Received: 30 April 2019 | Revised: 20 January 2020 | Accepted: 6 March 2020

DOI: 10.1111/exsy.12551

SPECIAL ISSUE PAPER

Expert Systems WILEY

Efficient weighted probabilistic frequent itemset mining in uncertain databases

Zhiyang Li[1] | Fengjuan Chen[1] | Junfeng Wu[2] | Zhaobin Liu[1] | Weijiang Liu[1]

[1]Information Science and Technology College, Dalian Maritime University, Dalian, China

[2]School of Information Engineering, Dalian Ocean University, Dalian, China

Correspondence
Zhaobin Liu, No. 1, Linghai Road, Ganjingzi District, Dalian, China.
Email: zhbliu@gmail.com

Funding information
China Postdoctoral Science Foundation, Grant/Award Number: 2018M640239; Liaoning Provincial Nature Science Foundation of China, Grant/Award Number: 2019-MS-028; National Nature Science Foundation of China, Grant/Award Numbers: 61672379, 61370198, 61300187, 61370199

Abstract

Uncertain data mining has attracted so much interest in many emerging applications over the past decade. An issue of particular interest is to discover the frequent itemsets in uncertain databases. As an item would not appear in a transaction of such database for certain, several probability models are presented to measure the frequency of an itemset, and the frequent itemset over probabilistic data generally has two different definitions: the expected support-based frequent itemset and probabilistic frequent itemset. Meanwhile, it is noted that the frequency itself cannot identify useful or meaningful patterns in some scenarios. Other measures such as the importance of items should be also taken into account. To this end, some studies recently have been done on weighted (importance) frequent itemset mining in uncertain databases. However, they are only designed for the expected support-based frequent itemset, and suffer from low efficiency due to generating too many frequent itemset candidates. To address this issue, we propose a novel weighted probabilistic frequent itemsets (w-PFIs) algorithm. Moreover, we derive a probability model for the support of a w-PFI candidate in our method and present three pruning techniques to narrow the search space and remove the unpromising candidates immediately. Extensive experiments have been conducted on both real and synthetic datasets, to evaluate the performance of our w-PFI algorithm in terms of runtime, accuracy and scalability. Results show that our algorithm yields the best performance among the existing algorithms.

KEYWORDS

probability model, pruning, uncertain database, weighted probabilistic frequent itemset

1 | INTRODUCTION

Recently, we have witnessed a rapid growth of uncertain data in emerging applications such as online market analysis, Radio Frequency Identification-based monitoring, traffic data analysis and location-based service (Aggarwal, Li, Wang, & Wang, 2009; Fournier Viger et al., 2017; Sun, Cheng, Cheung, & Cheng, 2010; Yang et al., 2020). Uncertain data mining has thus become an active research direction in the data mining community. A widely studied topic in certain data mining is to discover the itemset that occurs in at least a given number of transactions, aka frequent itemset (FI) mining. However, this is challenging when deal with uncertain data, since we generally cannot be sure whether an itemset occurs in a transaction or not (Aggarwal & Yu, 2009; Bernecker, Kriegel, Renz, Verhein, & Zuefle, 2009; Yang, Chao, & Shen, 2017).

Table 1 gives an example of uncertain data consisted of some customer behaviour records in a supermarket. There is a value associated with each item in the table, denoting the likelihood of a customer to buy this item in the future. The associated values are estimated by the purchasing history of the customers in this supermarket. Supposing Tom visited the market 10 times last month, and purchased milk four times, then it can

TABLE 1 An uncertain database

Customer	Transaction (item, probability)
Tom	(Milk, 0.4), (Fruit, 1.0), (Video, 0.3)
Lucy	(Milk, 1.0), (Fruit, 0.8)

be estimated that Tom will have a 40% chance to purchase milk when he visits the market in the future. It is obviously useful to mine FI in such database to understand customer behaviours. However, the likelihood value associated with the item makes most certain data mining techniques inapplicable or inaccurate.

To overcome this problem, a direct way is to binarize the uncertain database into certain database (binary database) by a threshold, and then perform classic FI mining method by counting the transactions that contain a candidate itemset (Agrawal, Imieliński, & Swami, 1993; Han, Pei, & Yin, 2000). Unfortunately, it will lose inherent information and lead to inaccuracies. Recent studies on FI mining in uncertain database are classified into two categories, both of which consider the support of an itemset as a random variable. One uses the expectation of the support to determine whether the itemset is frequent, referred as expected support-based frequent mining methods (Calders, Garboni, & Goethals, 2010a; Calders, Garboni, & Goethals, 2010b; Chui, Kao, & Hung, 2007; Leung & MacKinnon, 2014). The other uses the frequentness probability of the support instead, called as probabilistic frequent mining methods (Li, Zhang, & Zhang, 2017; Liu, Chen, & Zhang, 2013; Peterson & Tang, 2013). Generally speaking, the latter captures confidence information of the support and thus is more accurate and useful. However, the former is more efficient and economic. Recently, some research have shown an interesting fact that both two kinds of FIs have a rather close relation, and pointed out that the latter methods can be accelerated by some approximation techniques based on classic probability models (Tong, Chen, Cheng, & Yu, 2012).

The above FI mining methods in uncertain databases still suffer from an important limitation. They all treat the items in the transactions with the same importance. It will lead to inaccuracies sometimes since the importance of items may vary in certain situations. Take Table 1 for example. Supposing the fruit in the supermarket is overstocked, the supermarket manager may wonder which items are usually purchased accompanying with fruit by customers, to make a co-promotion. In this situation, if assign the item fruit a higher importance, more FIs containing fruit will be obtained by a carefully designed mining model. Meanwhile, FIs without fruit may be ignored during the mining process, which will help the manager to recognize the inherent information related to fruit better. Thus, in some scenarios, the weight of items can be set by the users according to their domain knowledge to improve the mining accuracy or satisfy their specific requirement (Lee, Yun, & Ryang, 2015).

Motivated by the fact that items may have different importance, the problem of weighted frequent itemsets (w-FIs) mining has been proposed and widely studied (Tao, Murtagh, & Farid, 2003; Vo, Coenen, & Le, 2013; Yun, 2008). However, the early solutions are focused on certain database. More recently, some studies have shown interest on w-FI mining over uncertain databases. A big challenge for w-FI mining is that the anti-monotonicity (aka downward closure) property widely employed in FI mining would not hold any longer. That is to say, a weighted infrequent itemset may have a weighted frequent superset due to the different weights of the items. Thus, the search space has to be explored deeper to make sure all the w-FIs are visited, which will greatly increase the overhead of the mining algorithm. To handle it, Lin, Gan, Fournier-Viger, Hong, and Tseng (2016) make an attempt to maintain the anti-monotonicity via the maximal weight of all items, named high upper-bound expected weighted downward closure (HEWI). Noticing that HEWI may generate too many candidates, Zhao, Zhang, Pan, Chen, and Sun (2018) present a weight judgement downward closure property (WD) to early prune the search space and improve the time efficiency. However, the current studies on w-FI mining over uncertain databases still suffer from the following two limitations:

- There are two kinds of FIs in uncertain databases as mentioned. The existing w-FI mining algorithms are only designed for the expected support-based FI. It may be because the expectation of the support is more simple and can be efficiently estimated through scanning the database once. The other one probabilistic FI is proved to be more useful and accurate. It is therefore necessary to do extensive studies on weighted probabilistic frequent itemset (w-PFI) mining. However, the studies on it are very limited.
- The two methods HEWI and WD for weighted expected support-based FI mining make attempts to maintain the anti-monotonicity property to narrow the search space but still return too many non-weighted FIs as the candidates. The lack of effective pruning techniques will undoubtedly limit their applications. It will be worse when handling probabilistic FI mining, which is proved to be more time-consuming than expected support-based FI mining.

In general, the existing methods cannot yet provide entirely satisfactory solutions to weighted FI mining in uncertain databases, especially when aiming at w-PFI mining. To address this issue, we propose a novel w-PFI discovery algorithm over uncertain databases. The algorithm is designed in a breadth-first search manner with a candidate generate and test paradigm. We also give three pruning techniques to narrow the search space and remove the unpromising candidates at once. To sum up, the principal contributions of this paper are given as follows:

- We find the anti-monotonicity property for w-PFI mining can be inherited by an extent of the WD method. Then, we present a novel pruning technique for the candidate generate and test paradigm, which is more efficient than the pruning method adopted in WD.
- We derive a probability model for the support of a w-PFI candidate in our candidate generation method. Based on this model, we then further prune the candidates which are not true w-PFIs in probability. The amount of pruned candidates can be intuitively controlled by a parameter which is easy to be tuned in practice.
- We propose a novel w-PFI mining algorithm based on the above pruning techniques. The algorithm can be implemented for both exact w-PFI mining and approximate w-PFI mining. Meanwhile, the approximate algorithm is more efficient and is scalable to large uncertain databases.
- Extensive experiments have been conducted on both real and synthetic datasets, to evaluate the performance of our proposed w-PFI mining algorithm in terms of runtime, accuracy and scalability.

The rest of this paper is organized as follows: Section 2 describes the problem of w-PFI mining and then presents our solution. Section 3 evaluates the performance of the proposed w-PFI algorithm on four public datasets. Finally, we conclude in Section 4.

2 | PROBLEM DEFINITION

In this section, we first introduce preliminaries and state the problem of discovering w-PFI over uncertain databases. Then we present our algorithms for both exact w-PFI mining and approximate w-PFI mining. Table 2 summarizes the notations used in this paper.

2.1 | Preliminaries

In our data model, an uncertain dataset DB consists of a set of transactions DB = $\{T_1, T_2, ..., T_n\}$. Each transaction contains a subset of an itemset $I = \{I_1, I_2, ..., I_m\}$. Different from the classic certain (aka deterministic) database, there is an existential probability for each item I_i in the transaction T_j of DB, $p_{ij} = \Pr(I_i \subseteq T_j)$. A formal definition of the uncertain dataset DB is given as follows.

Definition 1 (Uncertain dataset). An uncertain dataset DB is a function from the domain $T \times I$ to the interval [0, 1], where T is a transaction identifier set, and I is an itemset.

Besides the existential probability, each item in our model is also assigned a real-valued weight to indicate its importance. Due to its simplicity in database design, the weight of an item is non-negative and remains the same across the different transactions, as employed by most existing methods. Hence, a weight table $w = \{w(I_1), w(I_2), ..., w(I_m)\}$ is given in our model.

Table 1 in Section 1 gives an example of our uncertain dataset with two transactions and three items. The item {Milk} exists in transaction T_1 (Tom) with a probability of 0.4. Table 3 gives a weight table of this uncertain dataset. In real-world applications, the weight of items is usually set

TABLE 2 Summary of the notations

Notation	Description
DB	An uncertain dataset of size n
T	A set of transaction identifiers
I	An itemset of size m
p_{ij}	An existential probability for item I_i appearing in the transaction T_j
w	A weight table for the itemset I. Each item has a real-valued weight
esup	An integer between (0, n], the minimum support for expected support-based FI mining
msup	An integer between (0, n], the minimum support for probabilistic FI mining
t	A real value between (0, 1], the probabilistic frequent threshold for probabilistic FI mining
α	A real value between [0, 1], the scale factor

TABLE 3 The weight table

Item	Milk	Fruit	Video
Weight	0.4	0.9	0.6

by users to indicate their relative importance, for example, interestingness, profit. From Table 3, the item {Fruit} has the greatest weight, indicating that it has more importance over other two items in the following mining task.

The existential probability of the item indicates the uncertainty in the database, which makes traditional techniques inapplicable. To interpret the uncertain transactions well, the Possible World Semantics (PWS) model is often adopted (Bernecker et al., 2009). We give the definition of possible world in Definition 2, from which a possible world is actually a certain form of the uncertain database DB. As each item occurs in DB with a probability, there would be an exponential number of possible worlds for DB.

Definition 2 (Possible world). For a given uncertain dataset DB, $T \times I \rightarrow [0, 1]$, a possible world W_i is a subset of the domain $T \times I$, with an existential probability of

$$\Pr(W_i) = \prod_{<T_s, I_t> \in W_i} \Pr(I_t \in T_s) \prod_{<T_s, I_t> \in W_i} (1 - \Pr(I_t \in T_s)).$$

For the dataset in Table 1, Table 4 lists all its possible worlds and their existential probabilities. Take the possible world W_5, for example. In this world, Tom purchases two items {Fruit, Video} and Lucy buys only one item {Milk}. Thus, the probability of this world is 0.036, calculated by $(1 - 0.4) \times 1.0 \times 0.3 \times 1.0 \times (1 - 0.8)$. With the PWS model, it seems more easy to measure the support (frequency) of an itemset, which is fundamental in FI mining. In the literature, there are two definitions of FI based on PWS, which are respectively given in Definition 3 and Definition 4. Both definitions consider the support of an itemset as a random variable sampled in the possible worlds.

Definition 3 (Expected support-based frequent itemset). Given uncertain dataset DB and a minimum expected support, esup, a set of items, or an itemset $X \subseteq I$ is an expected support-based frequent itemset if and only if $\sum_{i=1}^{|W|} \Pr(W_i) S(X, W_i) \geq \text{esup}$. $S(X, W_i)$ is the support (occurrence) of the itemset X in the possible world W_i.

For an itemset X, this definition employs the mean of its support to measure the frequency because the mean is an important statistics of its support. Besides the expected support, one can also obtain the probability mass function of the support (s-pmf). Figure 1 shows the s-pmf of itemset X = {Milk, Fruit} in Table 4. For example, $\Pr(\sup(X) = 2) = 0.32$ is calculated by the sum of the existential probabilities of possible worlds W_4 and W_8. With the s-pmf of an itemset, the other kind of FI is defined as follows.

Definition 4 (Probabilistic frequent itemset). Given uncertain dataset DB, a minimum support, msup, and a probabilistic frequent threshold t, an itemset $X \subseteq I$ is a probabilistic frequent itemset if and only if $\Pr(\sup(X) \geq \text{msup}) \geq t$.

Here, $\Pr(\sup(X) \geq \text{msup})$ is obtained by the s-pmf of itemset X over the possible worlds, which follows:

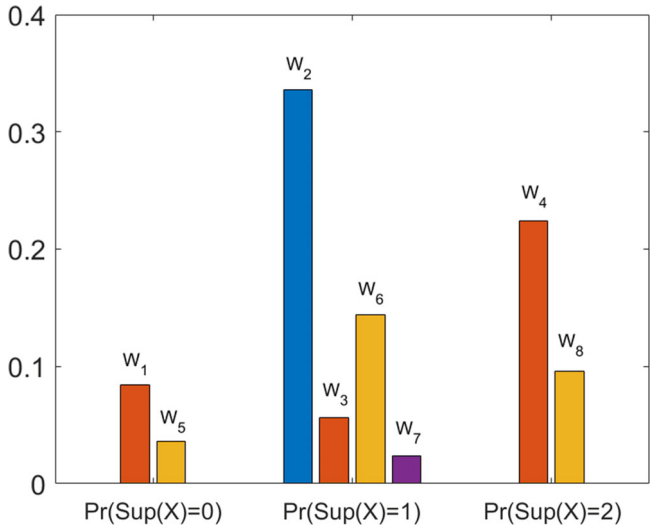
$$\Pr(\sup(X) \geq \text{msup}) = \sum_{\substack{W_i \in W \\ S(X, W_i) \geq \text{msup}}} \Pr(W_i). \quad (1)$$

In practice, it is not feasible to instantiate all the possible worlds to calculate $\Pr(\sup(X) \geq \text{msup})$. An equivalent definition of the s-pmf is given as follows:

TABLE 4 The possible worlds and their existential probabilities

W	Tuples in W	Probability
W_1	{Fruit}, {Milk}	0.084
W_2	{Fruit}, {Milk, fruit}	0.336
W_3	{Milk, Fruit}, {Milk}	0.056
W_4	{Milk, Fruit}, {Milk, Fruit}	0.224
W_5	{Fruit, Video}, {Milk}	0.036
W_6	{Fruit, Video}, {Milk, Fruit}	0.144
W_7	{Milk, Fruit, Video}, {Milk}	0.024
W_8	{Milk, Fruit, Video}, {Milk, Fruit}	0.096

FIGURE 1 Support pmf of the itemset X = {Milk, Fruit}



$$\Pr(\sup(X) \geq \text{msup}) = \sum_{\substack{T' \subseteq T \\ |T'| \geq \text{msup}}} \prod_{t \in T'} \Pr(X \subseteq t) \prod_{t \in T - T'} (1 - \Pr(X \subseteq t)). \tag{2}$$

In Equation (2), $|T'| \geq$ msup means there are at least msup itemset X in the transaction set T'. Compared to Equation (1), the computation of Pr(sup(X) ≥ msup) in Equation (2) is more easy. It can be efficiently solved by dynamic programming method or divide-and-conquer method (Sun et al., 2010). Here, we do not discuss the solution in detail due to the limited space.

As mentioned, a weight table can be introduced to give the different importance of items (e.g., Table 3), in order to improve the mining accuracy or satisfy a specific requirement. In the following, we discuss the problem of discovering w-PFI in uncertain databases. Formally, we give the definition of w-PFI.

Definition 5 (Weighted probabilistic frequent itemset). Given an uncertain dataset DB, a weight table w, a minimum support msup, and a probabilistic frequent threshold t, an itemset $X \subseteq I$ is a weighted probabilistic frequent itemset if and only if $w(X)\Pr(\sup(X) \geq \text{msup}) \geq t$.

Here, $w(X)$ is the weight of the itemset X. In our paper, $w(X)$ is calculated by the average weight of the items in X, which is widely adopted in the weighted frequent item mining (Lin et al., 2016; Zhao et al., 2018).

Definition 6 (Itemset weight). For an itemset $X \subseteq I$, the weight of X is the average weight of the items in the itemset X,

$$w(X) = \frac{1}{|X|} \sum_{x \in X} w(x).$$

2.2 | Exact weighted probabilistic frequent algorithm

In this section, we discuss how to mine the w-PFI efficiently. From Definition 5 and Definition 6, it is easy to obtain the equivalence between the probabilistic frequent itemset (PFI) and the w-PFI.

Theorem 1 (Equivalence between PFI and w-PFI). *An itemset X is a w-PFI under a probabilistic frequent threshold t, if and only if the itemset X is a PFI under a probabilistic frequent threshold $t/w(X)$.*

According to Theorem 1, it seems that any PFI mining algorithm can be directly extended to w-PFI mining. However, it is not true in practice. Since the number of candidate itemsets is exponential, the efficiency of PFI mining algorithm mainly depends on the anti-monotonicity (aka downward closure) property of PFI. Specially, the anti-monotonicity property tells that any subset $X' \subseteq X$ is a PFI if the itemset X is a PFI. Thus, one can avoid examining an itemset if one of its sub-itemset is not a PFI. It will efficiently prune the search space of PFI mining algorithms.

Theorem 2 (Anti-monotonicity property for PFI). *If an itemset X is a PFI, then any itemset $X' \subseteq X$ is also a PFI.*

Unfortunately, it is easy to find a counter-example of anti-monotonicity property in weighted PFI mining due to the fact that the itemsets have different weights. For example, let msup = 2 and t = 0.2, the itemset X = {Milk, Fruit} in Table 1 is a w-PFI because $w(X)\Pr(\sup(X) \geq \text{msup}) = 0.208$. However, one of its subset X' = {Milk} is not a w-PFI since $w(X')\Pr(\sup(X') \geq \text{msup}) = 0.16$. Thus, the classic anti-monotonicity property in Theorem 2 does not hold any longer in the w-PFI. Without the anti-monotonicity, one has to examine a candidate FI sizing up to exponential order, which is infeasible in practice. To overcome this problem, we present a novel anti-monotonicity property for w-PFI mining by extending the WD property designed for expected support-based FIs (Zhao et al., 2018).

Theorem 3 (Anti-monotonicity property for weighted PFI). *If an itemset X is a w-PFI, $|X| = k$, then there is at least one itemset $X' \subseteq X$, $|X'| = k - 1$ is a w-PFI.*

Proof. Let the itemset $X = \{I_1, I_2, \ldots, I_k\}$, its weight table $w = \{w(I_1), w(I_2), \ldots, w(I_k)\}$ and I_s is the item with the smallest weight. Then the itemset $\hat{X} = X - I_s$ is one of the sub-itemset of X. It is obvious that the following inequality holds $w(I_s) \leq w(X) \leq w\left(\hat{X}\right)$. The weight of an itemset is defined as the average weight of its members according to Definition 6.

Given a minimum support msup, and a probabilistic frequent threshold t, we might as well suppose the theorem is not correct, that is, all the subset itemset including $\hat{X}$ is not a weighted PFI. Thus, we will obtain $w\left(\hat{X}\right)P\left(\sup\left(\hat{X}\right) \geq \text{msup}\right) < t$ according to Definition 5. As for the itemset X, it follows:

$$
\begin{aligned}
w(X)P(\sup(X) \geq \text{msup}) &= w(X) \sum_{\substack{W_i \in W \\ S(X,W_i) \geq \text{msup}}} P(W_i) \\
&\leq w\left(\hat{X}\right) \sum_{\substack{W_i \in W \\ S(X,W_i) \geq \text{msup}}} P(W_i) \\
&\leq w\left(\hat{X}\right) \sum_{\substack{W_i \in W \\ S\left(\hat{X},W_i\right) \geq \text{msup}}} P(W_i) \\
&\leq w\left(\hat{X}\right)P\left(\sup\left(\hat{X}\right) \geq \text{msup}\right) \\
&< t
\end{aligned}
\tag{3}
$$

Here, $S(X, W_i)$ is the support of item X in the possible world W_i. From the above derivation, we obtain a contradiction that itemset X is not a weighted PFI. Thus, there is at least one itemset $X' \subseteq X$, $|X'| = k - 1$ is a weighted PFI.

Based on the anti-monotonicity property in the above Theorem 3, we design our weighted probabilistic frequent itemset mining (wPFI-Apriori) algorithm following the classic Apriori framework, which is widely used in FI mining (Agrawal et al., 1993).

In lines 2 and 7 of Algorithm 1, the main task is to verify whether the itemsets in the candidate set are true weighted PFIs or not. For an itemset X, it is obvious that $w(X)P(\sup(X) \geq \text{msup}) \geq t \Leftrightarrow P(\sup(X) \geq \text{msup}) \geq t/w(X)$. Thus, any mining algorithm and its prune techniques designed for PFI verification can be employed in w-PFI. We adopt the dynamic programming method to compute $P(\sup(X) \geq \text{msup})$, which is widely used in PFI mining. The implementation detail of dynamic programming method can be found in the paper (Bernecker et al., 2009). We will also give a more efficient way to approximately compute $P(\sup(X) \geq \text{msup})$ in next section.

Our concern in this paper is line 6 of Algorithm 1. The main task of line 6 is to generate w-PFI candidates based on the true w-PFI in the former loop. Since the anti-monotonicity property of w-PFI is different from the one of classic PFI, we have to design a novel candidate generation algorithm. According to Theorem 3, a baseline implementation for size-k w-PFI candidate Ck is to combine all the itemsets in WPFI_{k-1} and the items in I except the combinations of duplicate items. However, there are nearly $|\text{WPFI}_{k-1}||I|$ candidates, which usually makes it very time-consuming for the subsequent size-k w-PFI verifying process. The paper (Zhao et al., 2018) which proposed WD property had given a pruning technique called wConnection. We find it sometimes still time-consuming and generates too many candidates. To overcome this problem, we propose a novel candidate pruning method based on a simple corollary of Theorem 3.

Corollary 1. *Given a w-PFI $X \in \text{WPFI}_{k-1}$, the itemset I and the weight table w, an itemset $X' = X \cup I_s$ is not a w-PFI if $w(I_s) \geq \min_{x \in X} w(x)$ and $I_s \in I - I'$, where $I' = \{I_i \mid I_i \in Y, Y \subseteq \text{WPFI}_{k-1}\}$ consists of all the size-1 items of WPFI_{k-1}.*

Algorithm 1

wPFI-Apriori algorithm

```
Require: A uncertain dataset DB, the dataset size n, a weight table w, a minimum support msup,
    a probabilistic frequent threshold t, a scale factor α.
Ensure: A weighted probabilistic frequent itemset WPFI.
 1: Initialize WPFI = ∅, I is the set of all the items in DB.
 2: WPFI_1, μ_1 = Scan_Find_Size_1_wPFI(I, DB, w, msup, t) by Definition 5.
 3: Add WPFI_1 into WPFI.
 4: k = 2.
 5: while WPFI_{k-1} ≠ ∅ do.
 6:     Candidate Ck = wPFI_Apriori_Gen(WPFI_{k-1}, I, μ_{k-1}, w, α, n, msup, t) by Theorem .
 7:     WPFI_k, μ_k = Scan_Find_Size_k_wPFI(Ck, DB, w, msup, t) by Definition 5.
 8:     Add WPFI_k into WPFI.
 9:     k = k + 1.
10: end while
```

Proof. Let the item $I_m = \text{argmin}_{x \in X} w(x)$. The itemset $\hat{X} = (X - I_m) \cup I_s$ is not a size-$(k-1)$ w-PFI since I_s is not a member of I'. Thus, we can obtain $w(\hat{X}) P(\sup(\hat{X}) \geq \text{msup}) < t$, and $w(\hat{X}) \geq w(X')$ according to Definition 6. As to the itemset X', it follows:

$$\begin{aligned} w(X')P(\sup(X') \geq \text{msup}) &= w(X') \sum_{\substack{W_i \in W \\ S(X', W_i) \geq \text{msup}}} P(W_i) \\ &\leq w(\hat{X}) \sum_{\substack{W_i \in W \\ S(X', W_i) \geq \text{msup}}} P(W_i) \\ &\leq w(\hat{X}) \sum_{\substack{W_i \in W \\ S(\hat{X}, W_i) \geq \text{msup}}} P(W_i) \\ &\leq w(\hat{X}) P(\sup(\hat{X}) \geq \text{msup}) \\ &< t \end{aligned} \tag{4}$$

Hence, the itemset $X' = X \cup I_s$ is not a w-PFI.

Based on Corollary 1, we design our first candidate generation and pruning method, summarized in Algorithm 2.

Different from the baseline implementation of Apriori candidate generation, we add the inequality $w(I_i) < w(I_m)$ in line 11 to prune the non-weighted PFI directly based on Corollary 1. It performs well because the weighted PFIs usually only contain a small set of items and thus $|I - I'|$ is rather large. In experiments, we find that our pruning method is more efficient than the method wConnection in both running time and pruning ability. Certainly, our method will still generate many non-weighted PFIs as candidates. To this end, we propose two more pruning approaches based on a probability model in the next section to immediately remove the candidates that are not weighted PFIs of high probabilities.

2.3 | Approximate weighted probabilistic frequent algorithm

Recent studies on PFI try to approximate s-pmf of a PFI by some probability models, that is, Poisson distribution (Wang, Cheng, Lee, & Cheung, 2010) or Normal distribution (Calders et al., 2010a, 2010b). Our motivation is that we may determine the itemset $X \cup I_i$ is a w-PFI or not in Algorithm 2 by the statistical properties of X and I_i. To explain our idea more explicitly, we first introduce a probability model by Theorem 4 to represent the s-pmf, which was first proposed by Wang et al. (2010) and widely adopted in PFI mining. After that, we propose our probability

Algorithm 2

w-PFI candidate generation and pruning

Require: A size-$(k-1)$ w-PFI set WPFI_{k-1}, the itemset I, the weight table w, probabilistic frequent threshold t.
Ensure: A size-k w-PFI candidate set Ck.

```
Initialize Ck = ∅.
I' = {I_i | I_i ∈ Y, Y ⊆ WPFI_{k-1}}
for each itemset X ∈ WPFI_{k-1} do
    for each item I_i ∈ I' − X do
        if w(X ∪ I_i) ≥ t then
            Add X ∪ I_i into Ck
        end if
    end for
    I_m = argmin_{x∈X} w(x)
    for each item I_i ∈ I − I' − X do
        if w(X ∪ I_i) ≥ t And w(I_i) < w(I_m) then
            Add X ∪ I_i into Ck
        end if
    end for
end for
```

model by Theorem 6 for the support of a w-PFI candidate, and give two novel candidate pruning methods based on this model. Finally, we present an efficient w-PFI mining algorithm which can return nearly all the w-PFIs with high confidence.

Given an uncertain database DB, $T \times I \rightarrow [0, 1]$, Each itemset $X \subseteq I$ will be associated with n random variables V_1^X, V_2^X, ..., V_n^X, where $n = |T|$. Here, V_i^X is a Poisson trail with the success probability of $\Pr(X \subseteq T_i)$. Since these random variables have their own probability of success, the random variable $V^X = \sum_{i=1}^{n} V_i^X$ follows a Poisson binomial distribution with the assumption that the transactions in DB are independent to each other. This assumption is widely adopted in the research of frequent item mining.

Theorem 4 *For an itemset X, the support of X denoted by* $\mathrm{Sup}(X)$ *is a random variable, following a Poisson binomial distribution.* $\mathrm{Sup}(X) \sim \mathrm{PBD}(\mu, \sigma^2)$ *where* $\mu = \sum_{i=1}^{n} p_i^X$ *and* $\sigma^2 = \sum_{i=1}^{n} p_i^X (1 - p_i^X)$. $p_i^X = \Pr(X \subseteq T_i)$ *is the probability of the itemset X in the transaction* T_i.

Proof. For an itemset X, its support $Sup(X)$ in the uncertain database DB is actually the sum of $\mathrm{Sup}(X)$ in each transaction of DB. It is easy to see that $\Pr(\mathrm{Sup}(X) = s) = \Pr(V^X = s)$. Thus, $\mathrm{Sup}(X)$ follows the same distribution with V^X. As mentioned, V^X is the sum of n Poisson trails. Following the results of probability theory, we can obtain the mean of V^X is $\mu = \sum_{i=1}^{n} p_i^X$ and the variance is $\sigma^2 = \sum_{i=1}^{n} p_i^X (1 - p_i^X)$.

According to probability theory, a Poisson binomial distribution can be approximated by a Poisson distribution. Given a minimum support msup, the frequentness probability of an itemset X can be computed by $\Pr(\mathrm{Sup}(X) \geq \mathrm{msup}) = 1 - F(\mathrm{msup} - 1, \mu^X)$ where $F(\cdot)$ is the cumulative distribution function (cdf) of Poisson distribution. The efficiency of this model mainly depends on the following important property of frequentness probability $\Pr(\mathrm{Sup}(X) \geq \mathrm{msup})$.

Theorem 5 *For an itemset X and a minimum support* msup, *the frequentness probability* $\Pr(\mathrm{Sup}(X) \geq \mathrm{msup})$ *increases monotonically with* μ^X.

Due to the limited space, we do not prove this theorem here. Proofs can be found in the paper (Wang et al., 2010). The theorem tells that an itemset X is more likely to be a PFI if it has a higher μ^X. When given a probabilistic threshold t, let $1 - F(\mathrm{msup} - 1, \hat{\mu}) = t$. Then any itemset Y with $\mu^Y < \hat{\mu}$ will be probably not a PFI.

Apart from the Poisson distribution, we can also approximate the Poisson binomial distribution $\mathrm{Sup}(X)$, by a Normal distribution (Calders et al., 2010a, 2010b). To achieve it, we should estimate the mean and the variance of $\mathrm{Sup}(X)$ and then compute the probability $\Pr(\mathrm{Sup}(X) \geq \mathrm{msup})$ by the cdf of standard Normal distribution (Calders et al., 2010a, 2010b). After that, we can compare the probability with the minimum probabilistic

frequent threshold t, to determine whether X is a w-PFI candidate or not. However, we find the above process is less effective than our method based on the Poisson distribution. The main reason is that our method has no need for the computation of Pr(Sup(X) ≥ msup) thanks to Theorem 5.

In the following, we discuss w-PFI mining and derive our probability model for the support of a w-PFI candidate. In our candidates generation of Algorithm 2, a size-k w-PFI $X \cup I_i$ is obtained by combining one size-($k-1$) w-PFI X and one size-1 item I_i. Since the support of itemset X and item I_i can both be represented by Poisson binomial distributions, our motivation is that we may immediately determine the itemset $X \cup I_i$ is a w-PFI or not by the statistical properties of X and I_i.

Theorem 6 *For two itemsets X and Y, $\mathrm{Sup}(X) \sim \mathrm{PBD}(\mu_1, \sigma_2^2)$ where $\mu_1 = \sum_{i=1}^{n} p_i^X$ and $\sigma_1^2 = \sum_{i=1}^{n} p_i^X(1-p_i^X)$, $\mathrm{Sup}(Y) \sim \mathrm{PBD}(\mu_2, \sigma_2^2)$ where $\mu_2 = \sum_{i=1}^{n} p_i^Y$ and $\sigma_2^2 = \sum_{i=1}^{n} p_i^Y(1-p_i^Y)$, the support of their union set $\mathrm{Sup}(X \cup Y)$ will also follow a Poisson binomial distribution $\mathrm{Sup}(Y) \sim \mathrm{PBD}(\mu, \sigma^2)$ where $\mu = \sum_{i=1}^{n} p_i^X p_i^Y$ and $\sigma = \sum_{i=1}^{n} p_i^X p_i^Y(1-p_i^X p_i^Y)$, if the two itemsets X and Y satisfy $X \cap Y = \emptyset$.*

Proof. Supposing Sup(X) and Sup(Y) are the sum of n independent Poisson trails $V^X = \sum_{i=1}^{n} V_i^X$ and $V^Y = \sum_{i=1}^{n} V_i^Y$, respectively, then Sup($X \cup Y$) can be considered as also a sum of n independent Poisson trails $V^{X \cup Y} = \sum_{i=1}^{n} V_i^{X \cup Y}$. Each $V_i^{X \cup Y}$ has a success probability of $\Pr(X \cup Y \subseteq T_i)$. Thus, Sup($X \cup Y$) follows a Poisson binomial distribution. Since two itemsets X and Y satisfy $X \cap Y = \emptyset$, it follows that $\Pr(X \cup Y \subseteq T_i) = \Pr(X \subseteq T_i) \times \Pr(Y \subseteq T_i)$ for any transaction $T_i \in T$, that is, $p_i^{X \cup Y} = p_i^X \times p_i^Y$. Thus, we can obtain the mean of Sup($X \cup Y$), $\mu = \sum_{i=1}^{n} p_i^X p_i^Y$ and the variance $\sigma^2 = \sum_{i=1}^{n} p_i^X p_i^Y(1-p_i^X p_i^Y)$.

From the above theorem, it is easy to see that $\mu^{X \cup Y} \le \min\{\mu^X, \mu^Y\}$. With this observation, we can obtain our second pruning method for Algorithm 2 based on the following corollary.

Corollary 2. *Given an itemset X, an item I_s, the weight table w, the minimum support msup and the probabilistic frequent threshold t, the itemset $X' = X \cup I_s$ is not a w-PFI if $\min(\mu^X, \mu^{I_s}) < \hat{\mu}$. Here, $\hat{\mu}$ the solution of the equation $1 - F(\mathrm{msup} - 1, \mu) = t/m$, and $m = \max\{w(X), w(I_s)\}$.*

Proof. Since the itemset $X' = X \cup I_s$, the support Sup(X') follows a Poisson binomial distribution according to Theorem 6. Meanwhile, $\mu^{X'} \le \min\{\mu^X, \mu^Y\}$. Thus, it follows that $\mu^{X'} < \hat{\mu}$. Since $F(\mathrm{msup} - 1, \mu)$ is decreasing monotonically with the increasing of μ, we obtain that $1 - F(\mathrm{msup} - 1, \mu^{X'}) < t/m$, that is, $\Pr(\mathrm{Sup}(X') \ge \mathrm{msup}) < t/m$. As $w(X') < m$ and $m = \max\{w(X), w(I_s)\}$, we finally obtain $w(X')\Pr(\mathrm{Sup}(X') \ge \mathrm{msup}) < t$. Thus, the itemset $X' = X \cup I_s$ is not a w-PFI.

Corollary 2 can be directly added in line 5 and line 11 of Algorithm 2 to remove the non-weighted PFI at once, as a complement of the first pruning method. Notice that it will be more efficient if employing the mean of $X' = X \cup I_s$ directly. However, from Theorem 6, its computation needs to know the success probabilities of X and I_s in each transaction beforehand, which is generally infeasible in practice. Thus, we give an estimation of the mean of $X' = X \cup I_s$ by the mean of X and I_s through the following Corollary 3.

Corollary 3 *Given an itemset X, an item I_s, the weight table w, the minimum support* msup *and the probabilistic frequent threshold t, the itemset $X' = X \cup I_s$ is probably not a w-PFI if $\mu^X \mu^{I_s}/n < \alpha\hat{\mu}$. Here, n is the number of transactions, $\alpha \in (0, 1]$ is a scale factor, $\hat{\mu}$ is the solution of the equation $1 - F(\mathrm{msup} - 1, \mu) = t/m$, and $m = \max\{w(X), w(I_s)\}$.*

Proof. Since Sup(X) and Sup(I_s) both follow the Poisson binomial distribution, we suppose the means of Sup(X) and Sup(I_s) are represented by $\mu^X = \sum_{i=1}^{n} p_i^X$ and $\mu^{I_s} = \sum_{i=1}^{n} p_i^{I_s}$, respectively. As the itemset $X' = X \cup I_s$, Sup(X') follows Poisson binomial distribution too, and the mean $\mu^{X'} = \sum_{i=1}^{n} p_i^X p_i^{I_s}$. Supposing P^X and P^{I_s} are two random variable, $\{p_1^X, p_2^X, \ldots, p_n^X\}$ and $\{p_1^{I_s}, p_2^{I_s}, \ldots, p_n^{I_s}\}$ are their independent samples, respectively. Thus, the mean $E(P^X) = \frac{1}{n}\sum_{i=1}^{n} p_i^X$, $E(P^{I_s}) = \frac{1}{n}\sum_{i=1}^{n} p_i^{I_s}$. Let a random variable $P' = P^X P^{I_s}$, it follows that $E(P') = \frac{1}{n}\sum_{i=1}^{n} p_i^X p_i^{I_s}$. Assuming that P^X and P^{I_s} are independent, we obtain $E(P^X P^{I_s}) = E(P^X) \times E(P^{I_s})$. Thus, $\frac{1}{n}\sum_{i=1}^{n} p_i^X \times \frac{1}{n}\sum_{i=1}^{n} p_i^{I_s} = \frac{1}{n}\sum_{i=1}^{n} p_i^X p_i^{I_s}$. That is, $\mu^X \mu^{I_s}/n = \mu^{X'}$. It is noticeable that the equality is only correct from the viewpoint of probability. Anyway, $\mu^X \mu^{I_s}/n$ is a good approximation of $\mu^{X'}$ in probability. As $\mu^X \mu^{I_s}/n < \alpha\hat{\mu}$ and $0 < \alpha \le 1$, $\mu^{X'}$ will be probably smaller than $\hat{\mu}$. We introduce a scale factor α here to further increase the likelihood of $\mu^{X'} < \hat{\mu}$. Thus, it follows $1 - F(\mathrm{msup} - 1, \mu^{X'}) < t/m$, that is, $\Pr(\mathrm{Sup}(X') \ge \mathrm{msup}) < t/m$. Since $w(X') < m$, we finally obtain $w(X')\Pr(\mathrm{Sup}(X') \ge \mathrm{msup}) < t$. Hence, the itemset $X' = X \cup I_s$ is not a w-PFI.

With the above two corollaries, we give a candidate generation and pruning method in Algorithm 3. Note that we have not used the minimal one of $w(X)$ and $w(I_s)$ but adopted $m = \max\{w_i | w_i \in w\}$ instead, which is usually not a tight bounder but efficient in implementation. It is obvious

Algorithm 3

w-PFI candidate generation and pruning based on a probability model

Require: A size-$(k-1)$ weighted PFI $WPFI_{k-1}$, itemset I, weight table w, mean u, scale α, transaction size n, minimum support msup and probabilistic frequent threshold t.

Ensure: A size-k weighted PFI candidate set Ck

1: Initialize $Ck = \emptyset$
2: $m = \max\{w_i \mid w_i \in w\}$
3: $\hat{\mu} = \{\mu | 1 - F(\text{msup} - 1, \mu) < t/m\}$
4: $I' = \{I_i \mid I_i \in Y,\ Y \subseteq WPFI_{k-1}\}$
5: for each itemset $X \in WPFI_{k-1}$ do
6: for each item $I_i \in I' - X$ do
7: if $w(X \cup I_i) \geq t$ then
8: if $\min\{u^X, u^{I_i}\} \geq \hat{\mu}$ And $\mu^X \mu^{I_i} \geq \alpha n \hat{\mu}$ then
9: Add $X \cup I_i$ into Ck
10: end if
11: end if
12: end for
13: $I_m = \text{argmin}_{x \in X} w(x)$
14: for each item $I_i \in I - I' - X$ do
15: if $w(X \cup I_i) \geq t$ And $w(I_i) < w(I_m)$ then
16: if $\min\{u^X, u^{I_i}\} \geq \hat{\mu}$ And $\mu^X \mu^{I_i} \geq \alpha n \hat{\mu}$ then
17: Add $X \cup I_i$ into Ck
18: end if
19: end if
20: end for
21: end for

that this algorithm will be more efficient than our original method in Algorithm 2. Thus, Algorithm 3 is actually the implementation of the wPFI_Apriori_Gen function in our w-PFI mining algorithm (Algorithm 1).

To sum up, we give some properties of our w-PFI mining Algorithm 1. Firstly, it is designed in a breadth-first search manner with a candidate generate-and-test paradigm like Apriori. Thanks to this paradigm, our algorithm has low overhead and thus is scalable to large dataset. Secondly, the algorithm utilizes the anti-monotonicity property in Theorem 3 and has three pruning techniques, which further narrow the search space and remove the non-weighted PFI at once. The amount of pruned candidates can also be intuitively controlled by the scale factor α which is easy to be tuned in practice. Thirdly, the algorithm can be implemented for both exact w-PFI mining ($\alpha = 0$) and approximate w-PFI mining ($\alpha > 0$). Any candidate verification method designed for PFI mining can be employed in line 2 and line 7 of Algorithm 1. We employ two representative PFI verification methods in experiments to make comparison. One is dynamic programming based exact verification, and the other is Normal distribution-based approximate verification.

3 | EXPERIMENTAL EVALUATION

In this section, we introduce the experimental results and evaluate the performance of our w-PFI mining algorithms. The experiments are carried out on a computer with a Core8 Intel 2.60 GHz CPU and 64 GB of RAM. The code is implemented in MATLAB 2018b with some time-consuming parts written in C by a MEX interface, such as database scanning and dynamic programming based frequency computing.

3.1 | Experimental settings

We conduct the experiments on the following four public datasets including three real-world datasets and one synthetic dataset, which come from the Frequent Itemset Mining (FIMI) repository (Goethals, 2004) and the Open-Source Data Mining Library (SPMF) repository (Fournier-Viger et al., 2014):

- *Connect*: contains 67,557 transactions with 129 distinct items, which are prepared by Roberto Bayardo from UCI Machine Learning Repository.
- *Accidents*: contains 340,183 transactions with 468 distinct items, which are traffic accident records obtained from the National Institute of Statistics.
- USCensus: contains 1,000,000 transactions with 396 distinct items, which are transformed from the US Census 1990 dataset.
- *T40I10D100K*: contains 100,000 transactions with 942 distinct items, which are generated through the IBM synthetic data generator.

The characteristics of above datasets are shown in Table 5, all of which are actually deterministic databases. To generate the uncertainties in these databases, we assign an existential probability to each item in the transactions, as widely done by the existing methods in the current community. The probabilities are drawn from a Gaussian distribution with mean (0.5) and variance (0.125). Meanwhile, the weight table w for the items in each database are drawn from the uniform distribution in (0, 1] as suggested in the papers (Lin et al., 2016; Zhao et al., 2018).

To make comparison, we extend the two anti-monotonicity properties HEWI (Lin et al., 2016) and WD (Zhao et al., 2018) designed for expected-support w-FIs to w-PFI mining as the baseline methods, named p-HEWI and p-WD. For our proposed w-PFI mining Algorithm 1, we have designed three pruning methods in Corollary 1, 2 and 3. Since the first two prune methods can be considered as exact pruning, we combine them together and call this version of Algorithm 1 as w-PFI-EP. Certainly, we also conduct experiments with all the three pruning methods to test the performance of our proposed w-PFI algorithm, named w-PFI-AP, since the third pruning method prunes the search space approximately.

Thus, the four w-PFI mining methods include two baseline methods p-HEWI, p-WD and two our methods w-PFI-EP, w-PFI-AP. As to w-PFI mining, the most time-consuming process is the computation of frequentness probability. There are generally two kinds of solutions to it. One computes the probabilities in an exact way like the methods based on dynamic programming or divide-and-conquer strategy. The other estimates the probabilities through some probability models such as Poisson distribution or Normal distribution. In experiments, we implement the above four methods in both exact frequentness probabilities computation and approximate frequentness probabilities computation, to make a detailed performance comparison. We choose the techniques of dynamic programming (Bernecker et al., 2009) and Normal distribution (Calders et al., 2010a, 2010b) in the implementation, since they are the state-of-art methods in this area.

There are three main parameters in our algorithm: msup, t and α. The parameter msup is the minimum support for w-PFI mining in Definition 5, which is usually set as a percentage of the dataset size n, and is also related to the density of the dataset. The parameter $t \in (0, 1]$ is the probabilistic frequent threshold in Definition 5, which measures the minimal confidence of an itemset to be a w-PFI according to its frequentness probability. The parameter $\alpha \in [0, 1]$ is a scale factor in Algorithm 3, which is used to control the pruning ability of the algorithm. The default value of these parameters in each dataset is listed in Table 5. We will examine the performance of the methods with respect to these parameters in the following section.

3.2 | Comparisons on exact w-PFI mining

In this section, the frequentness probability is computed by dynamic programming and thus the above four methods discover the w-PFIs exactly. The time complexity for the dynamic programming computation is $O(m \times n \times \text{msup})$, where m is the number of candidate itemsets, n is the size of database. It usually costs too much time in certain situations. Thus, we use the first 10K transactions of the above four databases in practice to make the runtime more reasonable, as done by the model-based PFI mining (Wang et al., 2010). The mining over the whole database is considered in the next section.

Moreover, in existing PFI mining algorithms, Chernoff bound-based filtering with a time complexity of $O(m \times n)$ is widely employed at the beginning of the candidate verifying process to accelerate the subsequent dynamic programming step (Sun et al., 2010). In our experiments, we test the performance of the four methods in the both cases of adopting the Chernoff bound-based filtering or not.

TABLE 5 Characteristics and default parameters of datasets

Dataset	No. of transactions	No. of items	Average length	Density	msup	t	α
Connect	67,557	129	43	0.33	0.2n	0.6	0.6
Accidents	340,183	468	33.8	0.072	0.1n	0.6	0.6
USCensus	1,000,000	396	48	0.12	0.1n	0.6	0.5
T40I10D100K	100,000	942	39.6	0.042	0.01n	0.6	0.5

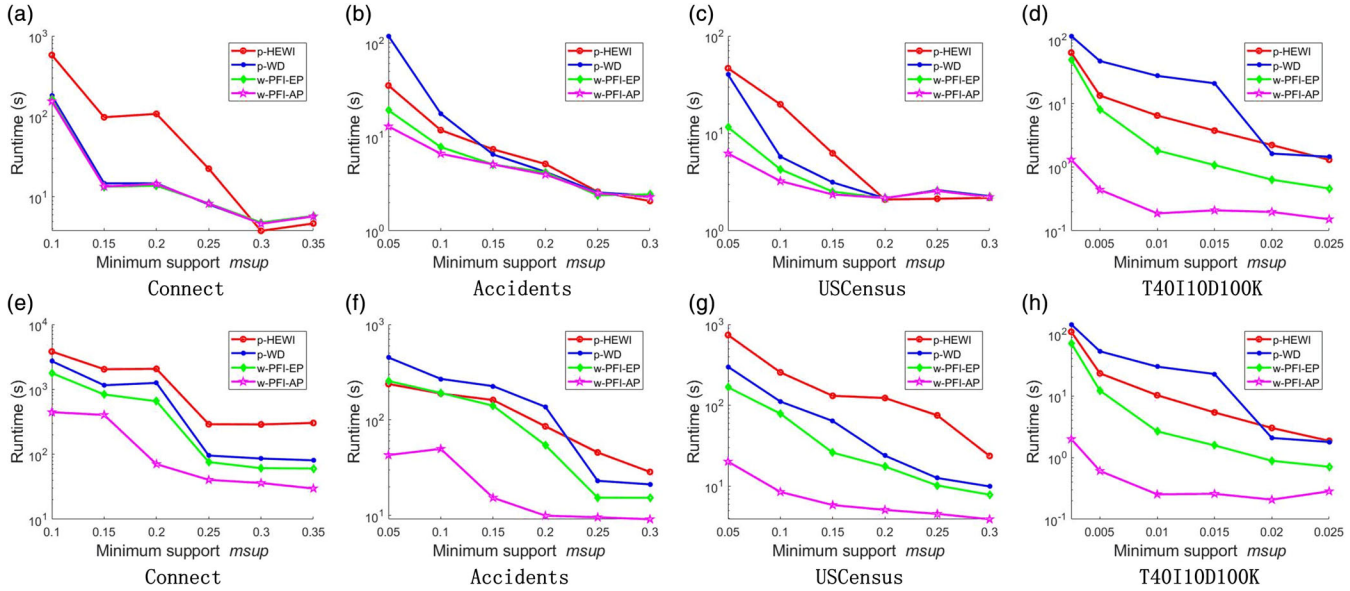


FIGURE 2 Runtime versus msup over the four datasets with Chernoff bound-based filtering (first row) or not (second row)

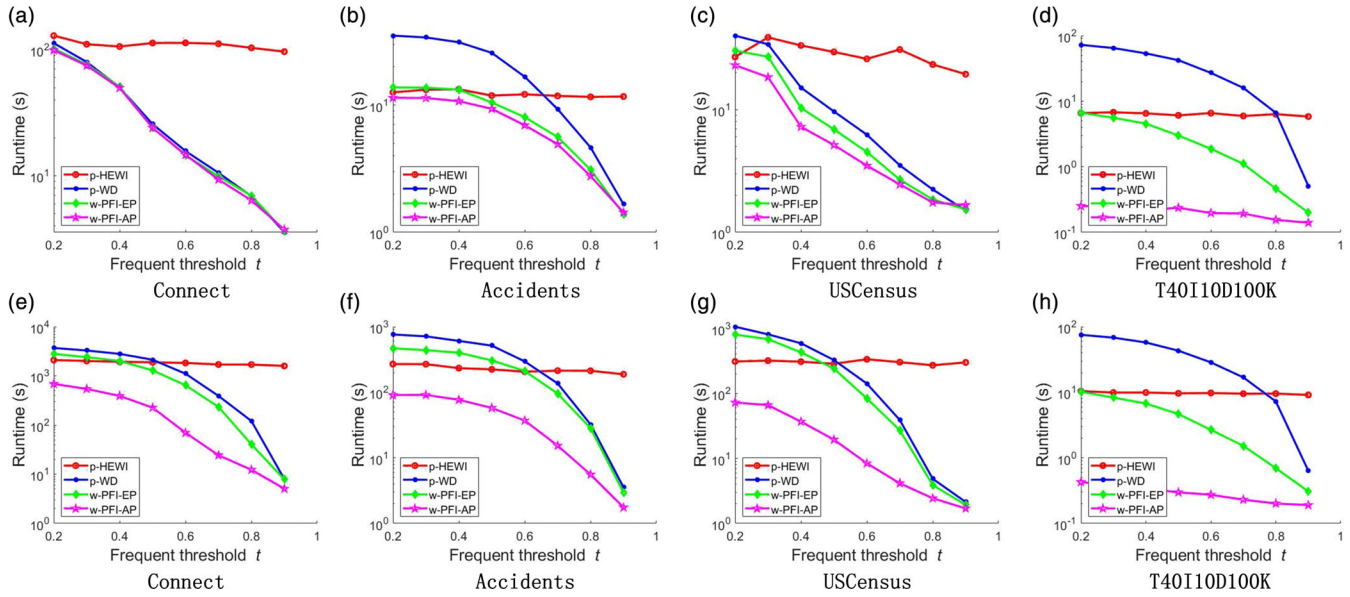


FIGURE 3 Runtime versus *t* over the four datasets with Chernoff bound-based filtering (first row) or not (second row)

3.2.1 | Effect of msup

Generally, the running time will increase with the decreasing of msup, because more w-PFIs will be discovered and thus more w-PFI candidates should be verified. Figure 2 shows the running time of the four methods with respect to the minimum support msup. In the first row of Figure 2, we implement the four methods with the Chernoff bound-based filtering. We can see that our methods w-PFI-EP and w-PFI-AP run faster than the other two baseline methods p-HEWI and p-WD. The probabilistic frequent threshold *t* is set as the default value 0.6.

In the second row of Figure 2, we implement the four methods without the Chernoff bound-based filtering. Our method w-PFI-AP is still the best one among the four methods. It is obvious that all the methods cost much more running time compared to the first row. Because without Chernoff bound-based filtering, more w-PFI candidates have to be verified by the mentioned dynamic programming process, which is very time-consuming. Fortunately, the performance of our method w-PFI-AP without the Chernoff bound-based filtering has not decreased so much, compared to other three methods. The reason is that the candidates of w-PFI-AP have been efficiently pruned by the proposed probability model

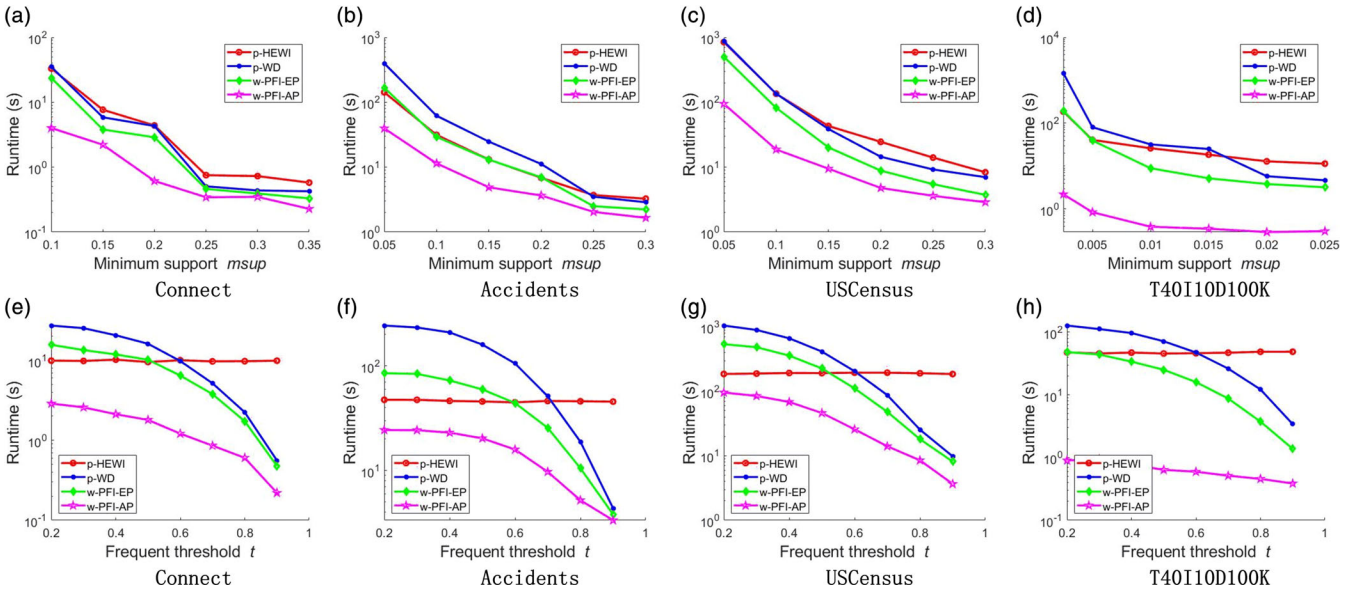


FIGURE 4 Runtime versus msup and t over the four datasets

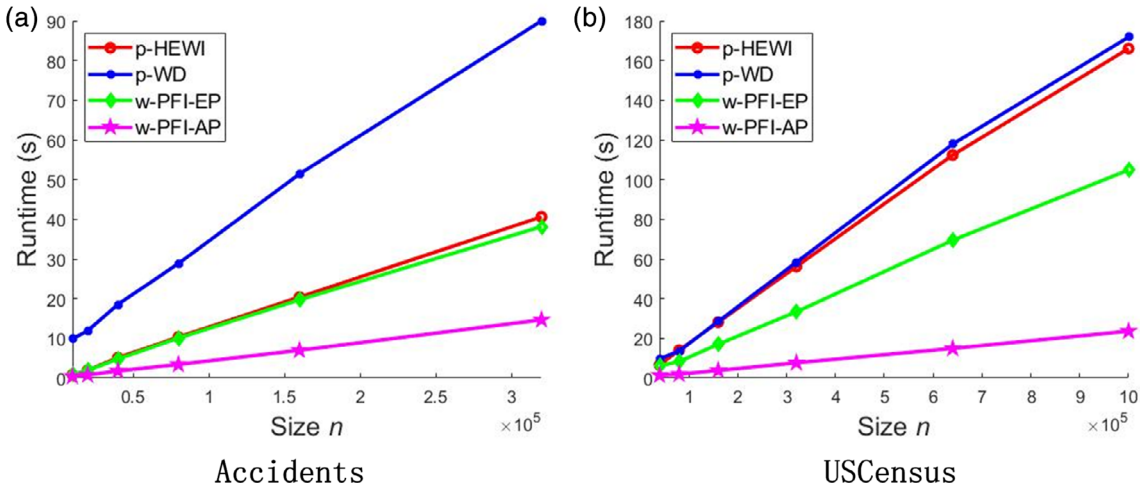


FIGURE 5 Runtime versus size n over the datasets accidents and USCensus

based on Corollary 3. Compared to the Chernoff bound-based filtering, our pruning technique has a main advantage that it is implemented in the candidate generation process, removing the non-weighted PFIs at once. However, the Chernoff bound-based filtering needs a scanning of whole database beforehand and is usually used in the later candidate verifying process.

From Figure 2d,h, it is interesting that the process of Chernoff bound-based filtering has little influence on the running time in the T40I10D100K dataset. The reason is that we have adopted the hash tree technique in the database scanning. As for the T40I10D100K dataset, there are only tens of transactions indexed for most itemsets, and thus the dynamic programming computation is greatly accelerated by hash trees. The Chernoff bound-based filtering has not favoured the T40I10D100K dataset much yet, compared to other three datasets.

3.2.2 | Effect of t

Figure 3 shows the runtime of the four methods with respect to the probabilistic frequent threshold t. To make fair comparison, the parameter msup and α are set to their default values. The parameter t indicates the minimal confidence of an itemset to be an w-PFI by its frequentness probability. In experiments, we find that the frequentness probabilities of most PFIs are nearly 1, which is consistent with the results in the paper (Tong et al., 2012). Thus, t is expected to have little impact on the running time of the four methods. However, the running time of p-WD and our

TABLE 6 Accuracies in the datasets of Accidents (left) and T40I10D100K (right)

msup/*n*	**0.05**	**0.1**	**0.15**	**0.2**	**0.25**	**0.3**
w-PFI-EP	1	1	1	1	1	1
w-PFI-AP	1	1	1	1	1	1
t	**0.2**	**0.4**	**0.5**	**0.6**	**0.8**	**0.9**
w-PFI-EP	1	1	1	1	1	1
w-PFI-AP	0.99	1	1	1	1	1
n	**1**	**5*k***	**10*k***	**20*k***	**50*k***	**100*k***
w-PFI-EP	1	1	1	1	1	1
w-PFI-AP	1	1	1	1	1	1
α	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
w-PFI-AP	1	1	1	1	1	1
msup/*n*	**0.0025**	**0.005**	**0.01**	**0.015**	**0.02**	**0.025**
w-PFI-EP	1	1	1	1	1	1
w-PFI-AP	0.91	0.97	1	1	1	1
t	**0.2**	**0.4**	**0.5**	**0.6**	**0.8**	**0.9**
w-PFI-EP	1	1	1	1	1	1
w-PFI-AP	1	1	1	1	1	1
n	**1*k***	**5*k***	**10*k***	**20*k***	**50*k***	**100*k***
w-PFI-EP	1	1	1	1	1	1
w-PFI-AP	1	1	1	1	1	1
α	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
w-PFI-AP	1	1	1	1	0.99	0.99

methods actually decreases with the increasing of t. The reason is that PFIs with weights smaller than t will be identified as non-weighted PFIs directly without the time-consuming w-PFI verifying process.

Thus, fewer w-PFIs will be obtained under higher values of t in practice. From Figure 3, we can see that our method w-PFI-AP consistently has the lowest running time in terms of varying t, no matter adopting the Chernoff bound-based filtering or not. Compared to other three methods, the method p-HEWI performs well when t is small, but its performance decreases relatively when t is large. To maintain the anti-monotonicity property of w-PFI, p-HEWI has to discover an additional PFI set under a new probabilistic frequent threshold $t' = t/\max(w)$, which is actually a superset of w-PFI. Hence, the parameter t almost has no impact on the running time of p-HEWI.

3.2.3 | Accuracy

Our pruning methods mainly depend on the availability of the probability models in Corollary 2 and 3. It is necessary to examine their accuracy with respect to the ground truth, since the methods may remove true w-PFIs from the candidate set. In experiments, we take the w-PFIs discovered by p-WD as the ground truth, which yields them based on exact pruning. We adopt the standard recall rate to measure the completeness of the mining results.

$$\text{recall} = \frac{|F_w \cap F_o|}{|F_w|}, \tag{5}$$

where F_w and F_o are the w-PFIs obtained by p-WD and our methods, respectively.

Table 6 lists the recall rate of our method w-PFI-EP, w-PFI-AP in the dataset Accident and T40I10D100K. The results in other datasets are similar. As we can see, the recall rate of w-PFI-EP remains 1 with respect to the parameters msup, t and n, although it is based on the probability model in Corollary 2. It can be actually regarded as an exact pruning method. Meanwhile, the recall rates of w-PFI-AP are always higher than 90%. Thus, w-PFI-AP can be adopted in practice since it efficiently prune the candidate set with almost all the w-PFIs left, especially when the support

TABLE 7 Accuracies in the datasets of Accidents (left) and T40I10D100K (right)

	0.05		0.1		0.15	
msup	**R**	**P**	**R**	**P**	**R**	**P**
p-HEWI	1	1	1	1	1	1
p-WD	1	1	1	1	1	1
w-PFI-EP	1	1	1	1	1	1
w-PFI-AP	1	1	1	1	1	1
	0.2		**0.4**		**0.6**	
t	**R**	**P**	**R**	**P**	**R**	**P**
p-HEWI	1	1	1	1	1	1
p-WD	1	1	1	1	1	1
w-PFI-EP	1	1	1	1	1	1
w-PFI-AP	1	1	1	1	1	1
	0.5		**0.7**		**0.9**	
α	**R**	**P**	**R**	**P**	**R**	**P**
w-PFI-AP	1	1	1	1	1	1
	0.0025		**0.005**		**0.01**	
msup	**R**	**P**	**R**	**P**	**R**	**P**
p-HEWI	0.99	0.99	1	1	1	1
p-WD	0.99	0.99	1	1	1	1
w-PFI-EP	0.99	0.99	1	1	1	1
w-PFI-AP	0.91	0.99	0.96	1	1	1
	0.2		**0.4**		**0.6**	
t	**R**	**P**	**R**	**P**	**R**	**P**
p-HEWI	1	1	1	1	1	1
p-WD	1	1	1	1	1	1
w-PFI-EP	1	1	1	1	1	1
w-PFI-AP	1	1	1	1	1	1
	0.5		**0.7**		**0.9**	
α	**R**	**P**	**R**	**P**	**R**	**P**
w-PFI-AP	1	1	0.99	1	0.99	1

of the itemset is large. It is a little surprising that w-PFI-AP can still discover all the w-PFIs when the size of the database is small, for example, $n = 1k$. While investigating the impact of one parameter by experiments, we set the other parameters to their default values listed in Table 5.

3.3 | Comparisons on approximate w-PFI mining

In this section, the frequentness probability is estimated by Normal distribution and thus the above four methods mine the w-PFIs approximately. The time complexity for this model-based frequentness probability computation is $O(m \times n)$, which is very efficiently and is scalable to large database. Hence, we can now mine w-PFIs over all the transactions of the four datasets. Certainly, an efficient pruning technique can reduce the number of w-PFI candidates m, which will further decrease the running time.

Figure 4 show the running time of the four methods with respect to minimum support msup and probabilistic frequent threshold t, respectively. The other parameters are set to their default values. Similar conclusions can be drawn as those from exact w-PFI mining. Our method w-PFI-AP consistently has the lowest running time among the four methods. The reason is that w-PFI-AP generates the smallest number of candidates thanks to the efficient pruning techniques.

In Figure 5, we examine the scalability of the four methods. We choose two larger datasets Accident and USCensus to conduct experiments and set the other parameters as the default value. As we can see, all the methods scale well with the size of dataset n due to the efficient approximate model based on Normal distribution. Our methods w-PFI-EP and w-PFI-AP both perform well, and w-PFI-AP outperforms the competitors and achieves the lowest running time.

In Table 7, we give both the recall (R) and the precision (P) of the four methods in the datasets Accident and T40I10D100K with respect to msup, t and α. The precision rate is defined as the percentage of true w-PFIs in the discovered w-PFIs. The ground truth of w-PFIs in the two datasets is obtained beforehand by an exact w-PFI mining over these datasets through a mentioned dynamic programming process. From Table 7, the recall and precision rates of our method w-PFI-EP are the same with P-HEWI and p-WD, and our method w-PFI-AP has some false negatives but the recall rates are still higher than 90%. Certainly, the recall rate can be further increased through decreasing the scale factor α.

4 | CONCLUSIONS

To discover w-PFIs from uncertain databases, we propose a novel mining algorithm which is designed by an efficient candidate generate and test paradigm like Apriori. Moreover, we develop a new probability model for the support of a w-PFI candidate and present three pruning techniques to remove the unpromising candidates immediately. The amount of pruned candidates can be intuitively controlled by a parameter thanks to the probability model. We conduct extensive experiments on both real and synthetic datasets and compare our w-PFI mining algorithm with the existing algorithms. The results show that our algorithm is highly accurate and yields the best performance in terms of running time and scalability when used together with the three pruning techniques.

ACKNOWLEDGEMENTS

This work is supported by the National Nature Science Foundation of China (Nos 61672379, 61370198, 61300187 and 61370199), Liaoning Provincial Nature Science Foundation of China No. 2019-MS-028, and China Postdoctoral Science Foundation (No. 2018M640239).

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

ORCID

Zhiyang Li https://orcid.org/0000-0002-5396-3447

REFERENCES

Aggarwal, C. C., Li, Y., Wang, J., & Wang, J. (2009). Frequent pattern mining with uncertain data. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 29–38).

Aggarwal, C. C., & Yu, P. S. (2009). A survey of uncertain data algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering*, *21*(5), 609–623.

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data* (pp. 207–216).

Bernecker, T., Kriegel, H.-P., Renz, M., Verhein, F., & Zuefle, A. (2009). Probabilistic frequent itemset mining in uncertain databases. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 119–128).

Calders, T., Garboni, C., & Goethals, B. (2010a). Approximation of frequentness probability of itemsets in uncertain data. *2010 IEEE International Conference on Data Mining* (pp. 749–754).

Calders, T., Garboni, C., & Goethals, B. (2010b). Efficient pattern mining of uncertain data with sampling. *Proceedings of the 14th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining* (Vol. Part I, pp. 480–487).

Chui, C.-K., Kao, B., & Hung, E. (2007). Mining frequent itemsets from uncertain data. *Proceedings of the 11th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining* (pp. 47–58).

Fournier Viger, P., Lin, C.-W., Vo, B., Truong, T., Zhang, J., & Le, B. (2017). A survey of itemset mining. *WIREs Data Mining and Knowledge Discovery*, *7*, e1207.

Fournier-Viger, P., Gomariz, A., Gueniche, T., Soltani, A., Wu, C.-W., & Tseng, V. S. (2014). SPMF: A java open-source pattern mining library. *Journal of Machine Learning Research*, *15*(1), 3389–3393.

Goethals, B. (2004). Frequent itemset mining implementations repository. Retrieved from fimi.ua.ac.be/src/.

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (pp. 1–12).

Lee, G., Yun, U., & Ryang, H. (2015). An uncertainty-based approach: Frequent itemset mining from uncertain data with different item importance. *Knowledge-Based Systems*, *90*, 239–256.

Leung, C. K.-S., & MacKinnon, R. K. (2014). Blimp: A compact tree structure for uncertain frequent pattern mining. In L. Bellatreche & M. K. Mohania (Eds.), *DaWak* (pp. 115–123). Cham, Switzerland: Springer.

Li, H., Zhang, Y., & Zhang, N. (2017). Discovering top-k probabilistic frequent itemsets from uncertain databases. *Procedia Computer Science*, *122*, 1124–1132.

Lin, C. W., Gan, W., Fournier-Viger, P., Hong, T. P., & Tseng, V. S. (2016). Weighted frequent itemset mining over uncertain databases. *Applied Intelligence*, *44*(1), 232–250.

Liu, C., Chen, L., & Zhang, C. (2013). Summarizing probabilistic frequent patterns: A fast approach. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 527–535).

Peterson, E. A., & Tang, P. (2013). Mining probabilistic generalized frequent itemsets in uncertain databases. *Proceedings of the 51st ACM Southeast Conference* (pp. 1:1–1:6).

Sun, L., Cheng, R., Cheung, D. W., & Cheng, J. (2010). Mining uncertain data with probabilistic guarantees. *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 273–282).

Tao, F., Murtagh, F., & Farid, M. (2003). Weighted association rule mining using weighted support and significance framework. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 661–666).

Tong, Y., Chen, L., Cheng, Y., & Yu, P. S. (2012). Mining frequent itemsets over uncertain databases. *Proceedings of the VLDB Endowment*, *5*(11), 1650–1661.

Vo, B., Coenen, F., & Le, B. (2013). A new method for mining frequent weighted itemsets based on wit-trees. *Expert Systems with Applications*, *40*(4), 1256–1264.

Wang, L., Cheng, R., Lee, S. D., & Cheung, D. (2010). Accelerating probabilistic frequent itemset mining: A model-based approach. *Proceedings of the 19th ACM International Conference on Information and Knowledge Management* (pp. 429–438).

Yang, L., Chao, F., & Shen, Q. (2017). Generalized adaptive fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, *25*(4), 839–853.

Yang, L., Yang, Y., Mgaya, G., Zhang, B., Chen, L., & Liu, H. (2020). Novel fast networking approaches mining underlying structures from investment big data. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 1–11. https://doi.org/10.1109/TSMC.2019.2961378

Yun, U. (2008). A new framework for detecting weighted sequential patterns in large sequence databases. *Knowledge-Based Systems*, *21*(2), 110–122.

Zhao, X., Zhang, X., Pan, W., Chen, S., & Sun, Z. (2018). A weighted frequent itemset mining algorithm for intelligent decision in smart systems. *IEEE Access*, *6*, 29271–29282.

AUTHOR BIOGRAPHIES

Zhiyang Li is currently an Associate Professor at the Information Science and Technology College, Dalian Maritime University, China. He received the PhD degree in computation mathematics from Dalian University of Technology, China, in 2011. His research interests include computer vision, cloud computing and data mining. He has published more than 50 papers in international journals and conferences.

Fengjuan Chen is a PhD candidate at the Information Science and Technology College, Dalian Maritime University, China. Her research interests include data mining and machine learning.

Junfeng Wu was born in Dalian, China, in 1983. He is currently a Lecturer with the School of Information Engineering in Dalian Ocean University, China. He received PhD degree from Dalian Maritime University in 2017. His research interests include digital multimedia, data mining and computer vision. He has published over 20 technical papers in international journals and conferences.

Zhaobin Liu received the PhD degree in Computer Science from Huazhong University of Science and Technology, China, in 2004. He is currently a Professor in the School of Information Science and Technology, Dalian Maritime University, China. He has been a Senior Visiting Scientist at The University of Auckland, New Zealand, in 2017, and a visiting scholar at University of Central Florida, USA, in 2014 and University of Otago, New Zealand, in 2008 respectively. His research interests include big data, cloud computing and data privacy.

Weijiang Liu received the PhD degree in Computation Mathematics from Jilin University, Changchun, China, in 1998. He is currently a Professor in the School of Information Technology, Dalian Maritime University, China. He has published more than 50 papers and his current research interests include network measurement, network performance and network security.

How to cite this article: Li Z, Chen F, Wu J, Liu Z, Liu W. Efficient weighted probabilistic frequent itemset mining in uncertain databases. *Expert Systems*. 2020;e12551. https://doi.org/10.1111/exsy.12551