Fast Approximation of Probabilistic Frequent Closed Itemsets *

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

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

ABSTRACT

In recent years, the concept of and algorithm for mining probabilistic frequent itemsets (PFIs) in uncertain databases, based on possible worlds semantics and a dynamic programming approach for frequency calculations, has been proposed. The frequentness of a given itemset in this scheme can be characterized by the Poisson binomial distribution. Further and more recently, others have extended those concepts to mine for probabilistic frequent closed itemsets (PFCIs), in an attempt to reduce the number and redundancy of output. In addition, work has been done to accelerate the computation of PFIs through approximation, to mine approximate probabilistic frequent itemsets (A-PFIs), based on the fact that the Poisson distribution can closely approximate the Poisson binomial distribution—especially when the size of the database is large. In this paper, we introduce the concept of and an algorithm for mining approximate probabilistic frequent closed itemsets (A-PFCIs). A new mining algorithm for mining such concepts is introduced and called A-PFCIM. It is shown through an experimental evaluation that mining for A-PFCIs can be orders of magnitude faster than mining for traditional PFCIs.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*; G.3 [**Probability and Statistics**]: Distribution functions

General Terms

Algorithms, Design, Performance

*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.

ACMSE'12, March 29–31, 2012, Tuscaloosa, AL USA.

Keywords

Probabilistic frequent closed itemset mining, uncertain databases, probabilistic support of itemset, approximation of poisson binomial distribution

1. INTRODUCTION

In light of recent applications, such as location-based systems and wireless sensor networks—where the data may be inherently uncertain—or, applications where the attributes can have probabilities associated with them, such as the probability a drug had an effect on a pathogen in a patient, new algorithms have been proposed that mine these uncertain databases for so-called probabilistic frequent itemsets (PFIs). Unlike traditional itemset mining, where each item is known for certain to either occur or not within a certain transaction, items in probabilistic databases have existential probabilities associated with them, which denote the probability that the item will occur within the uncertain transaction.

Previous attempts to mine frequent itemsets from uncertain databases made use of an itemset's expected support [4, 5]. However, the aforementioned methods, as pointed out in [2], cannot adequately express the frequentness of an itemset in an uncertain database. Thus, in [2], Bernecker et al. devised an exact method to mine for PFIs, using an itemset's support probability mass function (pmf), which follows a Poisson binomial distribution. That method used a dynamic programming approach to calculate an itemset's support pmf, and an Apriori enumeration scheme. Later in [1], the same team devised a method that used a generation function method for calculating the support pmf, and pattern-growth enumeration.

Other researchers [3, 8] however have proposed methods for approximating the Poisson binomial distribution. More specifically, Wang et al. [8], proposed a method that uses the Poisson distribution to approximate the Poisson binomial distribution, and thus were able to mine approximate probabilistic frequent itemsets (A-PFIs). The researchers were able to show that the speed of their devised method for calculating the support pmf of an itemset was orders of magnitude faster than the exact dynamic programming method. In addition, they showed the accuracy of such a method was very good through experimental evaluations.

It is a well-known problem in the area of frequent itemset mining, that the size of output can be very large, especially when user-defined thresholds are set low. It is this problem that has led to the development of algorithms which mine

a specific subset of all frequent itemsets, that results in a less redundant, and therefore, more succinct output. In [7], Tang et al. proposed an algorithm for mining probabilistic frequent closed itemsets (PFCIs), which mined for only those probabilistic frequent itemsets I, such that there is no other $I' \supset I$ where the probabilistic support of I does not equal that of I'. The algorithm disseminated to mine such concepts, like that of Bernecker et al. [2], uses a dynamic programming approach to calculate the support pmf of a given itemset, as well as Apriori candidate enumeration.

In this paper, preliminary material is presented (Section 2), including the concepts of the uncertain data model (under the assumption of possible world semantics) are examined and related to the Poisson binomial distribution (Section 2.1), as well as, how these concepts compare and contrast with traditional itemset mining. Next, the concepts of probabilistic frequent itemsets (PFIs), Section 2.2, and probabilistic frequent closed itemsets (PFCIs), Section 2.3, are defined. Then, in Section 3, the new concept of approximate probabilistic frequent closed itemsets (A-PFCIs) are introduced and defined, as well as, an algorithm to mine for these newly defined concepts. This new algorithm (A-PFCIM), is shown to be orders of magnitude faster than the exact dynamic programming approach under several datasets. Lastly, conclusions are drawn in Section 5.

2. PRELIMINARIES

2.1 Uncertain Data Model

As is the case with traditional itemset mining, a set of items $A = \{a_1, a_2, \ldots, a_m\}$ is defined, and an itemset I is defined to be a subset of A, i.e., $I \subseteq A$. Further, a set of transactions $T = \{t_1, t_2, \ldots, t_n\}$, constituting the uncertain database, is defined where each t_j $(1 \leq j \leq n)$ is a set of pairs $(a_i, Pr(a_i \in t_j))$, where a_i is an item and $Pr(a_i \in t_j) \in (0, 1]$[1] is the probability of a_i appearing in transaction t_j. (This is unlike traditional itemset mining, where each item is known with certitude if it exists within a certain transaction or not.) a_i is called an *uncertain item* if $Pr(a_i \in t_j) \in (0, 1)$. If for all items a_i, in all transactions t_j, $Pr(a_i \in t_j) = 1$, the uncertain data model degenerates to a certain one, and all previous traditional itemset mining algorithms apply. However, if at least one uncertain item appears in the database, i.e., $Pr(a_i \in t_j) \in (0, 1)$, we call the database an *uncertain database* and we must apply newer data mining techniques to mine for frequent uncertain itemsets. Figure 1 shows an example uncertain database. In that example three uncertain transactions $T = \{t_1, t_2, t_3\}$ are shown where $A = \{1, 2, 3\}$. From the same example, we can deduce that $Pr(1 \in t_3) = 0.9$, since within transaction t_3, there exists the double $(1, 0.9)$.

TID	Uncertain 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: Example Uncertain Database

When mining for PFIs under uncertain databases, the principles and theories of possible world semantics must be used, and all corollary theories and definitions must conform to such semantics. Under possible world semantics, for each uncertain item a_i in each uncertain transaction t_j, there exists a certain database or database instance (possible world) that contains item a_i and another which does not. A possible world is a certain databases where an item a_i in a transaction t_j either exists or does not with certainty. Thus, the number of possible worlds of an uncertain database $T = \{t_1, t_2, \ldots, t_n\}$ is $2^{u_1} \cdot 2^{u_2} \cdots 2^{u_n} = 2^{u_1+u_2+\cdots+u_n}$ where u_j is the number of uncertain items in transaction t_j. If the assumption of independence between the transactions in T, as well as the items in each transaction is made, the probability of each possible world w can be calculated as follows:

$$Pr(w) = \prod_{t \in T(w)} \left(\prod_{a \in t} Pr(a \in t') \cdot \prod_{a \notin t} (1 - Pr(a \in t')) \right) \quad (1)$$

where $T(w)$ is the certain database of world w, t is a certain transaction in $T(w)$, t' is the corresponding uncertain transaction in uncertain database T, and $Pr(a \in t')$ is the probability of item a in the uncertain transaction t'.

Given a traditional (certain) itemset database $T = \{t_1, t_2, \ldots, t_n\}$, the probability of itemset I occurring in transaction t_j is 1 if $I \subseteq t_j$, or 0 otherwise. Whereas in an uncertain itemset database, the probability of I occurring in t_j, denoted as $Pr(I \subseteq t_j)$, is the marginal probability of I occurring in t_j in all possible worlds, i.e.:

$$Pr(I \subseteq t_j) = \sum_{w \in W, I \subseteq t_j(w)} Pr(w) \quad (2)$$

where $t_j(w)$ is the transaction t_j of the certain database in possible world w.

It can be proved, using (1) and (2), that this marginal probability can be expressed without enumerating all possible worlds as [2]:

$$Pr(I \subseteq t_j) = \prod_{a \in I} Pr(a \in t_j)$$

In a traditional (certain) itemset database T, the support of itemset I in transaction t_j, denoted $Sup_{t_j}(I)$, is 1 if $I \subseteq t_j$ or 0 otherwise. The support of I over the entire database T, denoted $Sup_T(I)$ is:

$$Sup_T(I) = Sup_{t_1}(I) + Sup_{t_2}(I) + \cdots + Sup_{t_n}(I) \quad (3)$$

Notice (3) is equal to the number of transactions that contain I, i.e., $Sup_T(I) = |\{t_j | I \subseteq t_j \wedge t_j \in T\}|$.

However, in an *uncertain* database T, the support of I in transaction t_j is no longer a concrete 0 or 1, and instead is a random variable X_j^I following a Bernoulli distribution with parameter p_j, where $p_j = Pr(X_j^I = 1)$ and $1 - p_j = Pr(X_j^I = 0)$ (or the probability of success and failure, respectively); here each $p_j = Pr(I \subseteq t_j)$. Therefore, in an uncertain database T, the support of I over the entire database T is a random variable $X^I = \sum_{j=1}^{n} X_j^I$, which follows a Poisson binomial distribution with parameters $p_1, p_2, \ldots, p_n$—if the assumption of independence between transactions is made. The probability that $X^I = i$, $(0 \leq i \leq n)$, is:

[1]An existential probability of zero is not considered, since the item would simply not appear in the transaction.

$$Pr(X^I = i) = \sum_{A \in F_i} \prod_{j \in A} p_j \prod_{j \in \bar{A}} (1 - p_j)$$

where F_i is the set of distinct sets of i integers selected from $\{1, 2, \ldots, n\}$.

The above equation represents the probability mass function (pmf) of the random variable X^I. It can be also expressed as:

$$Pr(X^I = i) = \sum_{S \subseteq T, |S|=i} (\prod_{t \in S} Pr(I \subseteq t) \cdot \prod_{t \in T - S} (1 - Pr(I \subseteq t)))$$

2.2 Probabilistic Frequent Itemsets

With traditional (certain) itemset mining, given a database $T = \{t_1, t_2, \ldots, t_n\}$, an itemset I is considered frequent if and only if $Sup_T(I) \geq minsup$, where $minsup \in [0, n]$ is some user-defined threshold. In uncertain databases, we have only the support probability distribution of itemset I, which follows the Poisson binomial distribution.

In [2], Bernecker et al. proposed the concept of the *frequentness probability*[2] of an itemset I to be $Pr(X^I \geq i)$.

Thus, given $minsup$, an itemset I is said to be frequent with confidence $\tau \in [0, 1]$, if and only if, $Pr(X^I \geq minsup) \geq \tau$.

Definition 1. *Given an uncertain database T and an itemset I, I is a* probabilistic frequent itemset (PFI) *if and only if $Pr(X^I \geq minsup) \geq \tau$, where $minsup \in [0, n]$ and $\tau \in [0, 1]$ are user-defined thresholds. [2]*

2.3 Probabilistic Frequent Closed Itemsets

One persistent problem of itemset mining in general, is the production of too many and/or redundant frequent itemsets. Several techniques have been proposed to combat this problem, which include mining only the set of maximal frequent itemsets (M) or the set of closed frequent itemsets (C). After mining all closed frequent itemsets (C), it is possible to reproduce all frequent itemsets—with their respective supports, whereas it is not possible with M. However set C could be larger than set M. The relationship between the three sets is: $M \subseteq C \subseteq F$. The literature generally agrees that mining closed frequent itemsets is a good compromise.

In [7][3], the *probabilistic support* of itemset I with confidence $\tau \in [0, 1]$ within an uncertain database T (denoted as $PS_T(I, \tau)$) is defined as:

$$PS_T(I, \tau) = argmax_{i \in [0,n]}(Pr(X^I \geq i) \geq \tau) \quad (4)$$

It was proved in [7] that probabilistic support, $PS_T(I, \tau)$, is anti-monotonic with respect to I, .i.e., $PS_T(I, \tau) \leq PS_T(I')$ if $I \supset I'$. Based on this, a probabilistic frequent closed itemset (PFCI) is defined in Definition 2.

Definition 2. *Given an uncertain database T, user-defined thresholds $minsup \in [0, n]$ and $\tau \in (0, 1]$, I is a* probabilistic frequent closed itemset (PFCI) *if and only if $Pr(X^I \geq minsup) \geq \tau$ and there is no $I' \supset I$ such that $PS_T(I', \tau) = PS_T(I, \tau)$. [7]*

[2]The notation used in [2] is different from that presented in this paper. A more conventional notation is used here.

[3]In [7], Tang et al. uses slightly different notation for probabilistic support, that is, $Sup_T(X, \tau)$. We use $PS_T(I, \tau)$ to avoid confusing it with $Sup_T(I)$, previously mentioned.

3. FAST APPROXIMATION OF PFCIS

It is well-known that the Poisson distribution can closely approximate the Poisson binomial distribution—especially for large values of n and small values of $p_j = Pr(I \subseteq t_j)$, $(j = 1, 2, \ldots, n)$. As disseminated in [8][4], if one lets $X_1, X_2, \ldots, X_n$ be a set of Poisson trails, in which $Pr(X_j = 1) = p_j$ and $X = \sum_{j=1}^{n} X_j$, then the random variable X follows a Poisson binomial distribution. Further, let μ be the mean of X (i.e., $\mu = E[X] = \sum_{j=1}^{n} p_j$)—then $Pr(X = i)$ can be approximated by the probability mass function (pmf) of the Poisson distribution with the same mean μ. In particular we have the following:

$$Pr(X = i) \approx f(i, \mu) = \frac{\mu^i}{i!} \cdot e^{-\mu} \quad (5)$$

Thus, the cumulative density function (cdf) of the Poisson distribution $F(i, \mu) = \sum_{k=0}^{i} f(k, \mu)$, can approximate $Pr(X \leq i)$ or the probability that X is less than or equal to i as:

$$Pr(X \leq i) \approx F(i, \mu) = e^{-\mu} \sum_{k=0}^{i} \frac{\mu^k}{k!}$$

$F(i, \mu)$ increases with i because $F(i, \mu) = \sum_{k=0}^{i} f(k, \mu)$ and $f(k, \mu) = \frac{\mu^k}{k!} e^{-k} > 0$. It is proved in [8] that $F(i, \mu)$ decreases with μ, i.e., $\frac{\partial F(i,\mu)}{\partial \mu} < 0$ and $F(i, \mu) < F(i, \mu')$ if $\mu > \mu'$.

In Section 2.1, we established that the support of an itemset I in an uncertain database $T = \{t_1, t_2, \ldots, t_n\}$ is a random variable X^I following a Poisson binomial distribution with $p_j = Pr(I \in t_j) = \prod_{a \in I} Pr(a \in t_j)$. Therefore, the mean of X^I, denoted as μ^I, is:

$$\mu^I = \sum_{j=1}^{n} p_j = \sum_{j=1}^{n} \prod_{a \in I} Pr(a \in t_j) \quad (6)$$

Thus, we have $Pr(X^I = i) \approx f(i, \mu^I) = \frac{(\mu^I)^i}{i!} e^{-\mu^I}$ and $Pr(X^I \leq i) \approx F(i, \mu^I) = e^{-\mu^I} \sum_{k=0}^{i} \frac{(\mu^I)^k}{k!}$.

Now one can approximate the probability of an itemset I having a support greater than or equal i as:

$$Pr(X^I \geq i) = 1 - Pr(X^I \leq i - 1) \approx 1 - F(i - 1, \mu^I) \quad (7)$$

If one denotes $Q(i, \mu^I) = 1 - F(i - 1, \mu^I)$, then (7) can be re-formulated as:

$$Pr(X^I \geq i) \approx Q(i, \mu^I)$$

Thus, the probabilistic support of an itemset I with confidence $\tau \in [0, 1]$, $PS(I, \tau)$ (defined in (4)), can be approximated by the *approximate probabilistic support*, denoted as $\widehat{PS(I, \tau)}$, as follows:

$$\widehat{PS_T(I, \tau)} = argmax_{i \in [0,n]}(Q(i, \mu^I) \geq \tau) \quad (8)$$

Since the function $F(i, \mu)$ increases with i and decreases with μ, the function $Q(i, \mu) = 1 - F(i - 1, \mu)$ decreases with i and increases with μ, i.e., $\frac{\partial Q(i,\mu)}{\partial i} < 0$ and $\frac{\partial Q(i,\mu)}{\partial \mu} > 0$.

[4]The accuracy of the approximation was also studied by Wang et al., which included a mathematical bound of the error.

Let μ_i $(i = 0, \ldots, n)$ be the real numbers satisfying $Q(i, \mu_i) = \tau$. That is,

$$Q(0, \mu_0) = Q(1, \mu_1) = \cdots = Q(n, \mu_n) = \tau \quad (9)$$

We can prove that $\mu_0 < \mu_1 < \cdots < \mu_n$ as follows. Since the function $Q(i, \mu)$ decreases with i, we have

$$Q(i+1, \mu_i) < Q(i, \mu_i)$$

for each $i \in [0, n)$. Because $Q(i, \mu_i) = Q(i+1, \mu_{i+1})$ (according to (9)), we have

$$Q(i+1, \mu_i) < Q(i+1, \mu_{i+1}) \quad (10)$$

Since the function $Q(i, \mu)$ increases with μ, we must have

$$\mu_i < \mu_{i+1}$$

for each $i \in [0, n)$. This is because, if $\mu_i \geq \mu_{i+1}$, we would have $Q(i+1, \mu_i) \geq Q(i+1, \mu_{i+1})$, which would contradict (10).

This is summarized in Theorem 1.

THEOREM 1. *Given a confidence threshold $\tau \in [0, 1]$ and μ_i such that $Q(i, \mu_i) = \tau$, then $\mu_i < \mu_{i+1}$ for $i = 0, \ldots, n-1$.*

Theorem 1 allows us to calculate $\widehat{PS(I, \tau)}$ for an itemset I as follows:

If μ^I (calculated using (6)) satisfies $\mu_i \leq \mu^I < \mu_{i+1}$ for an $i \in [0, n]$, then we have:

$$\tau = Q(i, \mu_i) \leq Q(i, \mu^I) \quad (11)$$

because function $Q(i, \mu)$ increases with μ. In addition, we also have the following—for the same reason:

$$Q(i+1, \mu^I) < Q(i+1, \mu_{i+1}) = \tau \quad (12)$$

Then, the combination of (11) and (12), shows that i is the largest value such that $Q(i, \mu^I) \geq \tau$. That proves that $\widehat{PS(I, \tau)} = i$, according to (8).

To determine whether an itemset I is a approximate probabilistic itemset, we only need to pre-calculate μ_i for $i = minsup, \ldots, n$, that satisfies $Q(i, \mu_i) = \tau$. Then, μ^I for itemset I is calculated. If $\mu^I < \mu_{minsup}$, then $\widehat{PS(I, \tau)} < minsup$ and I is not frequent. If $\mu^I \geq \mu_{minsup}$, we find the largest $i \in [minsup, n]$, such that $\mu_i \leq \mu^I$ and assign that value of i to $\widehat{PS(I, \tau)}$. Figure 2 shows the function `CalcApproxProbSup`, which does just that. In particular μ^I is calculated in lines 1–8. The uncertain database T is denoted as a matrix at line 5, where $T[j][a]$ accesses $P(a \in t_j)$

Figure 3 shows the main algorithm (`A-PFCIM`) which uses Apriori enumeration (i.e., breadth-first) to mine all A-PFCIs. Within it, each itemset I has associated with it it's approximate probabilistic support PS, and is accessed through $I.PS$. For completeness, Figure 4 displays the classic Apriori algorithm which is called from the `A-PFCIM` method.

```
function CalcApproxProbSup(itemset I)
1. float μ^I ← 0;
2. foreach transaction j ∈ T do
3.     float product ← 1;
4.     foreach a ∈ I do
5.         product ← product · T[j][a];
6.     end foreach
7.     μ^I ← μ^I + product;
8. end foreach
9. if μ^I < μ_minsup then
       return −1
   else
12.    for i = minsup + 1 to n do
           if μ^I < μ_i then
               return i − 1;
           end if
16.    end for
   end if
end function
```

Figure 2: Function for Calculating Approx. Probabilistic Support

```
procedure A-PFCIM(int minsup)
   C ← {a|a ∈ A};
   L ← {a|a ∈ C∧ CalcApproxProbSup(a) ≥ minsup};
   C′ ← AprioriGen(L);            // a.PS is set for all a ∈ L
   while C′ ≠ ∅ do
      foreach I ∈ C′ do
         PS ← CalcApproxProbSup(I);
         if PS ≥ minsup then
            I.PS ← PS;
            L′ ← L′ ∪ I;
         end if
      end foreach
      foreach I ∈ L do
         flag ← true;
         foreach I′ ∈ L′ do
            if I ⊂ I′ ∧ I.PS ≡ I′.PS then
               flag ← false;
               break;
            end if
         end foreach
         if flag then
            - Output s as a probabilistic frequent
              closed itemset;
         end if
      end foreach
      L ← L′;
      L′.clear();
      C′ ← AprioriGen(L);
   end while
   if L ≠ ∅ then
      - Output each I ∈ L as a prob. frequent closed itemset;
   end if
end procedure
```

Figure 3: A-PFCIM Algorithm

4. EXPERIMENTAL EVALUATION

In this section, an experimental evaluation of the A-PFCIM algorithm is disseminated. More specifically, the A-PFCIM algorithm is compared to the dynamic programming approach proposed in [2]. The two algorithms are compared using various real datasets and combinations of user-defined variables. All datasets used in this evaluation were taken from the Frequent Itemset Mining Dataset Repository <http://fimi.ua.ac.be/data/>[5]. However, since the datasets

[5]Information on the accidents dataset can be found in [6].

```
function AprioriGen(L)
   int k ← the size of the elements in L
   foreach I, I' ∈ L such that
      I_{1...(k-1)} ≡ I'_{1...(k-1)} ∧ I_k < I'_k do
      c ← I_{1...(k-1)} I_k I'_k;
      if all s ⊂ c such that |s| ≡ k, s ∈ L then
         C ← C ∪ {c};
      end if
   end foreach
   return C;
end function
```

Figure 4: AprioriGen Function

are exact or certain, transforming them into uncertain datasets was required. The procedure used to perform this transformation was done as follows: for each certain item in a transaction, the item is copied to the new uncertain dataset; a random probability $p \in (0, 1]$ is then chosen from the beta distribution with parameters $\alpha = 5$ and $\beta = 1$ for this item; finally, p is assigned to the item with probability 1/2 and $1-p$ is assigned with probability 1/2. This method of transforming a certain itemset database into an uncertain one, is different from other methods, in which the probability p is drawn from a uniform distribution [2, 1], or from a normal distribution [8]. We believe drawing probabilities from the beta distribution gives a possibly better representation of a real-world dataset, in which items are close either to existing or not, rather than being uniformly random (uniform distribution), or "ho-hum" average (normal distribution).

In Figure 5 displays the execution times culled from experiments on several real and synthetic datasets. Specifically, Figure 5(a) shows the execution time as a function of $minsup/n$ using the A-PFCIM algorithm and Figure 5(b) shows the same but using the dynamic programming approach. For each of the two algorithms in Figures 5(a) and 5(b), τ is held constant at 0.9[6]. Note that the T10I4D100K dataset seems to be largely unaffected by changes in $minsup/n$. The explanation, is one of dataset characteristics. The
T10I4D100K dataset is large and sparse and thus even for small values of $minsup/n$, the number of A-PFCIs are very small.

Figure 6 displays the execution times of the two algorithms as a function of τ. In both experiments (Figure 6(a) and Figure 6(b)) $minsup/n$ is set constant at 0.35.

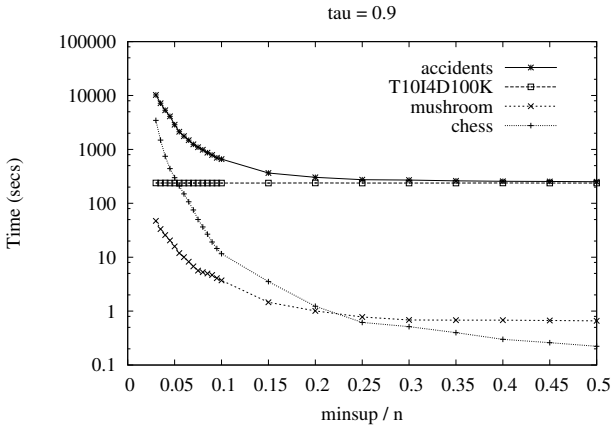
Next in Figure 7, we examine the amount of speedup, in terms of execution time, of the approximation method when compared to the dynamic one when varying $minsup/n$ (Figure 7(a)) and varying τ (Figure 7(b)) over all the experimental datasets. Speedup is defined as:

$$Speedup = \frac{Time_{Dynamic}}{Time_{Approx}} \tag{13}$$

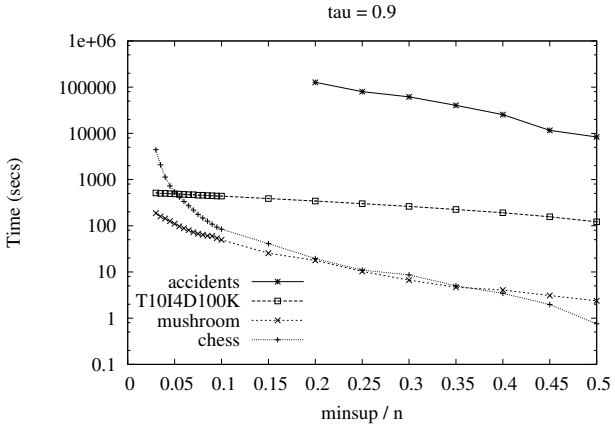
where $Time_{Dynamic}$ is the execution time in seconds of the dynamic approach, and $Time_{Approx}$ is the execution time in seconds of the approximation approach.

We find that when looking at Speedup, and varying

[6]In the case of the Dynamic algorithm, values of $minsup/n$ that are less than 0.2 for the accidents dataset, are not tested because the execution times involved are too onerous.



(a) Approximation Method (A-PFCIM)



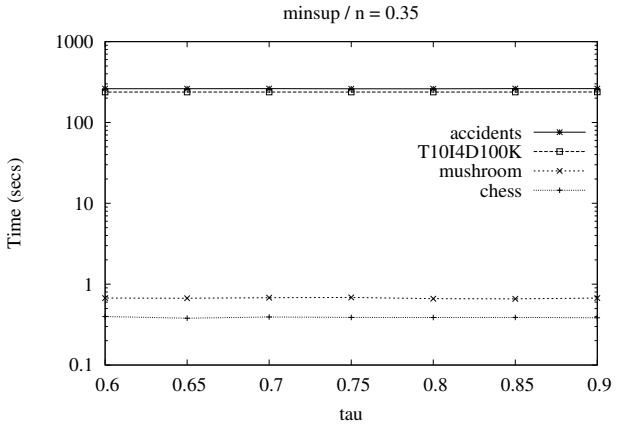
(b) Dynamic Method

Figure 5: Execution Time When Varying $minsup/n$

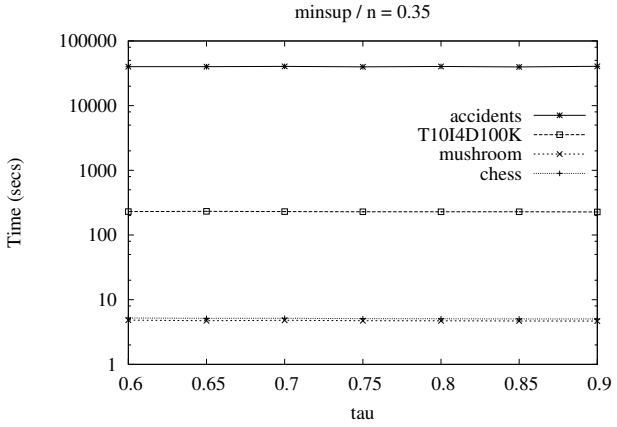
$minsup/n$ (Figure 7(a)), one can observe that Speedup increases as $minsup/n$ decreases for values between 0.2 and 0.5, for the accidents and T10I4D100K datasets. For the T10I4D100K dataset is continues to increase for lower values. However, for the chess and mushroom datasets—after initially going higher for lower values of $minsup/n$—once the value reaches approx. 0.25 for the chess dataset and 0.15 for the mushroom, the Speedup starts to decrease. And this is of course when the number of itemset candidates increases. Further, one can see that when $minsup/n$ is set relatively high (e.g., 0.35 and greater), the dynamic slightly outperforms the approximation method on the synthetic dataset T10I4D100K.

When looking at Speedup and varying τ, we find that Speedup remains relatively flat over the experimental values and datasets. Again, we find the synthetic dataset
T10I4D100K slightly outperforms the dynamic approach, which is not surprising given that $minsup/n = 0.35$ and what we found in Figure 7(a).

From this experimental evaluation we see that in most cases the approximation method (A-PFCIM) performs the mining of probabilistic frequent closed itemsets using approximation, orders of magnitude faster than the exact dynamic approach.

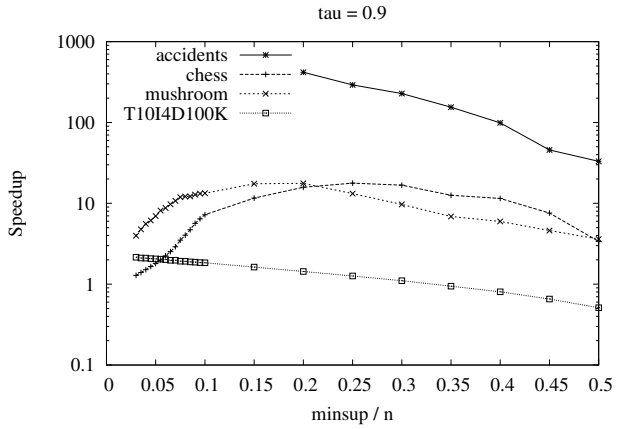


(a) Approximation Method (A-PFCIM)

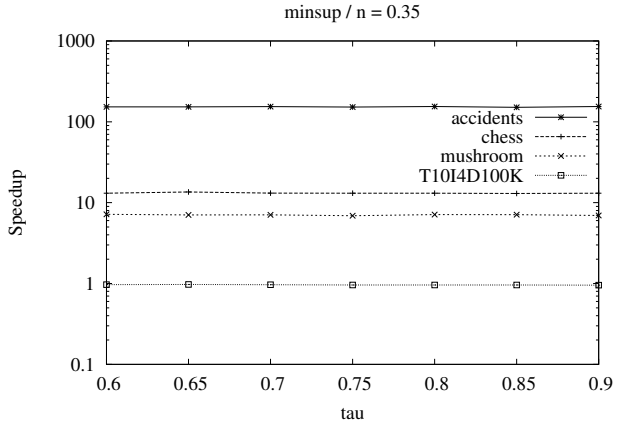


(b) Dynamic Method

Figure 6: Execution Time When Varying τ



(a) Speedup Varying $minsup/n$



(b) Speedup Varying τ

Figure 7: Speedup of Execution Time

5. CONCLUSION

This paper disseminated a new algorithm for mining approximate probabilistic frequent closed itemset (A-PFCIs). The algorithm (A-PFCIM) still conforms to possible world semantics, but uses the Poisson distribution to approximate the Poisson binomial distribution, as a means to approximate the probability mass function of the support of an itemset in an uncertain database. The experimental evaluation given shows that the approximation method can in most cases, mine probabilistically frequent closed itemsets (PFCIs) orders of magnitude faster than an existing exact method.

6. REFERENCES

[1] T. Bernecker, H.-P. Kriegel, M. Renz, and F. Verhein. Probabilistic Frequent Pattern Growth for Itemset Mining in Uncertain Databases (Technical Report). *arXiv.org*, cs.DB, Aug. 2010.

[2] 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*, pages 119–127, June 2009.

[3] T. Calders, C. Garboni, and B. Goethals. Approximation of Frequentness Probability of Itemsets in Uncertain Data. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 749–754, 2010.

[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] 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*, pages 1–18, 2003.

[7] P. Tang and E. A. Peterson. Mining Probabilistic Frequent Closed Itemsets in Uncertain Databases . In *ACM-SE 49: Proceedings of the 49th Annual ACM Southeast Regional Conference*, pages 86–91. ACM, 2011.

[8] L. Wang, R. Cheng, S. D. Lee, and D. Cheung. Accelerating probabilistic frequent itemset mining: a model-based approach. In *CIKM '10: Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 429–438. ACM Request Permissions, Oct. 2010.