let Z be a superset of X of length greater than $k+1$, i.e. $Z \supset X$ and $|Z| > k+1$. Then, there must be an itemset Y of length $k+1$ such that $Z \supset Y \supset X$. According to the anti-monotonous property (Lemma 1), we have $Sup_T(Z,\tau) \leq Sup_T(Y,\tau) \leq Sup_T(X,\tau)$. Since we know $Sup_T(Y,\tau) \neq Sup_T(X,\tau)$, we have $Sup_T(Z,\tau) \leq Sup_T(Y,\tau) < Sup_T(X,\tau)$. Thus,*

[1] $i-1$ is returned because i is incremented by the `for` loop before exiting it.

```
function ProbSup(itemset X, float τ,
                 int minsup, optional L ← NULL)
begin
1. itemset Y;
2. if L ≠ NULL then
     Y ← prefix(X, L);
   else
     Y.P[1 ... |T|] ← 1;
   endif
   int i;
8. for (i ← 0; i ≤ |T|; i++) do
     int j;
10.  for (j ← i; j ≤ |T| − minsup + i ∧ j ≤ |T|; j++) do
11.    if i ≡ 0 then
         matrix[i][j] ← 1;
         continue;
       endif
       if j ≡ i then
         matrix[i][j − 1] ← 0;
17.    endif
18.    item x ← postfix(X);
19.    matrix[i][j] ← matrix[i − 1][j − 1] ∗ Y.P[j] ∗ T[j][x]+
         matrix[i][j − 1] ∗ (1 − (Y.P[j] ∗ T[j][x]));
20.    if i = 1 ∨ j = |T| − minsup + i then
         X.P[j] ← Y.P[j] ∗ T[j][x];
       endif
     endfor
24.  if matrix[i][j − 1] < τ ∧ i ≠ 0 then
       return i − 1;
     endif
   endfor
28 return i − 1;
end
```

Figure 3: Probabilistic Support Function

$Sup_T(Z, \tau) < Sup_T(X, \tau)$. Therefore, X is an closed itemset according to Definition 2.

Using this property, the algorithm need only keep itemsets of length k and $k + 1$, when checking for closure.

The pseudocode for the `PFCIM` algorithm is shown in Figure 4. The algorithm begins by placing all items (singletons) in the set C (line 1). Next, at line 2, all itemsets X that are in set C and also have a $Sup_T(X, \tau)$ value greater than or equal to *minsup*, are placed in set L. At line 3, L is passed to the `Apriori-Gen` algorithm (from [2]), returning itemset candidates of size 2 which are placed in the set C'. Line 4 begins a `while` loop that continues until C' is empty. Starting at line 5 and ending at line 11, each itemset X in C' is checked to see if it is a probabilistic frequent itemset or not. This entails calling the function `ProbSup`, which returns the itemset's probabilistic support value. If MS is greater than or equal to *minsup*, the itemset is frequent, $Sup_T(X, \tau)$ is assigned to $X.MS$, and X is added to the set L'. Next (lines 12 through 23), each itemset X of length $k-1$ (found in set L) is compared to each itemset of length k (found in L'). If no superset of X in L is found in L', which has the same support of X, X is outputted as a closed probabilistic frequent itemset. Finally, at line 24, L is assigned the values of L', and then (line 25) `Apriori-Gen` is called again to generate candidates of length $k + 1$, which are assigned to C'.

5 Experimental Evaluation

The PFCIM algorithm was put through a series of experimental evaluations, which provide some idea of its computation costs. These tests were performed using well-known and available datasets[2], and were evaluated varying both of the independent variables *minsup* and τ. Figure 5 shows each of the datasets used in the evaluation and their characteristics ($|A|$ denotes the number of attributes). All datasets used were found at the Frequent Itemset Mining Dataset Repository <http://fimi.cs.helsinki.fi/data/>. This dataset repository contains well-known datasets that have been converted into itemset transactions. During this transformation, each possible item in the original datasets become their own attribute—this explains the large number of

[2]More information about the accidents dataset can be found in [7].

function `PFCIM`(int $minsup$, float τ)
begin
1. $C \leftarrow \{X | X \in I\}$;
2. $L \leftarrow \{X | X \in C \wedge$ `ProbSup`$(X, \tau, minsup) \geq minsup\}$;
3. $C' \leftarrow$ `Apriori-Gen`(L); // $X.MS$ is set for all $X \in L$
4. **while** $C' \neq \emptyset$ **do**
5. **foreach** $X \in C'$ **do**
   $MS \leftarrow$ `ProbSup`$(X, \tau, minsup, L)$;
   if $MS \geq minsup$ **then**
   $X.MS \leftarrow MS$;
   $L' \leftarrow L' \cup X$;
   endif
11. **endfor**
12. **foreach** $s \in L$ **do**
   $flag \leftarrow$ true;
   foreach $t \in L'$ **do**
   if $s \subset t \wedge s.MS \equiv t.MS$ **then**
   $flag \leftarrow$ false;
   break;
   endif
   endfor
   if $flag$ **then**
   - Output s as a probabilistic frequent
   closed itemset;
   endif
23. **endfor**
24. $L \leftarrow L'$;
25. $C' \leftarrow$ `Apriori-Gen`(L);
endwhile
end

function `Apriori-Gen`(L)
begin
  int $j \leftarrow |L[1]|$; //j is the size of the elements in L
  foreach $p, q \in L$ such that
    $p_{1\ldots(j-1)} \equiv q_{1\ldots(j-1)} \wedge p_j < q_j$ **do**
    $c \leftarrow p_{1\ldots(j-1)} p_j q_j$;
    if all $s \subset c$ such that $|s| \equiv j$, $s \in L$ **then**
      $C \leftarrow C \cup \{c\}$;
    endif
  endfor
  return C;
end

Figure 4: PFCIM Algorithm

attributes in the datasets shown in Figure 5.[3]

Dataset	$\|T\|$	$\|A\|$	Density
Accidents	$10,000$	310	10.94%
Chess	$3,196$	75	49.33%
Mushroom	$8,124$	119	19.33%
T10I4D100K	$10,000$	866	1.16%
T40I10D100K	$20,000$	941	4.21%

Figure 5: Experimental Dataset Characteristics

Each of the aforementioned datasets (even through they have been converted into itemset transactions) are still certain datasets. To transform these datasets into uncertain datasets, the following steps were taken (Figure 6(a) and 6(b) show an example certain dataset and a possible uncertain transformation of it, respectively):

1. A boolean valued matrix B is created of size $|T| \times |A|$;

[3]The first $10,000$ transaction were taken for the Accidents and T10I4D100K datasets, and $20,000$ for T40I10D100K.

2. If a certain item i_i ($1 \leq i \leq |A|$) is present in transaction t_j ($1 \leq j \leq |T|$), then $B[j][i]$ is marked as true, else false;

3. Each element i of $B[j]$ is checked. If i is marked true, then a random number is generated according to the beta distribution (denoted as r) with $\alpha = 5$ and $\beta = 1$, else a random $1 - r$ is generated. Item j is then outputted, followed by a ":" and the random number (either r or $1 - r$). Thus, the number before the ":" represents the item and the value after its corresponding existential probability;

4. Once all elements i of $B[j]$ have been enumerated, a new line is inserted into the new file, and step 3 is performed for $j + 1$. Finally, once all j have been enumerated the process is stopped;

TID	Itemset
1	1 2 4
2	2 6 7 9
3	3 7

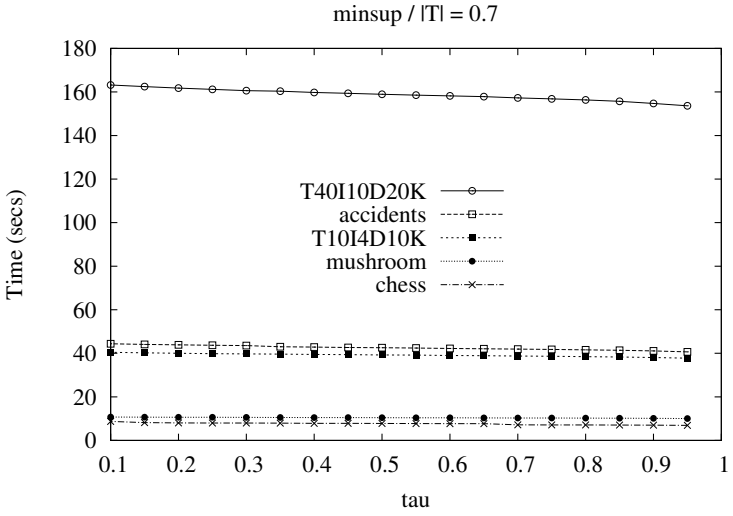
(a) Certain Dataset

TID	Itemset
1	1:0.89 2:0.99 3:0.12 4:0.78 5:0.03 6:0.1 7:0.11 8:0.03 9:0.14
2	1:0.12 2:0.89 3:0.04 4:0.05 5:0.13 6:0.91 7:0.88 8:0.03 9:0.91
3	1:0.04 2:0.11 3:0.8 4:0.2 5:0.02 6:0.02 7:0.99 8:0.05 9:0.2

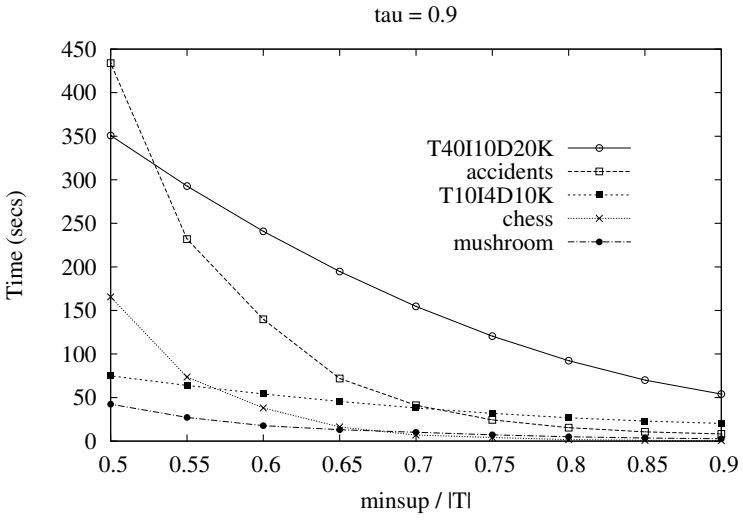
(b) Uncertain Dataset

Figure 6: Example Certain to Uncertain Dataset Transform

We believe the aforementioned database transformation procedure, produces an uncertain database from a certain one, that more resembeles to a real-world uncertain dataset. Figure 7(a) shows effect of τ on the execution time of the algorithm. One can see that execution time is linear with respect to τ. In Figure 7(b) the effect of *minsup* on the execution time of the algorithm is shown. For each dataset a *minsup* value was chosen relative to the number of transactions in it that would produce a $minsup/|T|$ value equal to the percentages shown. The results show that the execution time is exponential with respect to *minsup* (a result consistent with intuition and other studies of itemset mining).



(a) Effect of τ on Execution Time



(b) Effect of *minsup* on Execution Time

Figure 7: Performance Evaluations

6 Conclusions

In this paper, we defined probabilistic support and frequent closed itemsets based on it in uncertain databases for the first time. In addition, we proposed a probabilistic frequent closed itemset mining (PFCIM) algorithm to mine probabilistic frequent closed itemsets from uncertain databases. An experimental evaluation was given that displays some of the algorithm's execution complexities; which were performed on a variety of well-known real and synthetic datasets.

References

[1] C. Aggarwal, Y. Li, J. Wang, and J. Wang. Frequent pattern mining with uncertain data. *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, Jun 2009.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *Proc. 20th Int. Conf. Very Large Data Bases*, Jan 1994.

[3] T. Bernecker, H.-P. Kriegel, M. Renz, F. Verhein, and A. Zuefle. Probabilistic frequent itemset mining in uncertain databases. *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, Jun 2009.

[4] C. Chui and B. Kao. A decremental approach for mining frequent itemsets from uncertain data. *PAKDD'08, Proceedings of the 12th Pacific-Asia conference on Advances in knowledge discovery and data mining*, pages 64–75, Jan 2008.

[5] C. Chui, B. Kao, and E. Hung. Mining frequent itemsets from uncertain data. *Advances in Knowledge Discovery and Data Mining*, pages 47–58, 2007.

[6] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, Jul 2002.

[7] K. Geurts, G. Wets, T. Brijs, and K. Vanhoof. Profiling high frequency accident locations using association rules. *Proceedings of the 82nd Annual Transportation Research Board, Washington DC. (USA), January 12-16*, page 18pp, 2003.

[8] C. Leung, M. Mateo, and D. Brajczuk. A tree-based approach for frequent pattern mining from uncertain data. *PAKDD'08: Proceedings of the 12th Pacific-Asia conference on Advances in knowledge discovery and data mining*, pages 653–661, 2008.

[9] S. Wang, G. Wang, X. Gao, and Z. Tan. Frequent items computation over uncertain wireless sensor network. *Hybrid Intelligent Systems, 2009. HIS '09. Ninth International Conference on*, 2:223 – 228, 2009.

[10] Y. Xia, Y. Yang, and Y. Chi. Mining association rules with non-uniform privacy concerns. *DMKD '04: Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, Jun 2004.

[11] Q. Zhang, F. Li, and K. Yi. Finding frequent items in probabilistic data. *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, Jun 2008.

[12] D. Zhi-Feng, L. Yuan-Xiang, H. Guo-Liang, T. Ya-La, and S. Xian-Jun;. Uncertain data management for wireless sensor networks using rough set theory. *Wireless Communications, Networking and Mobile Computing, 2006. WiCOM 2006.International Conference on*, pages 1 – 5, 2006.

[13] E. Zimányi and A. Pirotte. Imperfect information in relational databases. *Uncertainty management in information systems: from needs to solutions*, pages 35–88, 1997.