


Computing Theoretically-Sound Upper Bounds to Expected Support for Frequent Pattern Mining Problems over Uncertain Big Data

Alfredo Cuzzocrea¹ and Carson K. Leung²

¹ DIA Department, University of Trieste and ICAR-CNR, Trieste, TS, Italy
alfredo.cuzzocrea@dia.units.it

² Department of Computer Science, University of Manitoba, Winnipeg, MB, Canada
kleung@cs.umanitoba.ca

Abstract. *Frequent pattern mining* aims to discover implicit, previously unknown, and potentially useful knowledge in the form of sets of frequently co-occurring items, events, or objects. To mine frequent patterns from *probabilistic datasets of uncertain data*, where each item in a transaction is usually associated with an existential probability expressing the likelihood of its presence in that transaction, the UF-growth algorithm captures important information about uncertain data in a UF-tree structure so that *expected support* can be computed for each pattern. A pattern is considered frequent if its expected support meets or exceeds the user-specified threshold. However, a challenge is that the UF-tree can be large. To handle this challenge, several algorithms use smaller trees such that *upper bounds to expected support* can be computed. In this paper, we examine these upper bounds, and determine which ones provide tighter upper bounds to expected support for frequent pattern mining of uncertain big data.

Keywords: Uncertainty · Data analysis · Big data · Data science · Data mining

1 Introduction

Uncertain big data (e.g., [21, 33, 34, 38]) are becoming more and more popular in modern applications [23] (e.g., social computing [20, 22], data warehousing and OLAP [10]) because (big) data in real-life scenarios are typically *imprecise and uncertain* (e.g., [14, 17, 19]). *Mining uncertain big data* (e.g., [6, 40]) is problematic due to the fact that models, techniques, and algorithms running on such data must consider uncertainty as a fundamental characteristic of big data while this challenging property is not foreseen by classical large-scale data mining approaches. As a consequence, mining uncertain big data is a first-class problem to deal with, and several interesting initiatives that focus the attention on this problem are appearing recently in active literature [12, 35, 45].

Among the wide class of data mining tasks [4, 16, 42, 43], *frequent pattern mining* [2] is a very popular problem that has attracted the attention of a large

community of data miners. Frequent pattern mining aims to discover implicit, previously unknown, and potentially useful knowledge in the form of sets of frequently co-occurring items, events, or objects (i.e., frequent patterns). It also serves as building blocks for various other data mining tasks such as stream mining [8, 9, 25, 26] (which mines data that come at a high velocity), constrained mining [13], and social network mining [27, 41]. Many existing algorithms mine frequent patterns from high volumes of precise data, in which users definitely know whether an item is present in, or absent from, a transaction in databases of precise data. However, there are situations in which users are uncertain about the presence or absence of items (e.g., a physician may suspect, but may not guarantee, that a fevered patient got a flu or Zika virus) in a *probabilistic dataset of uncertain data*. In such dataset, each item x_i in a transaction t_j is associated with an *existential probability* $P(x_i, t_j)$ expressing the likelihood of the presence of x_i in t_j .

To mine frequent patterns from high varieties of (high-value) uncertain data, various algorithms [1, 3, 44] have been proposed, including UF-growth [30]. The UF-growth algorithm first constructs a UF-tree structure with the goal of capturing important contents on uncertain data, from which frequent patterns can then be mined recursively. A pattern X is considered *frequent* if its expected support $expSup(X)$ in the entire uncertain dataset meets or exceeds the user-specified minimum support threshold $minsup$ [24]. Here, $expSup(X)$ over all n transactions in the uncertain dataset can be computed in terms of the sum of $expSup(X, t_j)$ over every transaction t_j containing X , as follows:

$$expSup(X) = \sum_{j=1}^n expSup(X, t_j) \quad (1)$$

while $expSup(X, t_j)$ can be computed in terms of the product of the existential probability $P(x_i, t_j)$ of every independent item x_i within the pattern $X = \{x_1, \dots, x_k\}$, as follows:

$$expSup(X, t_j) = \prod_{i=1}^k P(x_i, t_j) \quad (2)$$

In order to accurately compute the expected support of each pattern, paths in the corresponding UF-tree are shared only if tree nodes on the paths have the same item and the same existential probability. Due to this restrictive path sharing requirement, the UF-tree may be quite large.

A way to solve this large tree-size issue is to explore alternative mining approaches (e.g., UH-Mine algorithm [1] that uses hyper-structures, as well as sampling-based or vertical mining approaches [5]). Another way is to make the tree compact by capturing less information about uncertain data but sufficient for computing *upper bounds to the expected support* of patterns. Over the past few years, different computations on the upper bounds to expected support have been proposed. Many of them are reported to lead to more compact tree structures for capturing uncertain data than the UF-tree. These, in turn, shorten the

tree traversal time during the mining process, and thus help reduce the overall runtime. In addition, another benefit of using these upper bounds is that they are guaranteed *not* to generate any false negatives. Indeed, if an upper bound to expected support of a pattern X is less than *minsup*, then X is guaranteed to be infrequent. Moreover, these upper bounds are reported to be so tight that not too many false positives are generated-and-tested. Then, interesting questions to ask include the following: Among these upper bounds, which one is tighter? Which one leads to shorter runtime or mining time? In this paper, we examine these upper bounds, and re-formulate them so that we can compare them and determine which ones provide tighter upper bounds to expected support of patterns when mining frequent patterns from a high variety of high volumes of high-value uncertain data that may come at a high velocity (i.e., uncertain “streaming” big data). Our *key contributions* of this paper include our computation of theoretically-sound upper bounds to expected support for frequent pattern mining problems over uncertain big data.

The remainder of this paper is organized as follows. In Sect. 2, we provide a formal unifying model for computing upper bounds to expected support, as to obtain a (formal) model to be used throughout the paper. The section also contains relevant related work for our research. Section 3 reports a theoretical analysis on the bounds. In Sect. 4, we provide an experimental assessment and evaluation of our methods for computing these upper bounds, according to several experimental parameters. Finally, Sect. 5 presents conclusions and proposes future work of our research.

2 Computing Upper Bounds: A Unifying Model from the State-of-the-Art Analysis

In this section, we re-formulate upper bounds to expected support (as provided by the state-of-the-art analysis) via using a common expression or notion so that we can introduce a unifying model for easily comparing among the various proposals available in literature. This section also serves as analysis of related work that is relevant to our research.

To approximate an upper bound to expected support of a pattern X , CUF-growth [31] introduces the concept of *transaction cap* (TC), which is defined as the product of the two highest existential probabilities in the entire transaction $t_j = \{y_1, \dots, y_{r-1}, y_r, \dots, y_h\} \supseteq \{x_1, \dots, x_k\} = X$ (where $x_k = y_r$), as follows:

$$TC(X, t_j) = \begin{cases} P(y_1, t_j) & \text{if } h = 1 \\ TM_1(t_j) \times TM_2(t_j) & \text{if } h \geq 2 \end{cases} \quad (3)$$

where (i) $TM_1(t_j) = \max_{i \in [1, h]} P(y_i, t_j)$ is the *transaction maximum*, which is defined as the highest existential probability in t_j ; and (ii) $TM_2(t_j) = \max_{i \in [1, h] \wedge (i \neq g)} P(y_i, t_j)$ is the second highest existential probability in t_j where $y_g = \operatorname{argmax}_{i \in [1, h]} P(y_i, t_j)$ (i.e., $TM_1(t_j) = P(y_g, t_j)$).

While this transaction cap serves as a good upper bound to 2-itemsets, it may not be too tight for k -itemsets (where $k \geq 3$). To tighten the upper bound

to expected support for 3^+ -itemsets, *CUF*-growth* [31] extends the concept of transaction cap as to use the product of the three highest existential probabilities in t_j , as follows:

$$CUF^*(X, t_j) = \begin{cases} TC(X, t_j) & \text{if } k \leq 2 \\ TM_1(t_j) \times TM_2(t_j) \times [TM_3(t_j)]^{k-2} & \text{if } k \geq 3 \end{cases} \quad (4)$$

where $TM_3(t_j) = \max_{i \in [1, h] \wedge (i \neq g) \wedge (i \neq s)} P(y_i, t_j)$ is the third highest existential probability in t_j for $y_s = \operatorname{argmax}_{i \in [1, h] \wedge (i \neq g)} P(y_i, t_j)$ (i.e., $TM_2(t_j) = P(y_s, t_j)$).

On the one hand, the transaction cap can be easily pre-computed. On the other hand, it may not involve any items in X . To tighten the upper bound, *item cap (IC)* [37] involves at least one item in X . Specifically, the item cap is defined as the product of $P(x_k, t_j)$ and the highest existential probability $TM_1(t_j)$ in t_j , as follows:

$$IC(X, t_j) = \begin{cases} P(y_1, t_j) & \text{if } h = 1 \\ P(x_k, t_j) \times TM_1(t_j) & \text{if } h \geq 2 \end{cases} \quad (5)$$

For the special case where $TM_1(t_j) = P(x_k, t_j)$, *DISC-growth* [37] avoids multiplying $TM_1(t_j)$ twice. Instead, it multiplies $P(x_k, t_j)$ by the second highest existential probability $TM_2(t_j)$ in t_j , as follows:

$$DISC(X, t_j) = \begin{cases} P(y_1, t_j) & \text{if } h = 1 \\ P(x_k, t_j) \times TM_1(t_j) & \text{if } h \geq 2 \wedge x_k \neq y_g \\ P(x_k, t_j) \times TM_2(t_j) & \text{if } h \geq 2 \wedge x_k = y_g \end{cases} \quad (6)$$

To deal with 3^+ -itemsets, *DISC*-growth* [37] uses the self-product of $TM_2(t_j)$. For special cases where (i) $TM_1(t_j) = P(x_k, t_j)$ or (ii) $TM_2(t_j) = P(x_k, t_j)$, *DISC*-growth* uses the self-product of the third highest existential probability $TM_3(t_j)$ in t_j , as follows:

$$DISC^*(X, t_j) = \begin{cases} DISC(X, t_j) & \text{if } k \leq 2 \\ P(x_k, t_j) \times TM_1(t_j) \times [TM_2(t_j)]^{k-2} & \text{if } k \geq 3 \wedge x_k \neq y_g \wedge x_k \neq y_s \\ P(x_k, t_j) \times TM_1(t_j) \times [TM_3(t_j)]^{k-2} & \text{if } k \geq 3 \wedge x_k = y_s \\ P(x_k, t_j) \times TM_2(t_j) \times [TM_3(t_j)]^{k-2} & \text{if } k \geq 3 \wedge x_k = y_g \end{cases} \quad (7)$$

Recall from Eq. (2) that the expected support of X can be computed as the product of $P(x_k, t_j)$ and existential probabilities of proper prefix of x_k . Hence, it is more logical to approximate an upper bound to expected support of X by involving $P(x_k, t_j)$ and existential probabilities of proper prefix of x_k . This leads to the concept of *prefixed item cap (PIC)* [29], which is defined as the product of $P(x_k, t_j)$ and the highest existential probability $PM_1(y_r, t_j)$ among items in the proper prefix of $x_k=y_r$, as follows:

$$PIC(X, t_j) = \begin{cases} P(y_1, t_j) & \text{if } h = 1 \\ P(x_k, t_j) \times PM_1(y_r, t_j) & \text{if } h \geq 2 \end{cases} \quad (8)$$

where (i) $PM_1(y_r, t_j) = \max_{i \in [1, r-1]} P(y_i, t_j)$ is the *prefixed maximum*, which is defined as the highest existential probability in $\{y_1, \dots, y_{r-1}\} \subset t_j$.

PUF-growth [32] makes use of the above prefixed item cap to approximate a tight upper bound to expected support of 2-itemsets. To handle 3^+ -itemsets, *PUF*-growth* [36] multiplies $PIC(X, t_j)$ with self-product of the second highest existential probability $PM_2(y_r, t_j)$ in $\{y_1, \dots, y_{r-1}\} \subset t_j$, as follows:

$$PUF^*(X, t_j) = \begin{cases} PIC(X, t_j) & \text{if } k \leq 2 \\ P(x_k, t_j) \times PM_1(y_r, t_j) \times [PM_2(y_r, t_j)]^{k-2} & \text{if } k \geq 3 \end{cases} \quad (9)$$

where $PM_2(y_r, t_j) = \max_{i \in [1, r-1] \wedge (i \neq g)} P(y_i, t_j)$ is the second highest existential probability in $\{y_1, \dots, y_{r-1}\} \subset t_j$ for $y_g = \operatorname{argmax}_{i \in [1, h]} P(y_i, t_j)$ (i.e., $PM_1(y_r, t_j) = P(y_g, t_j)$).

Alternatively, the *BLIMP-growth* algorithm [28] multiplies $PIC(X, t_j)$ with existential probabilities of the first $(k-2)$ items in the proper prefix $\{y_1, \dots, y_{r-1}\} \subset t_j$, as follows:

$$BLIMP(X, t_j) = \begin{cases} PIC(X, t_j) & \text{if } k \leq 2 \\ P(x_k, t_j) \times PM_1(y_r, t_j) \times \prod_{i=1}^{k-2} P(y_i, t_j) & \text{if } k \geq 3 \end{cases} \quad (10)$$

3 Theoretical Analysis and Results

After re-formulating upper bounds to expected support of patterns in Sect. 2, let us analyze and evaluate these bounds by taking advantages from the unifying model introduced above. When dealing with singletons (1-itemsets), we do not need to use upper bounds because we could scan the entire uncertain dataset of n transactions and accurately obtain the expected support of each pattern $\{x\}$ by summing existential probabilities of $\{x\}$ in every transaction t_j containing $\{x\}$:

$$\operatorname{expSup}(\{x\}) = \sum_{j=1}^n P(x, t_j) \quad (11)$$

For any 2-itemset X , the upper bound computing models of Sect. 2 specialize as follows:

$$CUF^*(X, t_j) = TC(X, t_j) \quad (12)$$

$$DISC^*(X, t_j) = DISC(X, t_j) \quad (13)$$

$$PUF^*(X, t_j) = BLIMP(X, t_j) = PIC(X, t_j) \quad (14)$$

Among these groups of upper bounds of Eqs. (12)–(14), PIC involves the item having the maximum existential probability $PM_1(y_r, t_j)$ in the proper prefix of y_r , whereas IC (used by $DISC$ -growth) involves the item having the maximum existential probability $TM_1(t_j)$ in the proper prefix of y_r as well as its suffix. So, as $PM_1(y_r, t_j) \leq TM_1(t_j)$, we derive the following theoretical result:

$$PIC(X, t_j) \leq IC(X, t_j) \quad (15)$$

Moreover, IC also uses $P(x_k, t_j)$, whereas TC uses $TM_2(t_j)$ —which may not even involve any items in X —when $x_k \neq y_g$. So, as $P(x_k, t_j) \leq TM_2(t_j)$, we get the following result:

$$IC(X, t_j) \leq TC(X, t_j) \tag{16}$$

Hence, it is generally that

$$PIC(X, t_j) \leq IC(X, t_j) \leq TC(X, t_j) \tag{17}$$

i.e., PIC generally provides the tightest upper bounds to expected support when mining frequent 2-itemsets from high volumes of high-value uncertain data.

When mining 3^+ -itemsets, the following property holds:

$$CUF^*(X, t_j) \leq TC(X, t_j) \tag{18}$$

This is due to the extra multiplication term $[TM_3(t_j)]^{k-2}$ in CUF^* such that $0 < [TM_3(t_j)]^{k-2} \leq 1$. Hence, CUF^* provides tighter upper bounds to expected support than TC when mining frequent 3^+ -itemsets from high volumes of high-value uncertain data. Similar comments, due to the same reason, apply to $DISC^*$ (when compared with $DISC$), as well as PUF^* and $BLIMP$ (when both compared with PIC):

$$DISC^*(X, t_j) \leq DISC(X, t_j) \tag{19}$$

$$PUF^*(X, t_j) \leq PIC(X, t_j) \tag{20}$$

$$BLIMP(X, t_j) \leq PIC(X, t_j) \tag{21}$$

After analyzing the intra-group relationships among the aforementioned algorithms, let us analyze the inter-group relationships among CUF^* , $DISC^*$, PUF^* , and $BLIMP$ when they mine 3^+ -itemsets. If $x_k = y_g$, then the following property holds:

$$DISC^*(X, t_j) = CUF^*(X, t_j) \tag{22}$$

because $P(x_k, t_j) = P(y_g, t_j) = TM_1(t_j)$. The same property also holds when $x_k = y_s$ because $P(x_k, t_j) = P(y_s, t_j) = TM_2(t_j)$. Hence, when x_k is associated with the highest or the second highest existential probability in t_j , both $DISC^*$ and CUF^* provide the same upper bounds to expected support when mining frequent 3^+ -itemsets. Moreover, if $x_k \neq y_g$ and $x_k \neq y_s$, then the following property holds:

$$PUF^*(X, t_j) \leq DISC^*(X, t_j) \tag{23}$$

because both $PM_1(y_r, t_j) \leq TM_1(t_j)$ and $PM_2(y_r, t_j) \leq TM_2(t_j)$. Hence, when x_k does not associated with the highest or the second highest existential probability in t_j , PUF^* provides tighter upper bounds to expected support than $DISC^*$.

Furthermore, if $P(x_{k-1}, t_j) = PM_1(y_r, t_j)$ and $P(x_i, t_j) = P(y_i, t_j)$ for $i \in [1, k - 2]$, then we obtain:

$$BLIMP(X, t_j) = expSup(X, t_j) \tag{24}$$

Hence, when X is the first k items in t_j such that $P(x_{k-1}, t_j)$ happens to be the highest existential probability in the proper prefix $\{y_1, \dots, y_{r-1}\} \subset t_j$, *BLIMP* provides upper bounds that are so tight that they are indeed the expected support.

Note that all the aforementioned algorithms do not generate any false negatives. With tighter upper bounds to expected support, fewer false positives are produced. Hence, shorter runtime is needed to verify whether or not a pattern is true positive (i.e., frequent) or false positive (i.e., potentially frequent w.r.t. upper bounds but truly infrequent w.r.t. *minsup*).

In terms of memory consumption, the aforementioned frequent pattern mining algorithms are all tree-based. The number of nodes in the corresponding tree is small. With appropriate item ordering, the number of tree nodes for uncertain big data mining is identical to that of the FP-tree [18] for mining precise data. Note that each node in the FP-tree captures an item x and its actual support, respectively. Conversely, when mining 2-itemsets, each tree node captures x and its *TC* for CUF-growth. Similarly, each tree node captures x and *DISC* for DISC-growth; and each tree node captures x and *PIC* for PUF-growth. When mining 3^+ -itemsets, each tree node captures an additional information such as $TM_3(t_j)$ for the CUF*-growth algorithm, $TM_2(t_j)$ or $TM_3(t_j)$ for the DISC*-growth algorithm, $PM_2(y_r, t_j)$ for PUF*-growth, as well as $P(y_i, t_j)$ for BLIMP-growth, respectively.

It should be noted, as these theoretical results allow us to find tight upper bounds to expected support for frequent pattern mining problems over uncertain big data, they also introduce the nice amenity of effectively lowering the overall algorithm runtime efficiently. This will be completely demonstrated in our experimental assessment and analysis in Sect. 4.

4 Experimental Assessment and Evaluation

In this section, we evaluate several performance aspects on the optimization opportunities offered by the six different upper bounds to expected support described in Sect. 3. As regards the data layer of our experimental campaign, we considered the following well-known datasets: (i) IBM synthetic dataset, and (ii) mushroom dataset from the UC Irvine Machine Learning Depository. In particular, these datasets have been artificially made uncertain via a simple sampling-based routine that injects the existential probabilities as associated to the values of a pre-determined sub-set of attributes of the input dataset. As regards metrics, we focused on the following experimental benchmarks: (i) memory consumption, (ii) accuracy, and (iii) runtime. The final goal of our experimental campaign is to provide a comparative analysis and confirm our analytical findings provided in Sect. 3.

4.1 Memory Consumption Analysis

First, we analytically evaluate the memory consumption of the different approximations. Among them, we observe the following main behaviors that are relevant to our research:

- CUF-growth (which uses TC) requires the least amount of memory space because they are solely dependent on transaction t_j . In other words, only a single value (TC) is needed for each transaction t_j .
- CUF*-growth (which uses CUF^*) requires slightly more memory space because two values—both TC and $TM_3(t_j)$ —are needed for each transaction t_j in order to compute the CUF^* value for patterns of different cardinality k . Both CUF-growth and CUF*-growth do not need to store existential probabilities of any items in transaction t_j .
- DISC-growth and PUF-growth each requires a total of h values for each transaction t_j . Specifically, for each transaction $t_j = \{y_1, y_2, \dots, y_r, \dots, y_h\}$ with h items, a single value (IC or PIC) is needed for each item y_i in t_j .
- DISC*-growth, as an extension to DISC-growth, needs to store an additional value—namely, $TM_2(t_j)$ or $TM_3(t_j)$ depending on whether $x_k = y_g$ or y_s —for each item $x_k (= y_r)$ in transaction t_j .
- PUF*-growth, as an extension to PUF-growth, needs to store an additional value—namely, $PM_2(y_r, t_j)$ —for each item y_r in transaction t_j . Both DISC*-growth and PUF*-growth require the most amount of memory space because each of them requires a total of $2h$ values for each transaction t_j .

4.2 Accuracy Analysis

We measure the accuracy of the different frequent pattern mining algorithms when the derived theoretical upper bounds are applied. In this experiment series, we compare the tightness of the upper bounds as approximated expected support. From Sect. 3, Eqs. (12)–(14) are confirmed by results shown in Fig. 1. Note the following:

- CUF*-growth and CUF-growth lead to the same number of false positives for 2-itemsets (i.e., cardinality = 2).
- DISC*-growth and DISC-growth, as well as PUF*-growth and PUF-growth, also lead to the same number of false positives for 2-itemsets.

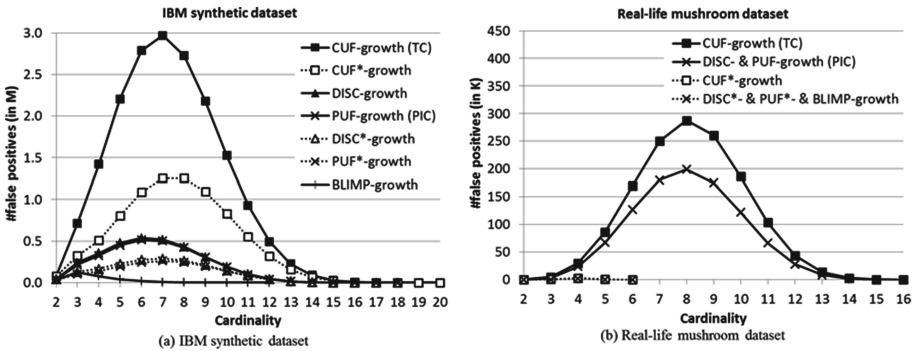


Fig. 1. Experimental results on accuracy analysis

Among these three groups of upper bounds, we also observe the following:

- PUF-growth involves the item having the maximum existential probability $PM_1(y_r, t_j)$ in the proper prefix of y_r .
- DISC-growth involves the item having the maximum existential probability $TM_1(t_j)$ in the proper prefix of y_r as well as its suffix.

As a consequence, since $PM_1(y_r, t_j) \leq TM_1(t_j)$, we can also experimentally illustrate Eq. (15). Moreover, *IC* also uses $P(x_k, t_j)$, whereas *TC* uses $TM_2(t_j)$ —which may not even involve any items in X —when $x_k \neq y_g$. So, as $P(x_k, t_j) \leq TM_2(t_j)$, we can also experimentally illustrate Eq. (16). These two experimental evidences support the observation that PUF-growth generally provides the tightest upper bounds to expected support when mining frequent 2-itemsets from high volumes of high-value uncertain data.

When mining 3^+ -itemsets, following the analysis provided in Sect. 3, we further observe the following results, which are also confirmed by our experimental evaluation (see Fig. 1):

- $DISC^*(X, t_j) \leq TC(X, t_j)$ due to the extra multiplication term $[TM_3(t_j)]^{k-2}$ in CUF*-growth such that $0 < [TM_3(t_j)]^{k-2} \leq 1$. Hence, CUF*-growth provides tighter upper bounds to expected support than CUF-growth when mining frequent 3^+ -itemsets from high volumes of high-value uncertain data.
- $DISC^*(X, t_j) \leq IC(X, t_j)$ and $PUF^*(X, t_j) \leq PIC(X, t_j)$ due to the same reason, i.e., the extra multiplication terms—which are in the range (0,1]—in DISC*-growth and PUF*-growth.

After analyzing the intra-group relationships between the aforementioned upper bounds, let us analyze the inter-group relationships among the four extensions when they mine k -itemsets, and simultaneously checking it on the experimental results shown Fig. 1 (which further confirm our theoretical analysis provided in Sect. 3):

- If $x_k = y_g$, then $DISC^*(X, t_j) = CUF^*(X, t_j)$ because $P(x_k, t_j) = P(y_g, t_j) = TM_1(t_j)$.
- If $x_k = y_s$, then $DISC^*(X, t_j) = CUF^*(X, t_j)$ because $P(x_k, t_j) = P(y_s, t_j) = TM_2(t_j)$.
- If $x_k \neq y_g$ and $x_k \neq y_s$, then $PUF^*(X, t_j) \leq DISC^*(X, t_j)$ because both $PM_1(y_r, t_j) \leq TM_1(t_j)$ and $PM_2(y_r, t_j) \leq TM_2(t_j)$.

Hence, it follows that, when x_k is associated with the highest or the second highest existential probability in t_j , both DISC*-growth and CUF*-growth provide the same upper bounds to expected support when mining frequent 3^+ -itemsets. Moreover, when x_k is not associated with the highest or the second highest existential probability in t_j , PUF*-growth provides tighter upper bounds to expected support than DISC*-growth.

The evaluation above shows the tightness of our upper bounds to expected support. Note that all these bounds do not lead to any false negatives but only false positives. The tighter the bound, the lower is the number of false positives.

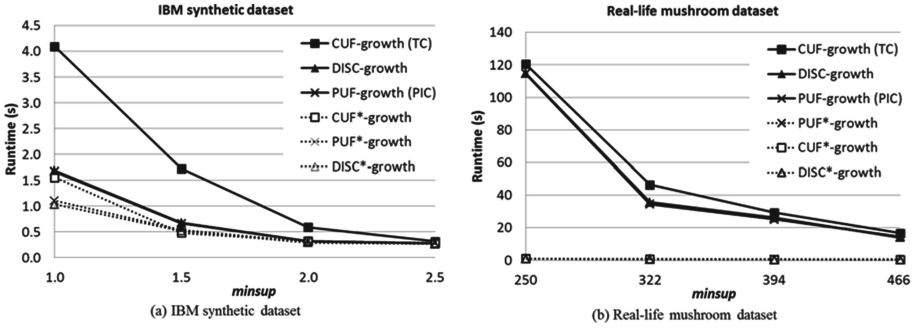


Fig. 2. Experimental results on runtime analysis

Our experimental results shown in Fig. 1 clearly support our analytical results. Specifically, CUF-growth leads to the highest numbers of false positives, whereas PUF*-growth leads to the lowest numbers (with DISC*-growth led to a close second lowest numbers) of false positives in IBM synthetic dataset and mushroom real-life dataset. Also, it is interesting to note that the tightness of the upper bound to expected support provided by the following extensions: CUF*-growth, DISC*-growth, PUF*-growth and BLIMP-growth. In fact, they do not generate any false positives beyond cardinality 6 for the mushroom dataset, as shown in Fig. 1.

4.3 Runtime Analysis

Recall that knowledge discovery and data mining algorithms use the aforementioned caps *TC*, *IC* and *PIC* to approximate expected support (see Sect. 2). The related algorithms find patterns with upper bounds to expected support meeting or exceeding the user-specified threshold *minsups*. This results in a collection of all potentially frequent 2^+ -itemsets that include both true positive (i.e., truly frequent patterns) and false positive (i.e., potentially frequent with respect to upper bounds but truly infrequent with respect to *minsups*). With tighter upper bounds to expected support, fewer false positives are produced. Hence, shorter runtimes result. Figure 2 shows overall runtime of the various alternatives using the proposed upper bounds. From the analysis Fig. 2, the following observations can be derived:

- Due to its highest number of false positives generated, CUF-growth introduces the longest runtime.
- As all three extensions (CUF*-growth, PUF*-growth and DISC*-growth) produce fewer false positives than the counterparts (CUF-growth, PUF-growth and DISC-growth), runtimes for the former are also shorter.
- As usual, when *minsups* increases, runtime decreases.
- Recall that $PUF^*(X, t_j) \leq DISC^*(X, t_j)$ if $x_k=y_g$ and $x_k=y_s$. For the cases where $x_k=y_g$ or $x_k=y_s$, it is possible (but not guaranteed) that $PUF^*(X, t_j)$

$\leq DISC^*(X, t_j)$. However, for some other cases (e.g., for short transactions in the IBM synthetic dataset or short frequent patterns mined from the real-life mushroom dataset), $DISC^*$ -growth beats PUF^* -growth.

4.4 Comparative Analysis

After evaluating the seven approximations as upper bounds to expected support, we observe the following:

- CUF-growth requires the least amount of memory space (with a single value per transaction), and CUF*-growth requires the second least amount of memory space (with two values per transaction);
- $DISC^*$ -growth and PUF^* -growth both produce fewest false positives due to the tightness of their bounds;
- $DISC^*$ -growth takes the shortest runtime, where PUF^* -growth and CUF*-growth take just slightly longer than $DISC^*$ -growth.

Hence, our recommendation is as follows: *If memory is an issue, it is better to use CUF*-growth due to its small memory requirements, few false positives and short runtimes. Otherwise, it is better to use DISC*-growth or PUF*-growth because their relatively low memory requirements (2h values for h items in a transaction) while they produce fewer false positives and run faster than others.*

5 Conclusions and Future Work

In this paper, we have examined the concepts of transaction cap TC , item cap IC and prefixed item cap PIC by viewing them as tight upper bounds to expected support of frequent k -itemsets when mining uncertain big data. Among these upper bounds, PIC provides the tightest upper bounds when mining frequent 2-itemsets, and thus produces the fewest false positives and the fastest running. When mining frequent 3^+ -itemsets, the concepts of TC , IC , and PIC have been extended to become CUF^* , $DISC^*$, PUF^* , and $BLIMP$. Our experimental results confirm our analytical findings that any of these four extensions could provide tighter upper bounds to expected support of frequent 3^+ -itemsets than the other three extensions on different mining parameters and/or distributions of uncertain data.

Future work is mainly oriented towards (i) studying optimization alternatives particularly targeted to distributed environments (e.g., *fragmentation techniques* [11, 15], which could allow us to improve the efficiency of our framework, and (ii) extending the proposed framework according to modern *big data analytics* predicates [7, 21, 39].

Acknowledgements. This project is partially supported by NSERC (Canada) and University of Manitoba.

References

1. Aggarwal, C.C., Li, Y., Wang, J., Wang, J.: Frequent pattern mining with uncertain data. In: ACM KDD 2009, pp. 29–37 (2009)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: VLDB 1994, pp. 487–499 (1994)
3. Ahmed, A.U., Ahmed, C.F., Samiullah, M., Adnan, N., Leung, C.K.: Mining interesting patterns from uncertain databases. *Inf. Sci.* **354**, 60–85 (2016)
4. Aryadinata, Y.S., Lin, Y., Barcellos, C., Laurent, A., Libourel, T.: Mining epidemiological dengue fever data from Brazil: a gradual pattern based geographical information system. In: Laurent, A., Strauss, O., Bouchon-Meunier, B., Yager, R.R. (eds.) IPMU 2014, Part II. CCIS, vol. 443, pp. 414–423. Springer, Heidelberg (2014)
5. Calders, T., Garboni, C., Goethals, B.: Efficient pattern mining of uncertain data with sampling. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010, Part I. LNCS (LNAI), vol. 6118, pp. 480–487. Springer, Heidelberg (2010)
6. Chen, L., Liu, C., Zhang, C.: Mining probabilistic representative frequent patterns from uncertain data. In: SIAM SDM 2013, pp. 73–81 (2013)
7. Cuzzocrea, A.: Analytics over big data: exploring the convergence of data warehousing, OLAP and data-intensive cloud infrastructures. In: IEEE COMPSAC 2013, pp. 481–483 (2013)
8. Cuzzocrea, A.: Approximate OLAP query processing over uncertain and imprecise multidimensional data streams. In: Decker, H., Lhotská, L., Link, S., Basl, J., Tjoa, A.M. (eds.) DEXA 2013, Part II. LNCS, vol. 8056, pp. 156–173. Springer, Heidelberg (2013)
9. Cuzzocrea, A.: Retrieving accurate estimates to OLAP queries over uncertain and imprecise multidimensional data streams. In: Cushing, J.B., French, J., Bowers, S. (eds.) SSDBM 2011. LNCS, vol. 6809, pp. 575–576. Springer, Heidelberg (2011)
10. Cuzzocrea, A., Bellatreche, L., Song, I.-Y.: Data warehousing and OLAP over big data: current challenges and future research directions. In: ACM DOLAP 2013, pp. 67–70 (2013)
11. Cuzzocrea, A., Darmont, J., Mahboubi, H.: Fragmenting very large XML data warehouses via K-means clustering algorithm. *Int. J. Bus. Intell. Data Min.* **4**(3/4), 301–328 (2009)
12. Cuzzocrea, A., Leung, C.K.: Upper bounds to expected support for frequent itemset mining of uncertain big data. In: ACM SAC 2015, pp. 919–921 (2015)
13. Cuzzocrea, A., Leung, C.K., MacKinnon, R.K.: Mining constrained frequent itemsets from distributed uncertain data. *Future Gener. Comput. Syst.* **37**, 117–126 (2014)
14. Cuzzocrea, A., Saccà, D., Ullman, J.D.: Big data: a research agenda. In: IDEAS 2013, pp. 198–203 (2013)
15. Daenen, J., Neven, F., Tan, T.: Gumbo: guarded fragment queries over big data. In: EDBT 2015, pp. 521–524 (2015)
16. Diaz-Valenzuela, I., Martin-Bautista, M.J., Vila, M.-A.: A fuzzy semisupervised clustering method: application to the classification of scientific publications. In: Laurent, A., Strauss, O., Bouchon-Meunier, B., Yager, R.R. (eds.) IPMU 2014, Part I. CCIS, vol. 442, pp. 179–188. Springer, Heidelberg (2014)
17. Fan, W., Bifet, A.: Mining big data: current status, and forecast to the future. *ACM SIGKDD Explor.* **14**(2), 1–5 (2012)

18. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM SIGMOD 2000, pp. 1–12 (2000)
19. Hodáková, P., Perfilieva, I., Hurtík, P.: F-transform and its extension as tool for big data processing. In: Laurent, A., Strauss, O., Bouchon-Meunier, B., Yager, R.R. (eds.) IPMU 2014, Part III. CCIS, vol. 444, pp. 374–383. Springer, Heidelberg (2014)
20. Jiang, F., Kawagoe, K., Leung, C.K.: Big social network mining for “following” patterns. In: C3S2E 2015, pp. 28–37 (2015)
21. Jiang, F., Leung, C.K.: A data analytic algorithm for managing, querying, and processing uncertain big data in cloud environments. *Algorithms* **8**(4), 1175–1194 (2015)
22. Jiang, F., Leung, C.K., Liu, D.: Efficiency improvements in social network communication via MapReduce. In: IEEE DSDIS 2015, pp. 161–168 (2015)
23. Leung, C.K.: Big data mining applications and services. In: BigDAS 2015, pp. 1–8 (2015)
24. Leung, C.K.: Uncertain frequent pattern mining. In: Aggarwal, C.C., Han, J. (eds.) *Frequent Pattern Mining*, pp. 417–453. Springer, Switzerland (2014)
25. Leung, C.K., Cuzzocrea, A.: Frequent subgraph mining from streams of uncertain data. In: C3S2E 2015, pp. 18–27 (2015)
26. Leung, C.K.-S., Cuzzocrea, A., Jiang, F.: Discovering frequent patterns from uncertain data streams with time-fading and landmark models. In: Hameurlain, A., Küng, J., Wagner, R., Cuzzocrea, A., Dayal, U. (eds.) TLDKS VIII. LNCS, vol. 7790, pp. 174–196. Springer, Heidelberg (2013)
27. Leung, C.K., Jiang, F., Pazdor, A.G.M., Peddle, A.M.: Parallel social network mining for interesting ‘following’ patterns. *Concurrency Computat. Pract. Exper.* (2016). doi:[10.1002/cpe.3773](https://doi.org/10.1002/cpe.3773)
28. Leung, C.K.-S., MacKinnon, R.K.: BLIMP: a compact tree structure for uncertain frequent pattern mining. In: Bellatreche, L., Mohania, M.K. (eds.) DaWaK 2014. LNCS, vol. 8646, pp. 115–123. Springer, Heidelberg (2014)
29. Leung, C.K., MacKinnon, R.K., Tanbeer, S.K.: Tightening upper bounds to expected support for uncertain frequent pattern mining. *Procedia Comput. Sci.* **35**, 328–337 (2014)
30. Leung, C.K.-S., Mateo, M.A.F., Brajczuk, D.A.: A tree-based approach for frequent pattern mining from uncertain data. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 653–661. Springer, Heidelberg (2008)
31. Leung, C.K.-S., Tanbeer, S.K.: Fast tree-based mining of frequent itemsets from uncertain data. In: Lee, S., Peng, Z., Zhou, X., Moon, Y.-S., Unland, R., Yoo, J. (eds.) DASFAA 2012, Part I. LNCS, vol. 7238, pp. 272–287. Springer, Heidelberg (2012)
32. Leung, C.K.-S., Tanbeer, S.K.: PUF-Tree: a compact tree structure for frequent pattern mining of uncertain data. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013, Part I. LNCS (LNAI), vol. 7818, pp. 13–25. Springer, Heidelberg (2013)
33. Li, X., Wang, Y., Li, X., Wang, X., Yu, J.: GDPS: an efficient approach for skyline queries over distributed uncertain data. *Big Data Res.* **1**, 23–36 (2014)
34. Liu, C., Chen, L., Zhang, C.: Summarizing probabilistic frequent patterns: a fast approach. In: ACM KDD 2013, pp. 527–535 (2013)
35. Liu, Y.-H.: Mining time-interval univariate uncertain sequential patterns. *Data Knowl. Eng.* **100**, 54–77 (2015)

36. MacKinnon, R.K., Leung, C.K.-S., Tanbeer, S.K.: A scalable data analytics algorithm for mining frequent patterns from uncertain data. In: Peng, W.-C., Wang, H., Bailey, J., Tseng, V.S., Ho, T.B., Zhou, Z.-H., Chen, A.L.P. (eds.) PAKDD 2014 Workshops. LNCS (LNAI), vol. 8643, pp. 404–416. Springer, Heidelberg (2014)
37. MacKinnon, R.K., Strauss, T.D., Leung, C.K.: DISC: efficient uncertain frequent pattern mining with tightened upper bounds. In: IEEE ICDM 2014 Workshops, pp. 1038–1045 (2014)
38. Nguyen, H.T.H., Cao, J.: Trustworthy answers for top-k queries on uncertain big data in decision making. *Inf. Sci.* **318**, 73–90 (2015)
39. Pei, J.: Some new progress in analyzing and mining uncertain and probabilistic data for big data analytics. In: Ciucci, D., Inuiguchi, M., Yao, Y., Ślęzak, D., Wang, G. (eds.) RSFDGrC 2013. LNCS (LNAI), vol. 8170, pp. 38–45. Springer, Heidelberg (2013)
40. Petry, F.E.: Data mining approaches for geo-spatial big data: uncertainty issues. *Int. J. Organ. Collective Intell.* **3**(1), 52–71 (2012)
41. Rahman, Q.M., Fariha, A., Mandal, A., Ahmed, C.F., Leung, C.K.: A sliding window-based algorithm for detecting leaders from social network action streams. In: IEEE/WIC/ACM WI-IAT 2015, vol. 1, pp. 133–136 (2015)
42. Saati, S., Hatami-Marbini, A., Tavana, M., Agrell, P.J.: A fuzzy data envelopment analysis for clustering operating units with imprecise data. *Int. J. Uncertainty Fuzziness Knowl. Based Syst.* **21**(1), 29–54 (2013)
43. Samet, A., Lefèvre, E., Ben Yahia, S.: Classification with evidential associative rules. In: Laurent, A., Strauss, O., Bouchon-Meunier, B., Yager, R.R. (eds.) IPMU 2014, Part I. CCIS, vol. 442, pp. 25–35. Springer, Heidelberg (2014)
44. Tong, Y., Chen, L., Cheng, Y., Yu, P.S.: Mining frequent itemsets over uncertain databases. *PVLDB* **5**(11), 1650–1661 (2012)
45. Xu, J., Li, N., Mao, X.-J., Yang, Y.-B.: Efficient probabilistic frequent itemset mining in big sparse uncertain data. In: Pham, D.-N., Park, S.-B. (eds.) PRICAI 2014. LNCS (LNAI), vol. 8862, pp. 235–247. Springer, Heidelberg (2014)