A Synopsis Based Approach for Itemset Frequency Estimation over Massive Multi-Transaction Stream

GUANGTAO WANG, University of Michigan and JD AI Research
GAO CONG, Nanyang Technological University
YING ZHANG, University of Technology
ZHEN HAI, Institute for Infocomm Research, A∗STAR
JIEPING YE, University of Michigan

The streams where multiple transactions are associated with the same key are prevalent in practice, e.g., a customer has multiple shopping records arriving at different time. Itemset frequency estimation on such streams is very challenging since sampling based methods, such as the popularly used reservoir sampling, cannot be used. In this article, we propose a novel k-Minimum Value (KMV) synopsis based method to estimate the frequency of itemsets over multi-transaction streams. First, we extract the KMV synopses for each item from the stream. Then, we propose a novel estimator to estimate the frequency of an itemset over the KMV synopses. Comparing to the existing estimator, our method is not only more accurate and efficient to calculate but also follows the downward-closure property. These properties enable the incorporation of our new estimator with existing frequent itemset mining (FIM) algorithm (e.g., FP-Growth) to mine frequent itemsets over multi-transaction streams. To demonstrate this, we implement a KMV synopsis based FIM algorithm by integrating our estimator into existing FIM algorithms, and we prove it is capable of guaranteeing the accuracy of FIM with a bounded size of KMV synopsis. Experimental results on massive streams show our estimator can significantly improve on the accuracy for both estimating itemset frequency and FIM compared to the existing estimators.

CCS Concepts: • **Information systems → Database transaction processing;**

Additional Key Words and Phrases: Data stream mining, massive multi-transaction stream data, k-minimum value synopsis, downward-closure estimator, itemset frequency estimation, ϵ-close frequent itemset mining

ACM Reference format:
Guangtao Wang, Gao Cong, Ying Zhang, Zhen Hai, and Jieping Ye. 2021. A Synopsis Based Approach for Itemset Frequency Estimation over Massive Multi-Transaction Stream. *ACM Trans. Knowl. Discov. Data.* 16, 2, Article 29 (July 2021), 30 pages.
https://doi.org/10.1145/3465238

Author's addresses: G. Wang, University of Michigan, MI 48109, and JD AI Research, CA94539; email: xjtuwgt@gmail.com; G. Cong, Nanyang Technological University, Singapore 639798; email: GAOCONG@ntu.edu.sg; Y. Zhang, University of Technology Sydney, Ultimo, NSW 2007, Australia; email: Ying.Zhang@uts.edu.au; Z. Hai, Institute for Infocomm Research, A∗STAR, Singapore 138632; email: HAIZ0001@ntu.edu.sg; J. Ye, University of Michigan, MI 48109; email: jpye@umich.edu.

1556-4681/2021/07-ART29 $15.00
https://doi.org/10.1145/3465238

1 INTRODUCTION

The task of exploring the frequency of itemsets over massive stream is a fundamental problem and arises in a variety of applications, such as frequent itemset mining on stream [26, 34, 42, 43], e-business [49], market-basket analysis [8, 23], attack/fake review dection [6, 50], and so on. As it is too expensive to compute the accurate frequencies of itemsets over such massive streams, various frequency estimation methods have been proposed for estimating the frequencies of individual items [12, 16–18, 20, 31, 36, 40, 43, 54, 63] and the frequency of itemsets [22, 38, 53, 55, 59, 60] over streams. These methods assume all the items related to a key arrive together. However, in reality, a key may correspond to multiple transactions, each of which contains a set of items and arrives at different time. For example, (1) Streaming check-in data. Each key corresponds to a user and each transaction records a place visited by the user. Other examples of similar types of data include phone record data, chat log data, and so on. (2) E-business stream. Each customer of online shopping sites, e.g., Amazon, has an ID and may have multiple transactions, which arrive at different time and interleave with the transactions of other users. Similar streams are encountered in many services, such as interactions of users with Web services, search logs, film/music/App downloading stream, twitter stream, and so on. (3) Multi-part uploading stream. Each large record is divided into multiple transactions, and each transaction is uploaded separately.

The common characteristic of these streams is that multiple transactions are associated with the same key and arrive in different time. We call such streams as "multi-transaction stream". In this article, we focus on the problem of itemset frequency estimation over such stream. In a multi-transaction stream, the frequency of an itemset is incremented if its items appear in the union of all transactions with the same key. However, in previous work on frequency estimation, as each transaction has a unique key, the frequency of an itemset is the number of transactions containing the itemset.

There are many types of multi-transaction data stream, application range of itemset mining on them is very wide also. If we view the key of each transaction as user-id, frequent itemset mining over multi-transaction stream generates itemsets at the user level. These user-level itemsets reveal the users' behavior directly, and will benefit personalized recommendation, sub-group discovery, and any downstream data mining task (such as classification, clustering, association rule mining, etc.) at the user level rather than single transaction level. Moreover, the user-level itemset mining will be robust to the bots or, increasingly, hack users accounts and place fake orders on website. Such as, deliberately generating abundant shopping transactions in short time (with same user id/ip address) to create false popular products in e-business, or producing abundant comments on social networks to create false hot events. In these situations, the multi-transaction based itmeset mining, i.e., user-level itemset frequency is able to naturally filter the impact of such fake transactions. This is very useful in security attack analysis, fake review/comment/event detection, and so on.[1]

Unfortunately, the difference in itemset frequency definition makes the existing methods [3, 19, 38, 45, 56, 58, 61] of frequency estimation inapplicable to the multi-transaction stream. The existing approaches, such as reservoir sampling based synopsis [45, 58, 61], AMS sketch [3], and count-min sketch [19], usually make a single pass of the stream and get a synopsis of the original stream to estimate the frequency of itemsets using a bounded amount of memory, and they have been widely

[1] Taking fake review detection as an example, we can first mine the user-level frequent word patterns (using our proposed method) and transation-level ones (using the existing methods) from the data stream, respectively; and then find out the pattern differernce for fake review detection. Generally speaking, the transaction-level patterns not appearing in user-level are more likely to be "irregular patterns".

used for item frequency estimation. We will illustrate with reservoir sampling as an example why these synopsis based methods are not applicable on multi-transaction stream in Section 3.

In this article, we demonstrate that, estimating the frequency of an itemset over multi-transaction streams can be done by counting the number of distinct keys appearing in a set of transactions containing the itemset (See details in Section 3.1). This motivates us to employ k **Minimum Value** (**KMV**) synopsis for estimating itemset frequency, which is a well-known technique to estimate the number of distinct values in massive data [7] and is also used for stream sampling [16] and estimating word frequencies in twitter stream [59, 60].

However, the estimator in [7, 59] suffers from the following issues which affect the effectiveness of frequency estimation for l-itemsets ($l > 1$). (i) The accuracy of the estimator drops as the length of itemsets (i.e., l) increases. (ii) It does not follow the well-known downward-closure property of frequency, and the frequency estimation will be counter-intuitive. For example, an itemset A might get a greater estimation than its subset $B \subset A$. Furthermore, an important application of frequency estimation is to mine frequent itemsets from streams. The downward-closure property plays the foundation of classic frequent itemset mining algorithms [2, 29], which cannot be used without this property. We find out that the root of these issues of the estimator is how the KMV synopsis is constructed for a union set (see details in Section 4.1).

To address these issues, we propose a new method to construct KMV synopsis for union sets, and further present a novel frequency estimator. The new estimator not only has a smaller estimation error but also follows the downward-closure property. We also prove that the error of the estimated frequency can be bounded theoretically. These theoretical results will greatly benefit the generalization of KMV synopsis based itemset frequency estimation to other applications over multi-transaction streams. For example, with the downward-closure property, we can incorporate the proposed estimator into any existing **frequent itemset mining** (**FIM**) method to mine frequent itemsets over multi-transaction streams, which is also a significant research problem in stream analysis. According to theoretical analysis and our experiments, we can achieve very high accuracy of FIM by setting a small size of KMV synopsis. Our contributions are as follows:

(1) We define an important research problem of itemset frequency estimation on massive multi-transaction stream.

(2) We propose a new KMV synopsis based estimator for frequency estimation on l-itemsets ($l > 1$), which is able to significantly improve the estimation accuracy comparing to the previous KMV based estimator [7, 59, 60]. We show that the new estimator follows downward-closure property and thus can be used in conjunction with any itemset mining algorithm over KMV synopsis.

(3) We integrate the proposed estimator into existing frequent itemset mining algorithms to solve ϵ-close FIM problem over multi-transaction streams, and theoretically establish the connection between the size of KMV synopsis and the accuracy of FIM. This enables us to support the mechanism of setting the size of synopsis to achieve a certain level of accuracy. However, the previous KMV synopsis based method [59, 60] fails to support this.

The rest of the article is organized as follows. Section 2 introduces the related work. Section 3 gives the problem definition and preliminary. Section 4 presents our proposed KMV synopsis based frequency estimator and its application in mining frequent itesmets. Section 5 presents the experimental results. Finally, Section 6 concludes our work.

2 RELATED WORK

Itemset Fequency estimation over stream data. Frequency estimation for itemsets in massive stream is a fundamental task and has a wide variety of applications. To handle massive streams, the synopsis, which can be viewed as a sample or summary of the entire stream, has been widely used for stream data analysis [1, 21, 38, 39, 45, 54, 61, 62]. Particularly, many synopsis-based estimation methods have been proposed [5, 12, 16–18, 20–22, 30, 31, 38, 40, 43, 58, 63], which maintain a random sample of the stream (e.g., [45, 61]) or the hashing based sketch (e.g., Count-min sketch [19] and AMS sketch [3]). However, these techniques are proposed for traditional streams where each transaction is associated with a unique key, and they are not applicable to multi-transaction stream data, such as reservoir sampling based approach [58], cannot be used for multi-transaction stream data.

To estimate the frequency of an itemset over a multi-transaction stream, we need to estimate the number of distinct keys which correspond to the set of transactions containing the itemset. KMV synopsis [16, 17], is a well-known technique for estimating the number of distinct values in massive data, and also has been studied for set intersection size estimation [7, 59, 60]. However, a straightforward extension for KMV-based estimator will lead to high estimation error especially when the length of itemsets increases [59, 60].

Frequent itemset mining over stream data. Frequent itemset mining has been an important research problem for stream data analysis. However, stream data challenge traditional FIMs. First, the streaming data comes continuously and is unbounded, it is impossible to keep all transactions in the main memory for analysis. Secondly, the streaming data are passed only once as thus traditional FIM method scanning data multiple passes is infeasible. For example, the well-known FP-growth [28] needs two-pass scan over the data for FIM. Thirdly, stream data analysis usually requires real-time response, and the combinational explosion of itemsets challenges mining frequent itemsets on stream in terms of both time and space (memory) efficiency.

Due to these challenges, the research studies on FIM have focused on approximating mining methods, which generate frequent itemsets within the stream's constrained environment of limited time and memory. And there have been different window based models proposed in the literature [23, 24, 35, 41, 53]. A window model usually extracts a set of transactions from the data stream for FIM. The difference among these models is that the development of the window size and how to assign a weight to each transaction extracted from the stream [53]. These window models are generally catergorized as three groups:

(i) The landmark window model [64], which collects transactions of the stream from some fixed landmark start time up to current transaction, and assigns same weight to each transaction. The representative algorithms include: Lossy Counting Algorithm [42] first buffers the transactions in as many blocks as memory available, then all buffered blocks together as a single batch, and generates the itemsets that are supported by the transactions in the current batch. In order to avoid the combinatorial explosion of the itemsets, it applies an Apriori-like pruning rule. FDPM [62] is derived from the Chernoff bound to guarantee the estimation error being bounded. And SApriori [51] divides stream into multiple blocks, and mines frequent itemset with some particular length on each block with an Apriori algorithm [2].

(ii) The sliding window [10, 13, 14, 33, 37, 52, 57], which extracts the most recent transactions for analysis, the window usually has a fixed size, and the new arriving transaction results in that the oldest transaction is deleted from the window, where each transaction in the filled window has a constant weight. The representative algorithms include: Moment [15] designs an in-memory prefix-tree-based structure, called the **Closed Enumeration Tree** (**CET**), to maintain a dynamically selected set of itemsets over the sliding-window. SWIM [44], which divides a window into several panes of identical size. A delay parameter controls the maximal number of panes processed before reporting a frequent itemset. CPS-TreeSW [52] designs a dynamic reorganizing tree structure over the sliding window that is mined with the FP-growth algorithm [28].

(iii) Time fading/damped window [25, 26, 64] where the transactions are associated with differernt weights, and the more rencent transactions have a higher weight than older ones. With

Table 1. Summary of Notations

Notation	Meaning
D	a multi-transaction stream
X	an l-itemset $X = \{x_i \mid 1 \le i \le l\}$
$\mathcal{L}$	the KMV synopsis of D
K	the size of KMV synopsis $\mathcal{L}$ extracted from D
$\mathbb{D}$	the set of transactions extracted by $\mathcal{L}$
D_{x_i}	the set of transactions that contain x_i and the other transactions with the same keys as those containing x_i
$\mathcal{L}_{x_i}$	the KMV synopsis of D_{x_i}
$\Phi(D)$	the function Φ that returns distinct keys in D
k_{x_i}	the size of $\mathcal{L}_{x_i}$
$U_{(k_{x_i})}$	the k_{x_i}th smallest value of $\mathcal{L}_{x_i}$
$K_\cap$	size of intersection set of l KMV synopses $\lvert \cap_{i=1}^{l} \mathcal{L}_{x_i} \rvert$
$D_\cup$	the set of distinct keys in $\cup_{i=1}^{l} D_{x_i}$, i.e., $\Phi(\cup_{i=1}^{l} D_{x_i})$
$D_\cap$	the set of distinct keys in $\cap_{i=1}^{l} D_{x_i}$, i.e., $\Phi(\cap_{i=1}^{l} D_{x_i})$
Δ	the maximum number of items in the union of all transactions with same key

each new coming transaction, the weight of all transactions decays by a factor and thus the long ago transactions will eventually be removed from consideration. One representative algorithm is EstDec [11], which finds recent frequent itemsets adaptively over an online data stream. The effect of old transactions on the mining result of the data stream is diminished by decaying the old occurrences of each itemset as time goes by.

All these window based models are desinged for approximation FIM over single-transaction data stream [53]. However, these algorithms cannot be directly applied for multi-transaction data streams as the frequency definition is different over multi-transaction stream (see Example 1 in Section 3.1 for details). The algorithm for mining co-occurrence words in streaming tweets [59, 60], which uses the existing KMV synopsis based estimator in [7], can be used for frequency estimation over multi-transaction stream. But it suffers from low accuracy as shown in our experiments and has no terminating condition in theory since the estimator does not satisfy the downward closure property. In this article, we first give the theoretical analysis of the root these issues of existing KMV synopsis based estimators, then propose a new accurate and down-closure itemset frequency estimator, and finally give the approximate FIM algorithms over multi-transaction stream data with the proposed estimator.

3 PROBLEM DEFINITION AND PRELIMINARIES

We first introduce the concept of multi-transaction streams, and give the problem definition of itemset frequency estimator on such streams. Afterwards, we present existing KMV synopsis based itemset frequency estimators [7, 59, 60]. For clarity, Table 1 summarizes the notations used in this article.

3.1 Problem Definition.

Given a data stream $D = \{T_1, T_2, \ldots, T_N\}$ comprising a sequence of N transactions observed so far, where each transaction T_i, denoted by $<tid_i, Y_i>$ $(1 \le i \le N)$, contains a set of items (denoted by Y_i) and is associated with a key tid_i. In traditional streams, each transaction has a unique key, i.e., $\forall i \ne j$, $tid_i \ne tid_j$. The stream D is called a multi-transaction stream if multiple transactions are associated with the same key.

Let $\Phi(D)$ denote the set of distinct keys in D, and $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$ be the set of items in D. We assume that $N \gg n$. An itemset $X = \{x_1, x_2, \ldots, x_l\}$ ($l \geq 1$) is called an l-itemset, where $x_i \in \mathcal{I}$ ($1 \leq i \leq l$). In a multi-transaction stream, the frequency of itemset X is incremented when X appears in the union set of items in all transactions with the same key. Let $B_{tid} = \{T_i \mid T_i.tid_i = tid \wedge T_i \in D\}$ be a bag of all transactions whose key is tid, and X is counted once if $X \subseteq \cup_{T_i \in B_{tid}} T_i.Y_i$, where $T_i = <tid_i, Y_i>$.

For a single item x, let $\mathcal{T}_x = \{T_i.tid_i | x \in T_i.Y_i\}$ be a set of keys of transactions containing x, and $D_x = \{T_i \mid T_i.tid_i \in \mathcal{T}_x \wedge T_i \in D\}$. It is noted that D_x not only includes transactions containing x but also consists of the other transactions which have the same key as the included transactions. The frequency of item x is $|\Phi(D_x)|$, where $\Phi(D_x)$ is the set of distinct keys in D_x.

For an l-itemset X ($l > 1$), let D_{x_i} be a set of transactions for each $x_i \in X$ and $D_\cap = \Phi(\cap_{i=1}^{l} D_{x_i})$. Then, frequency of X (denoted as $freq(X)$ in Equation (1)) is the number of distinct keys in $\cap_{i=1}^{l} D_{x_i}$.

$$freq(X) = |D_\cap| = |\Phi(\cap_{i=1}^{l} D_{x_i})|. \tag{1}$$

Example 1. *Suppose there is a multi-transaction stream D consisting of five transactions $\{< 1, \{a\} >, < 2, \{b\} >, < 1, \{b, c\} >, < 2, \{a, c\} >, < 3, \{b\} >\}$. To estimate the frequency of a given itemset $\{a, b\}$, we first construct $D_a = \{< 1, \{a\} >, < 2, \{b\} >, < 1, \{b, c\} >, < 2, \{a, c\} >\}$ and $D_b = \{< 1, \{a\} >, < 2, \{b\} >, < 1, \{b, c\} >, < 2, \{a, c\} >, < 3, \{b\} >\}$. Then $D_a \cap D_b = \{< 1, \{a\} >, < 2, \{b\} >, < 1, \{b, c\} >, < 2, \{a, c\} >\}$. The itemset $\{a, b\}$ appears twice in the union of transactions with same keys. Thus, the estimated frequency of $\{a, b\}$ is two, which is just the size of $D_\cap = \Phi(D_a \cap D_b) = \{1, 2\}$. Note that D_a also contains $< 2, \{b\} >$ and $< 1, \{b, c\} >$ although they do not contain a. This is critical. In contrast, if we construct $D_a = \{< 1, \{a\} >, < 2, \{a, c\} >\}$ and $D_b = \{< 1, \{b\} >, < 2, \{b, c\} >, < 3, \{b\} >\}$, which only include the single items a and b, respectively, we will get $D_a \cap D_b = \phi$, and further calculate the frequency of $\{a, b\}$ as $\Phi(\phi) = 0$, which is wrong.*

Problem Definition. We investigate the problem of estimating the frequency of an l-itemset over massive multi-transaction stream data.

For massive stream data, it is usually impractical to compute the exact frequency of itemsets. In contrast, researchers usually leverage a synopsis sampled from the stream to get an approximate solution. However, the existing synopsis based techniques (e.g., reservoirs sampling) assume each transaction has a unique key and cannot handle the multi-transaction stream. Taking the popularly used reservoir sampling as an example, which needs to decide whether to keep a new key, and discard an old key so that the set of selected keys are a random sample of the stream. In a multi-transaction stream, the transactions with same key might arrive at different time. For a new transaction, we have to check whether its key has been scanned before we can decide if it will be kept or discarded (if it has been scanned. the decision depends on the existing decision). However, this requires us to store all the scanned keys and record whether they are selected, which is unfortunately memory-consuming. It is not practical to store all the scanned keys in memory on massive streams. Let M be the number of distinct keys in the stream, and Δ be the maximum number of items in the union of all transactions with same key, the worst space comlexity of reservoir sampling over multi-transaction stream is $O(M \cdot \Delta)$. In practice, M is usually quite large and results in large memory consumption.

In this article, we propose a new KMV synopsis to address the problem. Our proposed solution is based on the entire data streams. However, it is easy to extend our estimator to estimate the frequency of an itemset based on sliding window model.

3.2 Preliminaries

The frequency of a single item x is the number of distinct keys in D_x, i.e., $\Phi(D_x)$. Based on the work [7, 16], the frequency of x can be quickly estimated by a KMV synopsis of D_x. Next, we briefly introduce the concepts of KMV synopsis [7, 16] and Inverted KMV Sketch for an item [59, 60], and then present the estimator [7, 59, 60] for estimating frequencies of itemsets.

Let h be a pairwise independent hash function which randomly and uniformly maps each transaction key into a value in the range of [0, 1], i.e., $h : \Phi(D) \rightarrow [0, 1]$. KMV synopsis of a stream data is defined as follows.

Definition 1. k Minimum Value Synopsis. After hashing each transaction keys of a stream data D by h, the k smallest hash values form a KMV synopsis of D, which is denoted as $\mathcal{L}$. The set of all transactions corresponding to the k minimum hash values is denoted as $\mathbb{D}$.

KMV synopsis $\mathcal{L}$ consists of k smallest hash values. Let $U_{(k)}$ be the kth smallest hash value. Then the number of distinct transaction keys in D can be estimated by Equation (2) [7, 59, 60].

$$|\Phi(\hat{D})| = \frac{k-1}{U_{(k)}}. \tag{2}$$

The Inverted KMV Sketch [59, 60] of an item x appearing in $\mathbb{D}$ is defined as follows.

Definition 2. Inverted KMV Sketch. The inverted KMV sketch of x is defined by $\mathcal{L}_x = \{h(tid) \mid T = <tid, Y> \in \mathbb{D} \wedge x \in Y\}$.

As the hash values generated by h follow the uniform probability distribution [7], the inverted KMV sketch $\mathcal{L}_x$ is a KMV synopsis of sub stream data $D_x \subseteq D$ [59, 60]. Thus, with Equation (2), we estimate the frequency of item x (i.e., 1-itemset) by $freq(x) = |\Phi(\hat{D}_x)| = \frac{k_x - 1}{U_{(k_x)}}$, where k_x is the size of $\mathcal{L}_x$, and $U_{(k_x)}$ is the k_xth smallest value in $\mathcal{L}_x$.

Frequency Estimator for an l-itemset. For a given l-itemset $X = \{x_i | 1 \leq i \leq l\}$ $(l > 1)$, according to Equation (1), its frequency $freq(X)$ can be computed as the number of distinct keys in the intersection of l sub stream data, i.e., $D_\cap = \Phi(\cap_{i=1}^{l} D_{x_i})$.

To estimate $freq(X)$ (i.e., $|D_\cap|$), let $D_\cup = \Phi(\cup_{i=1}^{l} D_{x_i})$. If we can get the KMV synopsis of $\cup_{i=1}^{l} D_{x_i}$, we can estimate the number of distinct keys of $\cup_{i=1}^{l} D_{x_i}$ by Equation (2), and then estimate $freq(X)$ by $\rho \times |D_\cup|$ based on the Jaccard similarity $\rho = \frac{|D_\cap|}{|D_\cup|}$.

Next, we introduce a theorem [7] to construct KMV synopsis for a union set. Consider substreams D_{x_1} and D_{x_2} with respect to items x_1 and x_2, along with their KMV synopses $\mathcal{L}_{x_1}$ and $\mathcal{L}_{x_2}$ of sizes k_{x_1} and k_{x_2}, respectively. Let $\mathcal{L}_{x_1} \oplus \mathcal{L}_{x_2}$ be the set of the k smallest values in $\mathcal{L}_{x_1} \cup \mathcal{L}_{x_2}$, where $k = \min(k_{x_1}, k_{x_2})$. Then, we can get a KMV synopsis of $D_{x_1} \cup D_{x_2}$ according to Theorem 1 [7].

Theorem 1. *The set* $\mathcal{L}_\cup = \mathcal{L}_{x_1} \oplus \mathcal{L}_{x_2}$ *is a size-k KMV synopsis of* $D_{x_1} \cup D_{x_2}$, *where* $k = \min(k_{x_1}, k_{x_2})$.

Theorem 1 can be extended to multiple data sets [7]. That is, the size of $D_\cup = \Phi(\cup_{i=1}^{l} D_{x_i})$ $(l \geq 2)$ can be estimated via the KMV synopsis $\mathcal{L}_\cup = \mathcal{L}_{x_1} \oplus \mathcal{L}_{x_2} \oplus \cdots \oplus \mathcal{L}_{x_l}$ of size $k = \min(k_{x_1}, k_{x_2}, \cdots, k_{x_l})$. Let $U_{(k)}$ be the kth smallest value in $\mathcal{L}_\cup$, and $K_\cap = |\mathcal{L}_{x_1} \cap \mathcal{L}_{x_2} \cap \cdots \cap \mathcal{L}_{x_l}|$. Since $\frac{K_\cap}{k}$ is an unbiased estimator of ρ [7], the frequency of l-itemset X can be estimated by

$$fre\hat{q}(X) = |\hat{D_\cap}| = \frac{K_\cap}{k} \times \frac{k-1}{U_{(k)}}. \tag{3}$$

Algorithm 1 illustrates the details of maintaining the KMV synopsis $\mathcal{L}$ of a multi-transaction stream D and Inverted KMV sketch $\mathcal{L}_x$ for each item x by a single scan of D. For each coming transaction T, if $|\mathcal{L}|$ is smaller than the given size K, we insert the hash value of $h(T)$ into both $\mathcal{L}$ and Inverted KMV sketch $\mathcal{L}_x$ for each item x in T (lines 2–7). Otherwise, if $h(T)$ is smaller than the Kth smallest hash value of $\mathcal{L}$, we update $\mathcal{L}$ by removing its current Kth smallest value and then inserting $h(T)$ into $\mathcal{L}$; and we update $\mathcal{L}_x$ by removing $h(T')$ (T' corresponds to the transaction

ALGORITHM 1: KMV_Synopsis_Extraction()

```
   Inputs : $D$: a multi-transaction stream data, $h$: the hash function;
   Output : $\mathcal{L}$: KMV synopsis of $D$ with size $K$, $\mathcal{L}_x$: inverted KMV sketch of item $x$ or KMV synopsis of $D_x$;
 1 $\mathcal{L} \leftarrow \phi$, $\mathcal{L}_x \leftarrow \phi$;
 2 foreach coming transaction ($T$=<$tid, Y$>) $\in D$ do
 3     $v = h(tid)$//h is a hash function
 4     if $|\mathcal{L}| \leq K$ then
 5         $\mathcal{L} \leftarrow \mathcal{L} \cup \{h(tid)\}$;
 6         foreach item $x \in Y$ do
 7             $\mathcal{L}_x \leftarrow \mathcal{L}_x \cup \{h(tid)\}$;
 8     else
 9         if $v <$ KSmall($\mathcal{L}$) then
               //KSmall($\mathcal{L}$) gets the $K$-th smallest value in $\mathcal{L}$
10             $T' \leftarrow$ transaction with $h(tid')$ == KSmall($\mathcal{L}$)//$T'$=<$tid', Y'$>
11             $\mathcal{L} \leftarrow \mathcal{L} - \{h(tid')\}$;
12             foreach item $x' \in Y'$ do
13                 $\mathcal{L}_{x'} \leftarrow \mathcal{L}_{x'} - \{h(tid')\}$;
14             $\mathcal{L} \leftarrow \mathcal{L} \cup \{h(tid)\}$;
15             foreach item $x \in Y$ do
16                 $\mathcal{L}_x \leftarrow \mathcal{L}_x \cup \{h(tid)\}$;
```

whose hash value is equal to the Kth smallest value of $\mathcal{L}$) and then inserting the value of $h(T)$ (lines 9–16).

Space complexity. Let Δ be the maximum number of items in the union of transactions with same key ($\Delta \gg 1$). The K hash values maintained by Algorithm 1 correspond to K distinct keys. By using the orthogonal list storage, the worst space complexity of Algorithm 1 is $O(K \cdot \Delta)$.

Time complexity. The time complexity of Algorithm 1 is $O(|D| + K \cdot m \cdot \log K \cdot \log |D|)$, where m denotes the average number of items in each transaction[7].

4 KMV SYNOPSIS BASED ITEMSET FREQUENCY ESTIMATOR

In this section, we first discuss the drawbacks of existing KMV synopsis based frequency estimator (Section 4.1). To overcome these drawbacks, we propose a new frequency estimator based on KMV synopsis (Section 4.2). We demonstrate how the proposed estimator can be integrated into existing frequent itemset mining algorithms to mine frequent itemsets with guaranteed accuracy over multi-transaction streams (Section 4.3).

4.1 Drawbacks of the Existing Frequency Estimator

The estimators (Equation (2) and Equation (3)) have been employed to estimate the frequency of word co-occurrence patterns in Twitter stream [59, 60]. However, the following two drawbacks of the estimator in Equation (3) limits the use of KMV synopsis technique for frequency estimation on stream data in practice.

(i) The frequency estimator for an l-itemset ($l > 1$) in Equation (3) has a high estimation error. The error increases with l, the length of itemset.

(ii) The frequency estimator in Equation (3) does not satisfy the downward-closure (also called anti-monotonicity) property. This will lead to unreasonable results, e.g., the frequency of an itemset may be greater than that of its subset.

We proceed to give detailed analysis of the two drawbacks in the following subsections.

4.1.1 Estimation Error Analysis. Let $X = \{x_i | 1 \leq i \leq l\}$ $(l > 1)$ be an l-itemset, $fr\hat{e}q(X)$ be the frequency of X estimated by the estimator in Equation (3). According to [7], the **mean squared error** (**MSE**) of $fr\hat{e}q(X)$ is

$$MSE[fr\hat{e}q(X)] = \frac{|D_\cap|(k|D_\cup| - k^2 - |D_\cup| + k + |D_\cap|)}{k(k-2)}, \tag{4}$$

where k is the size of KMV synopsis constructed under Theorem 1 for union set $\cup_{i=1}^{l} D_{x_i}$. $|D_\cap|$ and $|D_\cup|$ denote the number of distinct keys in the intersection and union sets of l sub stream data $\{D_{x_i} | 1 \leq i \leq l\}$, respectively, and their quantities are independent of estimator and thus can be viewed as constant values. Then, the value of $MSE[fr\hat{e}q(X)]$ only depends on k, i.e., size of KMV synopsis for $\cup_{i=1}^{l} D_{x_i}$. Next, we introduce a corollary about $MSE[fr\hat{e}q(X)]$ as follows.

COROLLARY 1. *$MSE[fr\hat{e}q(X)]$ is monotonically decreasing with k.*

PROOF. Let $C_1 = |D_\cap|$ and $C_2 = |D_\cup|$ denote constant values. We can view $MSE[fr\hat{e}q(X)]$ as a function $f(k)$ of k $(k > 2)$, i.e., $f(k) = \frac{C_1(C_2k - k^2 - C_2 + k + C_1)}{k(k-2)}$. By extending the function on continuous real domain y $(y \in \mathcal{R}^+)$, we can get $f'(y) < 0$ $(\forall y, 2 < y \leq C_1 \leq C_2)$. This indicates that function $f(k)$ decreases monotonically with k. □

In addition, based on Theorem 1, we have $k = \min(k_{x_1}, k_{x_2}, \ldots, k_{x_l})$. Thus, the value of k would decrease with the increase of l (the length of X) [48]. According to Corollary 1, the frequency $freq(X)$ estimated by Equation (3) has a high MSE for l-itemsets when l is large. This will reduce the reliability of estimated frequency. In other words, the estimator in Equation (3) is not suitable for estimating frequency of l-itemsets $(l > 1)$.

4.1.2 Downward-Closure Analysis. We first introduce the definition of downward-closure property [27].

Definition 3. Downward-closure Property. The downward-closure property is satisfied if the following equation is true.

$$\begin{aligned} &\forall X' \subseteq X, freq(X') \geq freq(X) \text{ and} \\ &\forall X'' \supseteq X, freq(X'') \leq freq(X). \end{aligned}$$

This property is very important and is adopted by almost all **frequent itemset mining** (**FIM**) algorithms. Unfortunately, the following corollary shows that the existing estimator violates this property.

COROLLARY 2. *The estimator in Equation* (3) *is not downward-closure.*

PROOF. The proof is by contradiction.
Assumption: Consider two sub multi-transaction streams D_{x_1} and D_{x_2} for items x_1 and x_2, along with their KMV synopses $\mathcal{L}_{x_1}$ and $\mathcal{L}_{x_2}$ of sizes k_{x_1} and k_{x_2}, respectively. Suppose (1) $k_{x_1} = k_{x_2} = k^*$; (2) the first $(k^* - 1)$ smallest hash values of $\mathcal{L}_{x_1}$ and $\mathcal{L}_{x_2}$ are identical; (3) the k^*th smallest value of $\mathcal{L}_{x_1}$ (resp. $\mathcal{L}_{x_2}$) is $U_{(k^*)}^{x_1}$ (resp. $U_{(k^*)}^{x_2}$) and $U_{(k^*)}^{x_1} < U_{(k^*)}^{x_2}$.

First, according to Theorem 1, the KMV synopsis $\mathcal{L}_\cup = \mathcal{L}_{x_1} \oplus \mathcal{L}_{x_2}$ of union set $D_{x_1} \cup D_{x_2}$ has size $\min\{k_{x_1}, k_{x_2}\} = k^*$. The k^*th smallest value of $\mathcal{L}_\cup$ is $U_{(k^*)}^{x_1}$, and the size of intersection set $\mathcal{L}_{x_1} \cap \mathcal{L}_{x_2}$ is $K_\cap = (k^* - 1)$ under the assumption. Thus, according to Equation (3), the estimated frequency of the itemset $X = \{x_1, x_2\}$ is $fr\hat{e}q(X) = \frac{k^*-1}{k*} \times \frac{k^*-1}{U_{(k^*)}^{x_1}}$.

Second, according to Equation (2), the estimated frequencies of x_1 and x_2 are $\hat{freq}(x_1) = \frac{k^*-1}{U^{x_1}_{(k^*)}}$ and $\hat{freq}(x_2) = \frac{k^*-1}{U^{x_2}_{(k^*)}}$, respectively.

In order to ensure the estimator is downward-closure, both of $\hat{freq}(X) \leq \hat{freq}(x_1)$ and $\hat{freq}(X) \leq \hat{freq}(x_2)$ should be always satisfied according to Definition 3. It is easy to get that $\hat{freq}(X) < \hat{freq}(x_1)$. Thus, $\hat{freq}(X) \leq \hat{freq}(x_2)$ should be always true, i.e., $U^{x_2}_{(k^*)} \leq U^{x_1}_{(k^*)} \times \frac{k^*}{k^*-1}$. Considering that $U^{x_2}_{(k^*)} > U^{x_1}_{(k^*)}$ in our assumption, $U^{x_2}_{(k^*)}$ should fall into a tight interval $(U^{x_1}_{(k^*)}, U^{x_1}_{(k^*)} \times \frac{k^*}{k^*-1}]$ to guarantee the downward-closure property.

However, according to Definition 1, the hash values in the KMV synopsis follow a uniform distribution over $[0, 1]$. Since $U^{x_2}_{(k^*)}$ is derived from these hash values, it should randomly and uniformly fall into the interval $(U^{x_1}_{(k^*)}, 1]$ rather than the tight interval $(U^{x_1}_{(k^*)}, U^{x_1}_{(k^*)} \times \frac{k^*}{k^*-1}]$.

This is a contradiction. Thus, $\hat{freq}(X) \leq \hat{freq}(x_2)$ is not always true. According to Definition 3, the frequency estimated by Equation (3) is not downward-closure. □

In summary, the existing frequency estimator of Equation (3) suffers from high estimation error and is not downward-closure.

4.2 New Itemset Frequency Estimator

To overcome these drawbacks, one of the core challenges is how to construct KMV synopsis for union set. We propose a new method for constructing KMV synopsis of union set to reduce the estimation error in Section 4.2.1. We present a new frequency estimator based on the new KMV synopsis in Section 4.2.2, demonstrate that the new estimator is downward-closure in Section 4.2.3, and prove its error bound in Section 4.2.4.

4.2.1 New KMV Synopsis for Union Set. According to Corollary 1, to reduce the MSE, an intuitive idea is to increase k, i.e., the size of KMV synopsis of union set. Therefore, we next present a theorem which ensure that we can construct a larger KMV synopsis for union set compared with that by Theorem 1 [7].

Let $\mathcal{L}_{x_1}$ and $\mathcal{L}_{x_2}$ be the KMV synopses of two sub stream data D_{x_1} and D_{x_2}, k_{x_1} and k_{x_2} be their sizes, respectively, and $U_{\max} = \max(U_{(k_{x_1})}, U_{(k_{x_2})})$.

Theorem 2. *The set $\mathcal{L}_\cup = \{h(tid) | (T =< tid, Y >\in D_{x_1} \cup D_{x_2}) \wedge (h(tid) \leq U_{max})\}$ is a size k KMV synopsis of $D_{x_1} \cup D_{x_2}$, and the kth minimum value of $\mathcal{L}_\cup$ is $U_{(k)} = U_{\max}$, where $k = |\mathcal{L}_\cup|$.*

Proof. The proof is straightforward based on the definition of KMV synopsis. Let $S = \{h(tid) | T =< tid, Y >\in D_\cup\}$ consists of all the hash values of transaction identities appearing in $D_\cup$. As $D_\cup = D_{x_1} \cup D_{x_2}$, $U_{(k_{x_1})} \in S$ and $U_{(k_{x_2})} \in S$. Therefore, $U_{(k)} = U_{\max} \in S$ as well, and $\mathcal{L}_\cup \subseteq S$. According to the Definition 1 of KMV synopsis, $\mathcal{L}_\cup$ is precisely the size k KMV synopsis of $D_\cup$, and its kth minimum is just $U_{\max}$. □

Theorem 2 can be extended to union of multiple data sets. The number of distinct keys in union set $D_{x_1} \cup D_{x_2} \cup \ldots \cup D_{x_l}$ can be estimated with its KMV synopsis $\mathcal{L}_\cup$ of size $k = |\mathcal{L}_\cup|$. It is easy to find that k is generally larger than $\min(k_{x_1}, k_{x_2}, \ldots, k_{x_l})$ in Theorem 1 since $k = |\mathcal{L}_\cup| \geq |\cup_{i=1}^{l} \mathcal{L}_{x_i}| \geq \min(k_{x_1}, k_{x_2}, \ldots, k_{x_l})$.

With Theorem 2, we can further estimate the frequency of an l-itemset $X = \{x_i | 1 \leq i \leq l\}$ by $\hat{freq}(x) = |\hat{D}_\cap| = \frac{K_\cap}{k} \times |\hat{D}_\cup|$. Different from the existing estimator, the size of new KMV synopsis for union set is $k = |\mathcal{L}_\cup|$, which is generally much larger than $\min(k_{x_1}, k_{x_2}, \ldots, k_{x_l})$ in Theorem 1 for the existing estimator [7]. Moreover, with the increase of l, the size of KMV synopsis constructed by Theorem 2 increases. This will significantly reduce the estimation error for long itemsets.

Note that constructing the KMV synopsis of union set by Theorem 2 is very expensive in practice. Fortunately, in the context of itemset frequency estimation, we do not need to maintain the KMV synopsis for an l-itemset ($l > 1$). This greatly simplifies the estimation process. See details in the next section.

4.2.2 New Estimator. For a given l-itemset $X = \{x_i | 1 \leq i \leq l\}$, with Theorem 2, we can construct a KMV synopsis $\mathcal{L}_\cup$ for union set $\cup_{i=1}^{l} D_{x_i}$, which has a large size (i.e., k). Then, we can safely employ the *Maximum Likelihood Estimator* (Eq.5) to estimate the size of $D_\cup$ instead of Equation (2) since these two estimators become indistinguishable when k is large (i.e., $(k-1) \approx k$) [7].

$$|\hat{D_\cup}|^{MLE} = \frac{k}{U_{(k)}}, \tag{5}$$

where $U_{(k)}$ is the kth minimum hash value of $\mathcal{L}_\cup$ constructed under Theorem 2. $k = |\mathcal{L}_\cup|$, the computation of $U_{(k)}$ is straightforward, i.e., $U_{(k)} = \max(U_{(k_{x_1})}, U_{(k_{x_2})}, \ldots, U_{(k_{x_l})})$. In our article, this maximum likelihood estimator is used for single item frequency estimation.

Moreover, with the maximum likelihood estimator $\frac{k}{U_{(k)}}$, we can further simplify the frequency estimation for l-itemset ($l > 1$) X as Equation (6).

$$|\hat{freq}(X)| = \frac{K_\cap}{k} \times \frac{k}{U_{(k)}} = \frac{K_\cap}{U_{(k)}}, \tag{6}$$

where $K_\cap = \cap_{i=1}^{l} \mathcal{L}_{x_i}$ is the same as that used in Equation (3), and $U_{(k)} = \max(U_{(k_{x_1})}, U_{(k_{x_2})}, \ldots, U_{(k_{x_l})})$. Note that, to estimate the frequency of l-itemset $l > 1$, the new estimator does not need to compute the size of KMV synopsis of the union set. This greatly simplifies the process of frequency estimation.

Space Complex. The space complexity of our new estimator and the existing estimator [7] for a given l-itemset X ($l > 1$) is dominated by the space complexity of Algorithm 1. Let $\mathcal{L}_{x_i}$ be the inverted KMV synopsis for each single item $x_i \in X$ maintained in memory by Algorithm 1. (1) For the new estimator, its space cost of KMV synopsis for X is $|\cup_{i=1}^{l} \mathcal{L}_{x_i}| \leq K$ (See Theorem 2). (2) The space cost to construct the KMV synopsis of X by the existing estimator is $\min(k_{x_1}, k_{x_2}, \ldots, k_{x_l}) \leq K$ (See Theorem 1 [7]). Therefore, the space cost of both estimators is negligible comparing to the space complexity $O(K \cdot \Delta)$ of Algorithm 1 since $\Delta \gg 1$ in practice.

Time complexity. Comparing to Equation (3) [59, 60], the new estimator in Equation (6) is more efficient to compute, since it only makes $O(l)$ comparisons to calculate $U_{(k)}$. But existing estimator needs to not only estimate the size of union set by $O(l)$ comparisons but also calculate the union set of l inverted KMV sketches of the items in X with complexity $O(l \cdot q)$, where q represents the average size of inverted KMV sketch for each item of X.

4.2.3 Downward-Closure Property Analysis.

Corollary 3. *The estimator of Equation (6) is downward-closure under Theorem 2.*

Proof. Let $X = \{x_i | 1 \leq i \leq l\}$ be an l-itemset ($l > 1$), and $\mathcal{L}_{x_i}$ be the KMV synopsis of D_{x_i} ($1 \leq i \leq l$), $X' = \{x'_i | 1 \leq i \leq l'\} \subseteq X$, $l' \leq l$, $K_\cap$ and $K'_\cap$ be the sizes of $\cap_{i=1}^{l} \mathcal{L}_{x_i}$ and $\cap_{i=1}^{l'} \mathcal{L}_{x'_i}$, respectively. As $X' \subseteq X$, $K_\cap \leq K'_\cap$.

According to Theorem 2, $U_{(k)} = \max(U_{(k_{x_1})}, U_{(k_{x_2})}, \ldots, U_{(k_{x_l})})$ and $U_{(k')} = \max(U_{(k_{x'_1})}, U_{(k_{x'_2})}, \ldots, U_{(k_{x'_l})})$. As $X' \subseteq X$, we can get that $U_{(k)} \geq U_{(k')}$, and so $\frac{1}{U_{(k)}} \leq \frac{1}{U_{(k')}}$.

Considering $K_\cap \leq K'_\cap$ and $\frac{1}{U_{(k)}} \leq \frac{1}{U_{(k')}}$, $\frac{K_\cap}{U_{(k)}} \leq \frac{K'_\cap}{U_{(k')}}$, i.e., $\hat{freq}(X) \leq \hat{freq}(X')$.

Analogously, let $X'' \supseteq X$. It is easy to get that $fre\hat{q}(X'') \leq fre\hat{q}(X)$ by estimator $\frac{K_\cap}{U_{(k)}}$. In summary, according to Definition 3, we can get that the new proposed estimator in Equation (6) is downward-closure. □

Corollary 3 enables us to incorporate the new estimator into any itemset mining algorithm utilizing the downward-closure property over KMV synopsis.

4.2.4 Error Bound Analysis of Estimated Frequency. Theorem 2 ensures a large-size KMV synopsis of union set is constructed for frequency estimation in Equation (6). $\frac{K_\cap}{U_{(k)}}$ of Equation (6) becomes indistinguishable from $\frac{K_\cap}{k} \times \frac{k-1}{U_{(k)}}$ as k is large (i.e., $k \approx (k-1)$).

Lemma 1 shows that $\frac{K_\cap}{k} \times \frac{k-1}{U_{(k)}}$ can be viewed as sum of a set of i.i.d. random variables. This allows us to bound the error of $\frac{K_\cap}{k} \times \frac{k-1}{U_{(k)}}$ by Chernoff inequality and further give an approximate error bound of the proposed estimator in Equation (6).

Lemma 1. $\frac{K_\cap}{k} \times \frac{k-1}{U_{(k)}}$ *is distributed as sum of a set of i.i.d. weighted Bernoulli random variables.*

Proof. (i) Let D_{x_1} and D_{x_2} be two sub streams with respect to items x_1 and x_2, along with their KMV synopses $\mathcal{L}_{x_1}$ $\mathcal{L}_{x_2}$, $K_\cap = |\mathcal{L}_{x_1} \cap \mathcal{L}_{x_2}|$ be the size of intersection set of their KMV synopses, $\mathcal{L}_\cup$ be a size-k KMV synopsis of union set $D_{x_1} \cup D_{x_2}$, and $U_{(k)}$ be the kth minimum hash value of $\mathcal{L}_\cup$, $D_\cup = \Phi(D_{x_1} \cup D_{x_2})$ and $D_\cap = \Phi(D_{x_1} \cap D_{x_2})$. Then, for each transaction $T \in D_{x_1} \cap D_{x_2}$, we can define an adjusted weight ω of T as follows:

$$\omega(T) = \begin{cases} \frac{k-1}{k}\frac{1}{U_{(k)}}, & \text{if } T \text{ is extracted by KMV synopsis } \mathcal{L}_\cup \\ 0, & \text{otherwise} \end{cases}$$

(ii) According to Definition 1, KMV synopsis consists of a set of hash values following uniform distribution [7]. This means that, $\forall T \in D_{x_1} \cap D_{x_2}$, the probability of T being extracted by $\mathcal{L}_\cup$ is $p(T) = \frac{K_\cap}{|D_\cap|}$. With the probability $p(T)$, the expected value of $\omega(T)$ is

$$\begin{aligned} E(\omega(T)) &= E(p(T) \times \tfrac{k-1}{k}\tfrac{1}{U_{(k)}}) + E((1-p(T)) \times 0) \\ &= E(\tfrac{K_\cap}{|D_\cap|} \times \tfrac{k-1}{k}\tfrac{1}{U_{(k)}}) + E((1-\tfrac{K_\cap}{|D_\cap|}) \times 0) \\ &= \tfrac{1}{|D_\cap|}E(\tfrac{K_\cap}{k} \times \tfrac{k-1}{U_{(k)}}) \end{aligned}$$

$E(\frac{K_\cap}{k} \times \frac{k-1}{U_{(k)}}) = |D_\cap|$ as $E(\frac{K_\cap}{k} \times \frac{k-1}{U_{(k)}}$ is an unbiased estimator of $|D_\cap|$[7]. Thus, $E(\omega(T)) = 1$. Therefore, $\omega(T)$ is a Bernoulli random variable with probability $\frac{K_\cap}{|D_\cap|}$, and $\omega(T) = 1$ is associated with a weight $\frac{k-1}{k}\frac{1}{U_{(k)}}$.

(iii) As an estimation of $|D_\cap|$, $\frac{K_\cap}{k} \times \frac{k-1}{U_{(k)}}$ can also be represented as $\sum_{T \in D_\cap} \omega(T)p(T)$. Therefore, $\frac{K_\cap}{k} \times \frac{k-1}{U_{(k)}}$ can be viewed as sum of $|D_\cap|$ i.i.d. random variables following weighted Bernoulli distribution. □

Given an l-itemset ($l > 1$) X, let $freq(X)$ be the true frequency of X, and $fre\tilde{q}(X)$ be an estimation of $freq(X)$ estimated by $\frac{K_\cap}{k} \times \frac{k-1}{U_{(k)}}$. According to Lemma 1, $fre\tilde{q}(X)$ is viewed as sum of a set of i.i.d. random variables. This enables us to bound the error of $fre\tilde{q}(X)$ by Chernoff bounds.

(i) Lower tail: $\forall \delta \in (0, 1)$

$$Pr(fre\tilde{q}(X) \leq (1-\delta)freq(X)) \leq \exp^{-\frac{\delta^2 freq(X)}{2}}. \tag{7}$$

(ii) Upper tail: $\forall\delta > 0$,

$$Pr(\tilde{freq}(X) \geq (1+\delta)freq(X)) \leq \exp^{-\frac{\delta^2 freq(X)}{3}}. \tag{8}$$

The lower tail is to establish a lower bound of an estimated frequency, i.e., the probability of the estimated frequency $\tilde{freq}(X)$ being smaller than $(1-\delta)freq(X))$ $(0 < \delta < 1)$ is less than $\exp^{-\frac{\delta^2 freq(X)}{2}}$.

The upper tail is to establish an upper bound of an estimated frequency, i.e., the probability of the estimated frequency $\tilde{freq}(X)$ being greater than $(1+\delta)freq(X))$ $(\delta > 0)$ is less than $\exp^{-\frac{\delta^2 freq(X)}{3}}$.

Let $\hat{freq}(X)$ be an estimation of $freq(X)$ via Equation (6). As $\hat{freq}(X) \approx \tilde{freq}(X)$ under Theorem 2, we can use the error bounds in Equation (7) and Equation (8) as the approximate error bounds of $\hat{freq}(X)$. The following section shows that these bounds are very effective for accuracy analysis of FIM over multi-transaction stream.

4.3 Application for ϵ-Close Frequent Itemset Mining on Multi-Transaction Stream

In this subsection, we first introduce the problem of ϵ-close FIM [7, 9, 12, 46, 47], which is one of the most popular definitions of FIM on streams. Then, we present how to integrate the proposed estimator into a FIM algorithm [28] to solve the ϵ-close FIM problem over multi-transaction streams. Finally, we establish the relationship between the accuracy of FIM and the size of KMV synopsis, and further give the guideline to set the size KMV synopsis to achieve a guaranteed accuracy of FIM.

Problem Definition. The problem of ϵ-close frequent itemset mining [7, 9, 12, 46, 47] over multi-transaction streams is defined as follows.

Definition 4. ϵ-close Frequent Itemset Mining. For a given multi-transaction stream data D, a support threshold θ and a close parameter ϵ $(0 < \epsilon < 1)$, ϵ-close FIM is to find a collection of itemsets that includes all θ-frequent itemsets and does not include any itemset whose frequency is less than $\theta(1-\epsilon)|\Phi(D)|$, while $|\Phi(D)|$ is the number of distinct transaction keys in D.

According to Definition 4, ϵ-close FIM loosens the restriction of support threshold θ and may report some itemsets whose frequency falls into the interval $[\theta(1-\epsilon)|\Phi(D)|, \theta|\Phi(D)|)$ as results.

4.3.1 KMV Synopsis Based FP-Growth. We first compute the KMV synopsis of a multi-transaction stream. We integrate the estimator of Equation (5) for a singleton item for constructing FP-Tree and integrate the estimator of Equation (6) into the process of FIM. Note that the FP-Tree over a multi-transaction stream is constructed on a new set of transactions where each transaction is obtained by merging all multiple transactions with the same key. And the merging is done during the process of KMV synopsis extraction.

The algorithm consists of three parts: (i) KMV synopsis extraction over entire stream, (ii) FP-Tree construction over KMV synopsis, and (iii) FIM by frequent itemset growth over FP-Tree. The algorithm takes a stream data D, size of KMV synopsis K, support threshold θ, and closeness parameter ϵ as inputs. For the first part (line 1), we invoke function `KMV_Synopsis_Extraction()` on D to maintain a KMV synopsis with size K along with Inverted KMV Sketch $\mathcal{L}_x$ for each item x. For the second part (lines 2–8), we first estimate the frequency by Equation (5) for each item x appearing in $\hat{\mathbb{D}}$, and filter the items whose estimated frequency is less than $(1-\epsilon/2)\theta|\Phi(D)|$. Noted that we use $(1-\epsilon/2)\theta|\Phi(D)|$ to filter items rather than $(1-\epsilon)\theta|\Phi(D)|$ in definition of ϵ-close FIM. The reason of this trick is that it is convenient for us to establish an accuracy analysis for FIM w.r.t. K in Corollary 4. Then, we sort the frequent items in a decreasing order of their estimated frequencies, and construct a **header table** (HT) to store these sorted frequent items. Finally, we invoke function `KMVFPtree_Build()` to construct *KMVFP-Tree* over $\hat{\mathbb{D}}$ based on HT. For the third part (lines 9–10), we report the ϵ-close frequent itemsets $\mathcal{FI}$ on *KMVFP-Tree* by invoking function `KMV_FP_Growth()`, in which the itemsets are generated by executing `KMV_FP_Growth()` recursively. Once an itemset is generated (lines 3 – 8), we estimate its frequency by Equation (6) and report the itemset as a result if the estimated frequency is greater than $(1 - \epsilon/2)\theta|\Phi(D)|$ (lines 4–5 and 9–10).

ALGORITHM 2: KMV Synopsis-Based FP-Growth

```
   Inputs : A multi-transaction stream data $D$, size of KMV synopsis $K$, support threshold $\theta$, closeness
            parameter $\epsilon$;
   Output : Reported frequent itemsets $\mathcal{FI}$;
   //Part 1: KMV synopsis extraction
1  $[\mathcal{L}, \mathcal{L}_x, \hat{\mathbb{D}}] \leftarrow$ KMV_Synopsis_Extraction $(D, K)$//See Algorithm 1, $\hat{\mathbb{D}} \leftarrow$ merging the
     transactions with same key in $\mathbb{D}$, and $|\hat{\mathbb{D}}| = K$
   //Part 2: FP-Tree construction
2  foreach item $x$ appearing in $\mathcal{L}$ do
3      $\hat{freq}(x)$ is estimated over $\mathcal{L}_x$ by Equation (5);
4      if $\hat{freq}(x) < (1 - \epsilon/2)\theta|\Phi(D)|$ then
5          $\mathcal{S} \leftarrow \mathcal{S} - \{< x, k_x, U_{(k_x)} >\}$
6  Sort items of $\mathcal{S}$ in a decreasing order by their estimated frequencies;
7  Create a header table $HT$ according to the sorted items, add $U_{(k_{x_i})}$ into the entry corresponding to $x_i$ of
     $HT$;
8  $KFP\_Tree \leftarrow$ KMVFPtree_Build $(\hat{\mathbb{D}}, HT)$;
   //Part 3: $\epsilon$-close FIM by FP-Growth
9  $\mathcal{FI} \leftarrow$ KMV_FP_Growth($KFP\_Tree, \phi, \theta, \epsilon$);
10 return $\mathcal{FI}$;

   Function: KMV_FP_Growth($Tree, X, \theta, \epsilon$)
1  if Tree contains single path $P$ then
2      foreach Combination (denoted as $\beta$) of items in $P$ do
3          $X'' \leftarrow \beta \cup X$ whose $freq(X'')$ is estimated by Equation (6);
4          if $\hat{freq}(X'') \geq (1 - \epsilon/2)\theta|\Phi(D)|$ then
5              Report $X''$ as a frequent itemst;
6  else
7      foreach $x_i$ in header table of Tree do
8          $X'' \leftarrow X \cup \{x_i\}$ whose $freq(X'')$ is estimated by Equation (6);
9          if $\hat{freq}(X'') \geq (1 - \epsilon/2)\theta|\Phi(D)|$ then
10             Report $X''$ as a frequent itemst;
11         Construct $X''$'s conditional FP-Tree $Tree_{X''}$;
12         if $Tree_{X''} \neq \phi$ then
13             KMV_FP_Growth($Tree_{X''}$ , $X''$, $\theta$, $\epsilon$);
```

Time complexity. The time complexity of the first part of KMV synopsis extraction is $O(|D| + K \cdot m \cdot \log K \cdot \log |D|)$, where m is the average number of items in each transaction. The time complexity of the second part of FP-Tree construction is $O(|\hat{\mathbb{D}}| \cdot \Delta) = O(K \cdot \Delta)$. The time complexity of the third part of ϵ-close FIM is $O(\mathbb{F}(\textit{KMVFP-Tree}, \theta, \epsilon))$, where $\mathbb{F}(\textit{KMVFP-Tree}, \theta, \epsilon)$ denotes a function of frequent pattern tree *KMVFP-Tree*, support threshold θ and close parameter ϵ. It increases with (1) the increase of the size (i.e., height and width) of *KMVFP-Tree*, (2) the decrease of θ, and (3) increases of ϵ. In summary, the time complexity of Algorithm 2 is $O(|D| + K \cdot m \cdot \log K \cdot \log |D|) + O(K \cdot \Delta) + O(\mathbb{F}(\textit{KMVFP-Tree}, \theta, \epsilon))$.

4.3.2 Accuracy Analysis of FIM. The accuracy of KMV synopsis based ϵ-FIM depends on the size of KMV synopsis (i.e., K). Generally, the larger the KMV synopsis is, the more accurate but less efficient it is. We establish the quantitative relationship between the size of KMV synopsis and the accuracy for solving ϵ-close FIM in the proposed algorithm as follows.

Corollary 4. *Let θ, ϵ, and η be the given support threshold, closeness, and failure parameters, respectively. If the size of KMV synopsis K is set as*

$$K \geq \frac{24}{\epsilon^2}\left(\Delta + \log\frac{5\eta}{(1-\epsilon)\theta} + 5\right) + 1, \tag{9}$$

then the probability for the algorithm proposed in Section 4.3.1 to successfully solve the ϵ-close FIM problem over multi-transaction stream is at least (1 - $\frac{4}{5\eta}$).

Where Δ is the maximum number of distinct items in the set of transactions with the same key in D.

Proof. According to Definition 4, there are two situations in Algorithm 2 where ϵ-close FIM makes an error on an itemset: (i) a θ-frequent itemset X is not reported when its estimated frequency $fre\hat{q}(X) < (1-\epsilon/2)\theta|\Phi(D)|$; (ii) an infrequent itemset X with frequency smaller than $(1-\epsilon)\theta|\Phi(D)|$ is falsely reported as a result when its estimated frequency $fre\hat{q}(X) \geq (1-\epsilon/2)\theta|\Phi(D)|$. Next, we attempt to bound the size of KMV synopsis (i.e., K) such that both the two situations occur in a very low probability.

(1) Size of KMV Synopsis for Reporting All Frequent Itemsets. Let B_0 represent the event {Not reporting all θ-frequent itemsets}. We call an itemset X θ-frequent if $freq(X) \geq \theta|\Phi(D)|$. The probability that event B_0 occurs can be bounded by Lemma 2 (See Proof in Appendix A).

Lemma 2. *$Pr(B_0) \leq \frac{1}{5\eta}$ when $K \geq \frac{8}{\epsilon^2\theta}(\Delta + \log\frac{5\eta}{\theta}) + 1$.*

Where Δ is the maximum number of distinct items in the set of transactions with the same key in D.

(2) Size of KMV Synopsis for Rejecting Infrequent Itemsets. To reject all the itemsets whose frequencies fall into the interval $[1, (1-\epsilon)\theta|\Phi(D)|]$ with a high probability, we divide the interval $R = [1, (1-\epsilon)\theta|\Phi(D)|]$ into several mutual exclusive sub-intervals and bound the size of KMV synopsis for ϵ-close FIM on each sub-interval [9].

Let $\rho = (1-\epsilon)\theta$ and $R = [1, \rho|\Phi(D)|]$. We construct a set of L sub-intervals $\{R_1, R_2, \ldots, R_L\}$, where $R_i = (\rho|\Phi(D)|/2^i, \rho|\Phi(D)|/2^{i-1}]$ $(1 \leq i \leq L-1)$, $R_L = (1, \rho|\Phi(D)|/2^{L-1}]$ and $L = \lceil\log(|\Phi(D)|\rho)\rceil$. That is, $\cup_{i=1}^{L} R_i = R$ and $R_i \cap R_j = \phi$ $(1 \leq i \neq j \leq L)$. Then, the event A = {Reporting infrequent itemsets in R} can be further divided into L events A_i = {Reporting infrequent itemsets in R_i} $(1 \leq i \leq L)$ which are independent of each other.

Next, we divide these L events into three groups. (i) $B_1 = \{A_i | 3 < i \leq L\}$; (ii) $B_2 = \{A_2, A_3\}$; and (iii) $B_3 = \{A_1\}$. Then, the probability that each of $B1$, B_2, and B_3 happens can be bounded by Lemma 3 (See Proof in Appendix B), Lemma 4 (See Proof in Appendix C) and Lemma 5 (See Proof in Appendix D), respectively.

Lemma 3. *$Pr(B_1) \leq \frac{1}{5\eta}$ when $K \geq (\frac{2}{(1-\epsilon)\theta}(\Delta + \log\frac{5\eta}{(1-\epsilon)\theta} + 5) + 1)$.*

Lemma 4. *$Pr(B_2) \leq \frac{1}{5\eta}$ when $K \geq (\frac{24}{(1-\epsilon)\theta}(\Delta + \log\frac{5\eta}{(1-\epsilon)\theta} + 3) + 1)$.*

Lemma 5. *$Pr(B_3) \leq \frac{1}{5\eta}$ when $K \geq (\frac{24}{\epsilon^2(1-\epsilon)\theta}(\Delta + \log\frac{5\eta}{(1-\epsilon)\theta} + 1) + 1)$.*

Table 2. Statistics of Multi-Transaction Stream Data

Data	# transactions	# items	Max
T10I4D10000K	32,153,865	2,939	32
T15I6D20000K	62,747,246	3,833	41
Tweets [32]	64,268,838	54,835	38

In summary, let B represent the event "Algorithm 2 fails in solving the ϵ-close FIM problem". $B = \{B_0, B_1, B_2, B_3\}$. The size of KMV synopsis in Equation (9) is greater than the value of K bounded by Lemmas 2–5 with respect to each event B_i ($0 \le i \le 3$). Applying union bound, we can get the probability of event B happening as follows.

$$\begin{aligned} Pr(\text{KMV based FIM fails}) &\le \textstyle\sum_{i=0}^{3} Pr(B_i) \\ &\le \tfrac{1}{5\eta} + \tfrac{1}{5\eta} + \tfrac{1}{5\eta} + \tfrac{1}{5\eta} = \tfrac{4}{5\eta} \end{aligned}$$

Therefore, the probability that KMV synopsis based FIM (Algorithm 2) successfully solves an ϵ-close FIM problem is $1 - \frac{4}{5\eta}$. □

According to Corollary 4, the bounded size of KMV synopsis is independent of either the number of transactions in D or the number of distinct keys in D, and only depends on the maximum number of distinct items in the set of transactions with same key in D. This is a very useful characteristic for handling FIM problem on multi-transaction stream data since the number of transactions of the stream data is usually quite large and even infinite.

5 EXPERIMENTAL STUDY

We introduce the experimental setting in Section 5.1. We report the comparison between different itemset frequency estimators in Section 5.2, and present the experimental results of FIM in Section 5.3.

5.1 Experimental Setup

Data Sets. We conduct experiments on three stream data. Two synthetic retail multi-transaction streams named "T10I4D10000K" and "T15I6D20000K"[2] are generated by the IBM data generator[3] where each customer might generate multiple transactions. The third stream data is the real world massive multi-transaction tweet sentence stream with 64M transactions from 13.4M users. Table 2 shows the number of transactions, number of distinct items, and the maximum number of distinct items in the union of transactions with the same key.

Frequency Estimators. For l-itemset ($l > 1$), the estimator of Equation (3) is employed as a baseline to compare with our proposed estimator in Equation (6). We use KMVE_l (KMV synopsis based Estimator) to denote the existing estimators for l-itemset ($l > 1$), and eKMVE_l (enhanced KMV synopsis based Estimator) to denote the proposed estimators. Note that the existing sampling approaches such as reservoir sampling cannot be used on our datasets as discussed in Section 3.1.

FIM Algorithms. We compare the following four KMV synopsis based algorithms. (1) The KMV synopsis based FIM algorithm proposed in [59, 60] which integrates the existing estimator [7] into the Apriori [2] algorithm, which is denoted as $\mathsf{FIM}_{\mathsf{KA}}$. (2) We integrate the existing estimators

[2] The numbers following "T", "I" ,and "D" denote the average transaction size, the average large itemset size, and the number of customers, respectively. The maximum number of transactions for each customer is 5. And the key of each transaction is defined as the customer id.

[3] Available at http://www.philippe-fournier-viger.com/spmf/.

Table 3. Parameter Setting for FIM

Parameter (notation)	Settings	Default
Support threshold (θ)	0.1% to 1%	0.5%
Closeness parameter (ϵ)	0.05 to 0.1	0.1
Failure parameter (η)	8 to 40	8

[7, 59, 60] into FP-Growth algorithm [28], which is denoted by $\mathsf{FIM}_{\mathsf{KFP}}$. (3) We integrate the proposed estimators with Apriori algorithm [2], which is denoted by $\mathsf{FIM}_{\mathsf{eKA}}$. 4) The proposed estimators are integrated with FP-Growth algorithm [28], which is denoted by $\mathsf{FIM}_{\mathsf{eKFP}}$.

Note that as the existing estimator [7, 59, 60] is not downward closure, algorithms $\mathsf{FIM}_{\mathsf{KA}}$and $\mathsf{FIM}_{\mathsf{KFP}}$ do not have a stop condition. To avoid this, we assume that the downward closure property holds for them too.

Evaluation Metrics. To evaluate the effectiveness of our proposed estimator, we employ the metric ***Absolute Ratio Error (ARE)*** [7] defined as $\frac{|\hat{freq}(X)-freq(X)|}{freq(X)}$. To evaluate the effectiveness of FIM algorithms, the commonly used metrics *Precision* and *Recall* [27, 59, 60, 62] and *F1* = 2 $\frac{Precision \times Recall}{Precision + Recall}$ are employed. For efficiency, we report the runtime.

Parameter Settings. There are three important parameters for ϵ-close FIM: support threshold θ, closeness parameter ϵ, and failure parameter η. We analyze the effect of these parameters on performance of different FIM algorithms by varying one while fixing the others as default values. The setting of each parameter is shown in Table 3. For a given setting for the three parameters, we bound the size of KMV synopsis K by Corollary 4. To make a fair comparison, we set the same size of synopsis for all FIM algorithms.

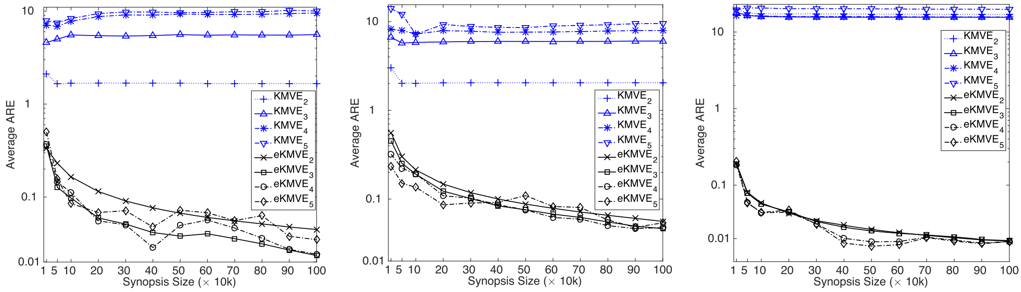
Objectives. We aim to empirically evaluate the following aspects: (i) The accuracy of the proposed estimator eKMVE_l by comparing with KMVE_l ($l > 1$); (ii) The accuracy and efficiency of the four FIM algorithms over streams; (iii) The effect of the lengths of streams on the performance of the four FIM algorithms over streams; and (iv) The effect of the size of KMV synopsis on the performance of our proposed algorithm $\mathsf{FIM}_{\mathsf{eKFP}}$.

All algorithms are implemented in Java on a workstation with Intel(R) Xenon(R) CPU E5-1620 v2 @3.7GHZ and 16G RAM. We use the **SIMD-oriented Fast Mersenne Twister** (**SFMT**) by following the work [59, 60] to simulate the hash function in Algorithm 1 for KMV synopsis extraction from a stream.

5.2 Evaluation on Estimators

For each stream, we first generate l-itemsets (l = 2, 3, 4, and 5) whose frequency is greater than 0.1%, and record the true frequency of each itemset. Afterwards, we estimate their frequency by using different estimators under different size of KMV synopsis extracted from the whole stream by Algorithm 1. We vary the size of KMV synopsis of the whole stream from 10,000 to 1,000,000. Figure 1 shows the comparison of different estimators in terms of ARE on three streams.

Estimation Error Comparison. Our proposed estimators (i.e., eKMVE_l, l = 2, 3, 4, and 5) outperform the respective existing ones (i.e., KMVE_l, l = 2, 3, 4, and 5) at least by an order of magnitude in terms of ARE on all the three streams. This is because, to estimate the frequency of an l-itemset X ($l > 1$), as discussed in Section 4.2, by using the same KMV synopsis extracted by Algorithm 1 on the original stream, our estimator can construct a much larger KMV synopsis of X comparing to the existing estimator. Furthermore, according to Section 4.1.1, a larger synopsis of X results in



(a) T10I4D10000K (b) T15I6D20000K (c) Twitter Stream

Fig. 1. Comparisons of different estimators in terms of ARE on three streams.

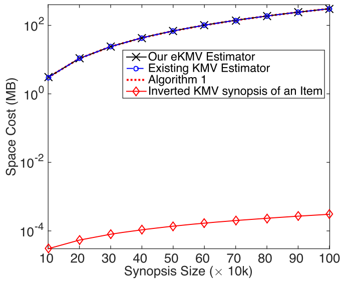
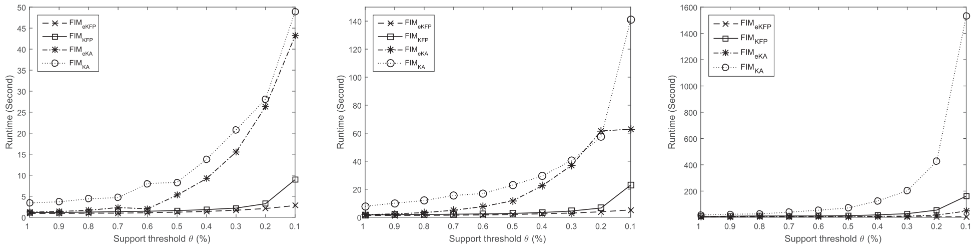


Fig. 2. Space cost comparisons over twitter stream.

a lower estimation error. Therefore, the estimation error of our proposed estimators is much lower than that of the existing ones.

Note that, according to the space complexity analysis in Section 4.2.2, the space cost of both estimators is dominated by Algorithm 1. The cost of the KMV synopsis for an l-itemset constructed by both estimators is negligible comparing to the space cost of Algorithm 1. Figure 2 shows the space cost of (1) the new eKMV estimator (the black line with ×), (2) the existing KMV estimator (the blue line with $\circ$), (3) Algorithm 1 (the dotted red line), and (4) the inverted KMV synopsis of a single item (the red line with $\diamond$) on Twitter stream. The results on the other two streams are qualitatively similar and thus omitted. We can observe that, the average space cost of the inverted KMV synopsis for a single item (the red line with $\diamond$) is much smaller than that of Algorithm 1. Both estimators construct the KMV synopsis for an l-itemset X ($1 < l \ll \Delta$) on l inverted KMV synopses of l single items in X. Therefore, the space cost of the KMV synopsis for X constructed by both estimators is also much smaller than the cost of Algorithm 1. That's why the lines w.r.t. both estimators overlap with the line w.r.t. Algorithm 1 in Figure 2. i.e., the dominant space cost of both estimators comes from Algorithm 1. This is consistent with space complexity analysis in Section 4.2.2.

Varying the Size of KMV Synopsis. With increasing the size of KMV synopsis of the whole stream, the estimation error ARE of our proposed estimators (i.e., eKMVE$_l$, l = 2, 3, 4, and 5) shows a declined tendency on all the three streams. This is because, the size of KMV synopsis constructed by our estimator for an l-itemset X increases when the size of KMV synopsis of the whole stream increases. And a larger KMV synopsis for X results in a lower estimation error according to Section 4.1.1. In contrast, the estimation error of the existing estimators (i.e., KMVE$_l$, l = 2, 3, 4, and 5) is not sensitive to the size of KMV synopsis K of the whole stream. That is, a larger KMV synopsis K does not mean a lower ARE for the existing estimators. This phenomenon seems counterintuitive at first glance. In fact, to estimate the frequency of an l-itmeset X, both estimators need



(a) T10I4D20000K (b) T15I6D20000K (c) Twitter Stream

Fig. 3. Runtime of FIM w.r.t. support threshold θ.

to construct the KMV synopsis for X based on l invert KMV synopses w.r.t. each single item of X. However, the KMV synopsis of X constructed by the existing estimator is much smaller than that constructed by ours according to Section 4.2.1. In this case, we can get that, *the estimation error ARE of the existing estimator is dominated by the ratio between the sizes of KMV synopsis of X constructed by our estimator and the existing estimator. And this ratio is independent of the size of KMV synopsis K of the original stream.* The detailed analysis of this phenomenon is given in Appendix E.

Varying Length of Itemset. For the l-itemsets ($l > 1$), with the increase of l (from 2 to 5), the estimation error ARE of our estimators generally decreases. For example, the ARE of eKMVE$_3$, eKMVE$_4$, and eKMVE$_5$ is generally smaller than that of eKMVE$_2$ over all the three streams. In contrast, the ARE of the existing estimators increases with l on the streams "T10I4D10000K" and "T15I6D20000K". This is because that, based on the same KMV synopsis extracted from the original stream by Algorithm 1, the size of KMV synopsis constructed by our estimator for the l-itemsets would increase with l. By comparison, the size of KMV synopsis constructed by the existing estimator for the same l-itemsets would decrease as l increases (according to Section 4.2.1). The estimation error of an l-itemset relies on the size of KMV synopsis constructed for the itemset. A larger synopsis results in a lower estimation error. Therefore, the estimation error ARE of our estimators generally decreases as l increases. Note that our estimator can construct a larger KMV synopsis for an l-itemset comparing to the existing estimator. However, their space complexities are the same since the space cost of both estimators is dominated by Algorithm 1 (See Figure 2 and the space complexity analysis in Section 4.2.2).

5.3 Results of Frequent Itemset Mining

5.3.1 Evaluation on Different Parameters. **Varying Support Threshold** θ We fix $\epsilon = 0.1$, $\eta = 8$, and vary θ from 0.1% to 1%. Figure 3 shows the runtime of different FIM algorithms over three streams. Tables 4, 5, and 6 shows the comparisons of different FIM algorithms in terms of Precision, Recall, and F1.

As shown in Figure 3, (i) with the decrease of θ, the runtime of all algorithms increases. (ii) FP-Growth based FIM algorithms are faster than Apriori based ones (i.e., FIM$_{KFP}$ > FIM$_{KA}$ and FIM$_{eKFP}$ > FIM$_{eKA}$) especially when the support threshold is small. (iii) Our proposed algorithms FIM$_{eKFP}$ is significantly faster than FIM$_{KFP}$ in the worst case of $\theta = 0.1\%$. This is because, many longer itemsets should be reported as results under such a lower support threshold. However, the previous estimator [7, 59, 60] usually makes an over-estimation for l-itemsets ($l > 1$), this results in that FIM$_{KFP}$ generates more false frequent itemsets and thus takes more time.

As shown in Tables 4 and 5, our proposed algorithms FIM$_{eKFP}$ and FIM$_{eKA}$ are slightly influenced by θ and usually achieve both high Precision and Recall comparing to FIM$_{KFP}$ and FIM$_{KA}$. With

Table 4. Varying θ over T10I4D10000K

$\theta(\%)$	$\mathsf{FIM}_{\mathsf{eKFP}}$			$\mathsf{FIM}_{\mathsf{KFP}}$			$\mathsf{FIM}_{\mathsf{eKA}}$			$\mathsf{FIM}_{\mathsf{KA}}$		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
1	0.99	0.97	0.98	0.87	1	0.93	0.98	0.93	0.95	0.97	0.93	0.95
0.9	0.96	0.97	0.96	0.86	1	0.92	0.99	0.95	0.97	0.98	0.95	0.96
0.8	0.99	0.97	0.98	0.89	1	0.94	0.99	0.96	0.97	0.97	0.96	0.96
0.7	0.96	0.98	0.97	0.86	1	0.92	0.97	0.98	0.97	0.92	0.98	0.95
0.6	0.97	0.99	0.98	0.83	1	0.91	0.97	0.98	0.97	0.9	0.98	0.94
0.5	0.97	0.97	0.97	0.83	0.99	0.90	0.96	0.96	0.96	0.86	0.97	0.91
0.4	0.98	0.99	0.98	0.82	1	0.90	0.98	0.97	0.97	0.83	0.97	0.89
0.3	0.98	0.98	0.98	0.78	1	0.88	0.97	0.98	0.97	0.79	0.98	0.87
0.2	0.98	0.98	0.98	0.81	1	0.90	0.96	0.98	0.97	0.79	0.99	0.88
0.1	0.98	0.99	0.98	0.84	0.99	0.91	0.97	0.96	0.96	0.86	0.96	0.91

⋆ "R": *Recall*, "P": *Precision.*

Table 5. Varying θ over T15I6D20000K

$\theta(\%)$	$\mathsf{FIM}_{\mathsf{eKFP}}$			$\mathsf{FIM}_{\mathsf{KFP}}$			$\mathsf{FIM}_{\mathsf{eKA}}$			$\mathsf{FIM}_{\mathsf{KA}}$		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
1	1	0.97	0.98	0.9	1	0.95	1	0.97	0.98	1	0.97	0.98
0.9	0.98	1	0.99	0.85	1	0.92	0.95	0.97	0.96	0.95	0.97	0.96
0.8	0.97	0.99	0.98	0.86	1	0.92	0.96	0.98	0.97	0.96	0.98	0.97
0.7	0.96	0.96	0.96	0.84	0.99	0.91	0.94	0.97	0.95	0.91	0.97	0.94
0.6	0.98	0.97	0.97	0.89	1	0.94	0.98	0.97	0.97	0.91	0.97	0.94
0.5	0.97	0.99	0.98	0.87	1	0.93	0.97	0.96	0.96	0.88	0.96	0.92
0.4	0.97	0.98	0.97	0.83	0.99	0.90	0.97	0.98	0.97	0.83	0.98	0.90
0.3	0.97	0.98	0.97	0.79	0.99	0.88	0.97	0.97	0.97	0.77	0.97	0.86
0.2	0.97	0.98	0.97	0.75	1	0.86	0.96	0.98	0.97	0.73	0.99	0.84
0.1	0.97	0.98	0.97	0.86	1	0.92	0.97	0.97	0.97	0.88	0.97	0.92

Table 6. Varying θ over Twitter Stream

$\theta(\%)$	$\mathsf{FIM}_{\mathsf{eKFP}}$			$\mathsf{FIM}_{\mathsf{KFP}}$			$\mathsf{FIM}_{\mathsf{eKA}}$			$\mathsf{FIM}_{\mathsf{KA}}$		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
1	0.98	0.99	0.98	0.43	1	0.60	0.97	0.99	0.98	0.22	1	0.36
0.9	0.98	1	0.99	0.45	1	0.62	0.98	1.00	0.99	0.21	1	0.35
0.8	0.99	1	0.99	0.45	1	0.62	0.99	1.00	0.99	0.25	1	0.40
0.7	0.99	0.99	0.99	0.44	1	0.61	0.97	0.99	0.98	0.22	1	0.36
0.6	1	0.99	0.99	0.45	1	0.62	0.98	0.99	0.98	0.22	1	0.36
0.5	0.99	0.99	0.99	0.60	0.99	0.75	0.99	0.99	0.99	0.3	0.99	0.46
0.4	0.99	0.99	0.99	0.54	0.995	0.70	0.98	0.99	0.98	0.28	0.99	0.44
0.3	0.98	0.99	0.98	0.43	1	0.60	0.98	0.99	0.98	0.23	0.99	0.37
0.2	0.98	0.96	0.97	0.41	1	0.58	0.84	0.98	0.90	0.23	0.99	0.37
0.1	0.94	0.97	0.95	0.60	0.99	0.75	0.98	0.99	0.98	0.29	0.99	0.45

the decease of θ, the performance of $\mathsf{FIM}_{\mathsf{KFP}}$ and $\mathsf{FIM}_{\mathsf{KA}}$ drops in general. This is because, more l-itemsets ($l > 1$) becomes frequent as θ decreases, and the existing estimator shows high estimation error and usually get a too great estimation for these itemsets. This leads to $\mathsf{FIM}_{\mathsf{KFP}}$ and $\mathsf{FIM}_{\mathsf{KA}}$ report more infrequent itemsets as results and thus have low Precision. For Table 6, there are many longer itemsets in Twitter stream even θ is large. $\mathsf{FIM}_{\mathsf{KFP}}$ and $\mathsf{FIM}_{\mathsf{KA}}$ usually report too many false frequent itemsets. This results in quite poor Precision and F1 comparing to our proposed algorithms $\mathsf{FIM}_{\mathsf{eKFP}}$ and $\mathsf{FIM}_{\mathsf{eKA}}$.

Varying Close Parameter. ϵ We vary ϵ from 0.05 to 0.1 while fixing $\theta = 0.5\%$, and$\eta = 8$. The Precision, Recall, and F1 of different FIM algorithms are shown in Tables 7, 8, and 9. We observe that the accuracy of different FIM algorithms usually increases with the decrease of ϵ. This can be

Table 7. Varying ϵ over T10I4D10000K

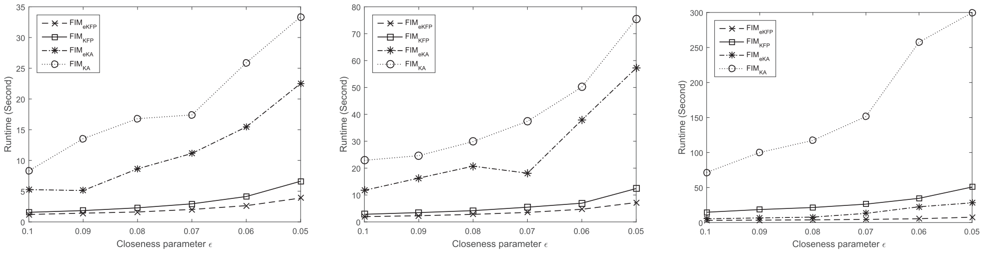
ϵ	FIM$_{\text{eKFP}}$			FIM$_{\text{KFP}}$			FIM$_{\text{eKA}}$			FIM$_{\text{KA}}$		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
0.1	0.97	0.97	0.97	0.83	0.99	0.90	0.96	0.96	0.96	0.86	0.97	0.91
0.09	0.97	0.97	0.97	0.84	0.99	0.91	0.97	0.96	0.96	0.87	0.97	0.92
0.08	0.99	0.97	0.98	0.85	1.00	0.92	0.98	0.97	0.97	0.88	0.97	0.92
0.07	0.98	0.98	0.98	0.86	1.00	0.92	0.98	0.97	0.97	0.88	0.97	0.92
0.06	0.99	0.98	0.98	0.87	1.00	0.93	0.99	0.97	0.98	0.89	0.97	0.93
0.05	0.99	0.99	0.99	0.88	1.00	0.94	0.99	0.97	0.98	0.89	0.97	0.93

Table 8. Varying ϵ over T15I6D20000K

ϵ	FIM$_{\text{eKFP}}$			FIM$_{\text{KFP}}$			FIM$_{\text{eKA}}$			FIM$_{\text{KA}}$		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
0.1	0.97	0.99	0.98	0.87	1	0.93	0.97	0.96	0.96	0.88	0.96	0.92
0.09	0.97	0.98	0.97	0.88	1	0.94	0.97	0.97	0.97	0.89	0.97	0.93
0.08	0.99	0.98	0.98	0.87	1	0.93	0.97	0.98	0.97	0.88	0.98	0.93
0.07	0.99	0.99	0.99	0.88	1	0.94	0.98	0.98	0.98	0.89	0.98	0.93
0.06	0.99	0.99	0.99	0.90	1	0.95	0.98	0.99	0.98	0.89	0.99	0.94
0.05	0.99	1	0.99	0.90	1	0.95	0.98	0.99	0.98	0.89	0.99	0.94

Table 9. Varying ϵ over Twitter Stream

ϵ	FIM$_{\text{eKFP}}$			FIM$_{\text{KFP}}$			FIM$_{\text{eKA}}$			FIM$_{\text{KA}}$		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
0.1	0.99	0.99	0.99	0.60	0.99	0.75	0.99	0.99	0.99	0.30	0.99	0.46
0.09	0.99	0.99	0.99	0.60	0.99	0.75	0.98	0.99	0.98	0.29	0.99	0.45
0.08	0.99	0.99	0.99	0.60	0.99	0.75	0.99	0.99	0.99	0.30	0.99	0.46
0.07	0.99	0.99	0.99	0.61	0.99	0.75	0.99	0.99	0.99	0.30	0.99	0.46
0.06	0.99	0.99	0.99	0.61	0.99	0.75	0.99	0.99	0.99	0.30	0.99	0.46
0.05	0.99	0.99	0.99	0.61	0.99	0.75	0.99	0.99	0.99	0.30	0.99	0.46



(a) T10I4D10000K (b) T15I6D20000K (c) Twitter Stream

Fig. 4. Runtime of FIM w.r.t. closeness parameter ϵ.

attributed to two reasons: First, a smaller ϵ results in a larger synopsis (according to Equation (9)), and a larger synopsis generally yields better accuracy for FIM. Second, a smaller ϵ means that the probability of reporting infrequent itemsets as results is low (according to Definition 4).

Tables 7–9 also show that our FIM algorithms FIM$_{eKFP}$ and FIM$_{eKA}$ achieve both high Recall and Precision, and is significant superior to FIM$_{\text{KFP}}$ and FIM$_{\text{KA}}$ in terms of F1. In contrast, algorithms FIM$_{\text{KFP}}$ and FIM$_{\text{KA}}$ that employ previous estimators [7, 59, 60] have low precision and F1 although they usually achieve high Recall. The precision of FIM$_{\text{KFP}}$ and FIM$_{\text{KA}}$ is particularly low on Twitter stream as shown in Table 9. This is due to the fact that many l-itemsets ($l > 1$) are discovered in Twitter stream and the estimators [7, 59, 60] tend to over-estimate the frequencies of long itemsets, and thus many infrequent itemsets are reported as results wrongly.

Table 10. Varying η over T10I4D10000K

η	FIM$_{\text{eKFP}}$			FIM$_{\text{KFP}}$			FIM$_{\text{eKA}}$			FIM$_{\text{KA}}$		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
8	0.97	0.97	0.97	0.83	0.99	0.90	0.96	0.96	0.96	0.86	0.97	0.91
16	0.97	0.97	0.97	0.84	0.99	0.91	0.96	0.97	0.96	0.86	0.97	0.91
24	0.97	0.97	0.97	0.84	0.99	0.91	0.97	0.96	0.96	0.86	0.96	0.91
32	0.97	0.97	0.97	0.83	0.99	0.90	0.97	0.96	0.96	0.86	0.96	0.91
40	0.97	0.97	0.97	0.86	1.00	0.92	0.99	0.98	0.98	0.57	0.99	0.72

Table 11. Varying η over T15I6D20000K

η	FIM$_{\text{eKFP}}$			FIM$_{\text{KFP}}$			FIM$_{\text{eKA}}$			FIM$_{\text{KA}}$		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
8	0.97	0.99	0.98	0.87	1	0.93	0.97	0.96	0.96	0.88	0.96	0.92
16	0.97	0.99	0.98	0.86	1	0.92	0.97	0.97	0.97	0.88	0.97	0.92
24	0.97	0.99	0.98	0.86	1	0.92	0.97	0.97	0.97	0.88	0.97	0.92
32	0.97	0.99	0.98	0.86	1	0.92	0.97	0.97	0.97	0.88	0.97	0.92
40	0.97	0.99	0.98	0.81	1	0.90	0.98	0.98	0.98	0.62	0.99	0.76

Table 12. Varying η over Twitter Stream

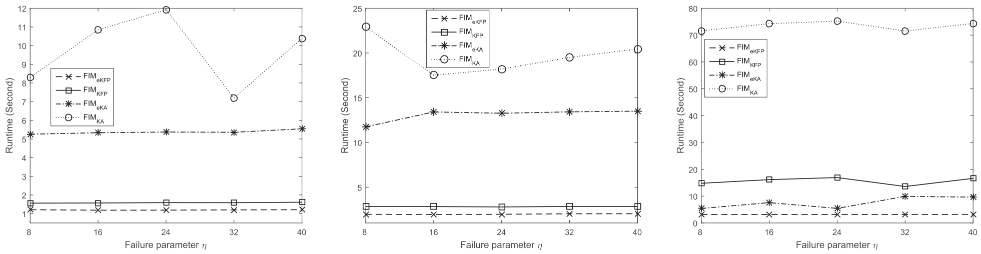
η	FIM$_{\text{eKFP}}$			FIM$_{\text{KFP}}$			FIM$_{\text{eKA}}$			FIM$_{\text{KA}}$		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
8	0.99	0.99	0.99	0.60	0.99	0.75	0.99	0.99	0.99	0.30	0.99	0.46
16	0.99	0.99	0.99	0.60	0.99	0.75	0.98	0.99	0.98	0.29	0.99	0.45
24	0.99	0.99	0.99	0.60	0.99	0.75	0.98	0.99	0.98	0.29	0.99	0.45
32	0.99	0.99	0.99	0.60	0.99	0.75	0.98	0.99	0.98	0.29	0.99	0.45
40	0.99	0.99	0.99	0.40	1	0.57	0.91	0.97	0.94	0.22	0.99	0.36

Figure 4 shows the runtime of all FIM algorithms increases with the decrease of ϵ. This is because smaller ϵ results in a larger synopsis for FIM, which need more time to process. As we observed in the last set of experiments, the FIM algorithms powered by our estimator outperform their counterparts using the previous estimator, i.e., FIM$_{\text{eKFP}}$ > FIM$_{\text{KFP}}$ and FIM$_{eKA}$ > FIM$_{\text{KA}}$. This is because on comparison to existing estimators, downward-closure property helps to stop itemset growth in advance and reduce the search space of itemsets, which is why this property can help us speedup both Apriori and FP-growth.

Additionally, we observe that the FP-Growth algorithm always runs faster than the Apriori based algorithms, irrespective of the estimators employed.

Varying Failure Parameter. η We vary η from 8 to 40 while fixing $\theta = 1\%$, $\epsilon = 0.1$. Tables 10, 11, and 12 report the Precision, Recall, and F1 of different FIM algorithms. We observe that the algorithms based on our estimator has both high precision and recall on all the three streams. For example, the Recall of FIM$_{\text{eKFP}}$ is at least 97% on T10I4D10000K and at least 99% on the other two streams and its precision is at least 97%. Although algorithms FIM$_{\text{KFP}}$ and FIM$_{\text{KA}}$ achieve high Recall as well, their Precision is much lower comparing to FIM$_{\text{eKFP}}$ and FIM$_{\text{eKA}}$. For example, the highest Precision of FIM$_{\text{KFP}}$ is 87% on T10I4D10000K, 88% of FIM$_{\text{KA}}$ on T15I6D20000K, and 60% on Twitter. This results in signifiant better F1 comparing to FIM$_{\text{eKFP}}$ and FIM$_{\text{eKA}}$.

Figure 5 compares the runtime of different FIM algorithms over three streams. As shown in Figure 5, the runtime of algorithms FIM$_{\text{eKFP}}$ and FIM$_{\text{eKA}}$, which use our proposed estimator, is almost not affected by θ over all the three streams. We can make similar observation for algorithms



(a) T10I4D10000K (b) T15I6D20000K (c) Twitter Stream

Fig. 5. Runtime of FIM w.r.t. failure parameter η.

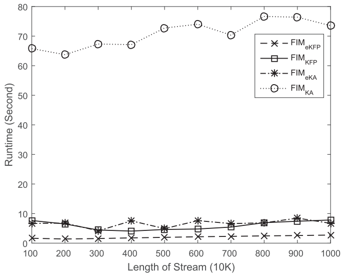


Fig. 6. Runtime of FIM w.r.t. stream length over twitter.

Table 13. Varying Stream Length over Twitter

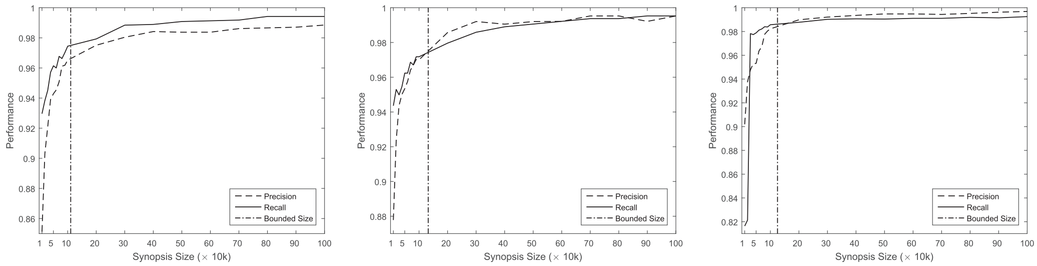
$\|D\|$ 10k	FIM$_{\text{eKFP}}$			FIM$_{\text{KFP}}$			FIM$_{\text{eKA}}$			FIM$_{\text{KA}}$		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
100	1	0.99	0.99	0.6	0.99	0.75	0.99	0.99	0.99	0.31	0.99	0.47
200	0.99	0.99	0.99	0.69	1	0.82	1	0.97	0.98	0.54	0.98	0.70
300	0.99	1	0.99	0.53	1	0.69	0.99	0.99	0.99	0.41	1.00	0.58
400	0.99	0.99	0.99	0.55	1	0.71	0.99	0.99	0.99	0.33	1.00	0.50
500	0.99	0.99	0.99	0.55	1	0.71	0.99	0.99	0.99	0.33	1.00	0.50
600	0.99	0.99	0.99	0.55	1	0.71	0.99	0.99	0.99	0.32	1.00	0.48
700	0.99	1	0.99	0.58	1	0.73	0.99	0.99	0.99	0.31	0.99	0.47
800	0.99	0.99	0.99	0.6	1	0.75	0.99	0.99	0.99	0.31	0.99	0.47
900	0.99	1	0.99	0.6	1	0.75	0.99	0.99	0.99	0.31	0.99	0.47
1000	0.99	0.99	0.99	0.56	1	0.72	0.99	0.99	0.99	0.31	1.00	0.47

FIM$_{\text{KFP}}$ and FIM$_{\text{eKA}}$. This is because the efficiency of these algorithms mainly depends on the size of KMV synopsis, and the failure parameter η has very little effect on the synopsis size.

Varying Length of Stream. This set of experiments is to evaluate the effect of the length of stream on different FIM algorithms. We vary the stream length from 100K to 1,000K while setting $\theta = 0.5\%$, $\epsilon = 0.1$, $\eta = 8$. The experimental results on the three streams are qualitatively similar and we only report the result on the *Twitter* stream due to the space limitation. Table 13 shows that FIM$_{\text{eKFP}}$ and FIM$_{\text{eKA}}$ achieve both high Recall and Precision (99% at least) for different length of streams, and the results are consistent over the different lengths. In contrast, the highest Precision of FIM$_{\text{KFP}}$ (resp. FIM$_{\text{KA}}$) is only 69% (resp. 54%), this causes their F1 values drops significantly. The runtime is shown in Figure 6. Similar to the results in previous experiments, the FIM algorithms using our new estimators always run faster than the counterparts using the previous estimators. The observation holds for different lengths of streams.

Table 14. Varying Number of Distinct Items over T10I4D10000K

# items	FIM_{eKFP}			FIM_{KFP}			FIM_{eKA}			FIM_{KA}		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
500	1	0.99	0.99	0.91	1	0.95	0.99	0.99	0.99	0.93	0.97	0.95
1,000	1	0.99	0.99	0.91	1	0.95	1	0.99	0.99	0.92	0.97	0.94
1,500	1	0.99	0.99	0.90	1	0.95	0.99	0.99	0.99	0.9	0.97	0.93
2,000	0.99	0.99	0.99	0.90	1	0.95	0.99	0.97	0.98	0.9	0.97	0.93



(a) T10I4D10000K (b) T15I6D20000K (c) Twitter Stream

Fig. 7. Impact of size of KMV synopsis on proposed FIM algorithm FIM_{eKFP} over three stream data.

Varying Number of Distinct Items. This set of experiments is to evaluate the effect of number of distinct items appearing in stream on different FIM algorithms. We vary the number of distinct items from 500 to 2,000 based on "T10I4D10000K"[4], while setting $\theta = 0.5\%$, $\epsilon = 0.1$, $\eta = 8$. Table 14 shows that the algorithms FIM_{eKFP} and FIM_{eKA} based on our estimator have both high precision and recall. The results are consistent over the different distinct item numbers. In comparison, although algorithms FIM_{KFP} and FIM_{KA} achieve high recall, their precision and F1 are signifiant lower and further drop when then number of distinct items increases.

5.3.2 Evaluation of Size of KMV Synopsis for FIM. Recall we establish the relationship between the size of KMV synopsis and the accuracy of the FIM algorithms using our new estimators in Section 4.3.2, and Corollary 4 is proposed to guide the setting for the size of KMV synopsis with a theoretical bound on accuracy. We vary the size of KMV synopsis from 10K to 1,000K, and run FIM_{eKFP} under the default settings of $\theta = 0.5\%$ and $\epsilon = 0.1$. Note that the similar results are from FIM_{eKA}. Recall we establish the relationship between the size of KMV synopsis and the accuracy of the FIM algorithms using our new estimators in Section 4.3.2, and Corollary 4 is proposed to guide the setting for the size of KMV synopsis with a theoretical bound on accuracy. We vary the size of KMV synopsis from 10K to 1,000K, and run FIM_{eKFP} under the default settings of $\theta = 0.5\%$ and $\epsilon = 0.1$. Note that the similar results are from FIM_{eKA}.

Figure 7 shows the *Recall* and *Precision* of FIM_{eKFP} over the three streams under different sizes of KMV synopsis. The vertical dotted line in Figure 7 indicates the size of KMV synopsis by following the bound established by Corollary 4 under default parameter setting for FIM. We make the following observation: (i) As the size of KMV synopsis increases, the performance of FIM_{eKFP} first increases rapidly and then grows slowly in terms of both *Precision* and *Recall* over all the three streams. (ii) The size of KMV synopsis by following the bound in Corollary 4 results in at least 96% *Precision* and *Recall.* This indicates that Corollary 4 is effective in bounding the size of KMV

[4] For example, to generate data stream with 500 distinct items, we first sampled 500 distinct items from "T10I4D10000K", and then replaced the items beyond these sampled 500 ones in each transaction by randomly selected items from 500 distinct items.

synopsis for FIM. (iii) Although the length of these stream data varies significantly, the size of KMV synopsis bounded by Corollary 4 is only hundreds of thousands, which is a very small fraction of the original stream. Specifically, the fraction of the KMV synopsis to the original stream is only 0.34% for "T10I4D10000K", 0.21% for "T15I620000K", and 0.19% for Twitter stream, respectively. The bounded size of KMV synopsis by Corollary 4 is independent of the length of the original stream, whose size may be infinite. This is very useful in practice.

6 CONCLUSIONS

In this article, we focus on a new format of streaming data (i.e., multi-transaction stream) where multi-transactions are associated with the same key and attempt to estimate the frequency of itemsets over massive such stream. We propose a new KMV-based method for itemset frequency estimation and apply it to the problem of FIM over multi-transaction stream. We first theoretically showed that the existing KMV-based estimator has low accuracy even for the short itemsets and does not satisfy the important downward-closure property. Then we put forward an enhanced KMV-based frequency estimator to address the above two issues. Afterwards, we integrated the new estimator into the FP-Growth algorithm to solve the well-known problem of ϵ-close frequent itemset mining in stream analysis. We also provided a comprehensive theoretical analysis on the estimator size required for a given the accuracy guarantee. Finally, we thoroughly evaluated the proposed new estimator and FIM algorithm over three popular stream datasets. The comprehensive experiments clearly showed the new estimator outperforms the existing work by a big margin, especially for long itemset. The new FIM algorithm is more efficient and meanwhile can significantly improve the precision of the results while keeping a competitive recall.

APPENDICES

A PROOF OF LEMMA 2

Proof. Let X be a θ-frequent itemset, and $\hat{freq}(X)$ be the frequency of X estimated by the proposed estimator. According to the lower tail of Equation (7), if we set $\delta = \epsilon/2$, the probability that X is not reported by Algorithm 2 is

$$\begin{aligned} Pr(X \text{ is not reported}) &= Pr(\hat{freq}(X) < (1-\epsilon/2)\theta|\Phi(D)|) \\ &\leq Pr(\hat{freq}(X) < (1-\epsilon/2)freq(X)) \\ &\leq \exp^{-\frac{\epsilon^2|\Phi(D)|\theta}{8}}. \end{aligned}$$

Meanwhile, $\forall\theta \in (0,1)$, the number of θ-frequent itemsets is at most $\frac{2^\Delta}{\theta}$ according to [9], where Δ is the maximum number of distinct items in the set of transactions with same identity in D. Then, by applying the union bound, we get,

$$Pr(B_0) \leq \frac{2^\Delta}{\theta}\exp^{-\frac{-\epsilon^2|\Phi(D)|\theta}{8}}.$$

Let $\frac{2^\Delta}{\theta}\exp^{-\frac{-\epsilon^2|\Phi(D)|\theta}{8}} \leq \frac{1}{5\eta}$. Replacing $|\Phi(D)|$ by $\frac{K-1}{U_{(K)}}$ estimated based on the KMV synopsis of D with size K, we get

$$U_{(K)} \leq \frac{(K-1)\epsilon^2\theta}{8(\log(5\eta/\theta) + \Delta)}.$$

To ensure this equation is true, i.e., $Pr(B_0) \leq \frac{1}{5\eta}$. As $U_{(K)} \in (0,1)$, we can set $\frac{(K-1)\epsilon^2\theta}{8(\log(5\eta/\theta)+\Delta)} \geq 1$. By solving this inequality, we get $K \geq \frac{8}{\epsilon^2\theta}(\Delta + \log\frac{5\eta}{\theta}) + 1$. □

B PROOF OF LEMMA 3

Proof. $B_1 = \{A_i | 3 < i \leq L\}$. We first give the probability that each even A_i happens. A_i is an event that the itemsets whose frequency falls into the region R_i $(i > 3)$ are falsely reported as results by Algorithm 2. Let X be an itemset with frequency falling into the interval R_i, that is, $freq(X) \in (\rho|\Phi(D)|/2^i, \rho|\Phi(D)|/2^{i-1}]$. According to Algorithm 2, X is falsely reported if $\hat{freq}(X) \geq (1 - \epsilon/2)\theta|\Phi(D)|$.

As $\frac{1-\epsilon/2}{1-\epsilon} \geq (1 + \epsilon/2)$, considering $\rho = (1 - \epsilon)\theta$, we get that $(1 - \epsilon/2)\theta|\Phi(D)| \geq (1 + \epsilon/2)\rho|\Phi(D)|$. Meanwhile, as $freq(X) \leq \rho|\Phi(D)|/2^{i-1}$, the probability of X being falsely reported is

$$\begin{aligned} Pr(X \text{ is reported}) &= Pr(\hat{freq}(X) \geq (1 - \epsilon/2)\theta|\Phi(D)|) \\ &\leq Pr(\hat{freq}(X) \geq (1 + \epsilon/2)\rho|\Phi(D)|) \\ &\leq Pr(\hat{freq}(X) \geq 2^{i-1}(1 + \epsilon/2)freq(X)). \end{aligned}$$

Let $\delta = 2^{i-1}(1 + \epsilon/2) - 1$. By applying the general form of upper tail of Chernoff bound [4], i.e., $Pr(\hat{freq}(X) \geq (1 + \delta)freq(X)) \leq (\frac{\exp^\delta}{(1+\delta)^{(1+\delta)}})^{freq(X)}$, we get

$$\begin{aligned} Pr(X \text{ is reported}) &\leq (\tfrac{\exp^\delta}{(1+\delta)^{(1+\delta)}})^{freq(X)} \\ &\leq (\tfrac{\exp^\delta}{(1+\delta)^{(1+\delta)}})^{|\Phi(D)|\rho 2^{-i}} \text{Since } \tfrac{\exp^\delta}{(1+\delta)^{(1+\delta)}} < 1 \\ &\leq (\tfrac{\exp^1}{(1+\delta)})^{(1+\delta)|\Phi(D)|\rho 2^{-i}} \\ &= (\tfrac{\exp^1}{2^{i-1}(1+\epsilon/2)})^{(1+\epsilon/2)|\Phi(D)|\rho/2} \\ &\leq (\tfrac{1}{2^{i-2}(1+\epsilon/2)})^{(1+\epsilon/2)|\Phi(D)|\rho/2} \\ &\leq (\tfrac{1}{2^{i-3}})^{(1+\epsilon/2)|\Phi(D)|\rho/2} \\ &\leq 2^{-(i-3)|\Phi(D)|\rho/2}. \end{aligned}$$

By replacing $|\Phi(D)|$ as $\frac{K-1}{U_{(K)}}$ and recalling $U_{(K)} \leq 1$, then

$$Pr(X \text{ is reported}) \leq 2^{-(j-3)(K-1)\rho/2}$$

If we take the size of KMV synopsis as $K \geq \frac{2}{\rho}(\Delta + \log\frac{5\eta}{\rho} + 5) + 1$, we get

$$Pr(X \text{ is reported}) \leq 2^{-(j-3)(\Delta+\log\frac{5\eta}{\rho}+5)}.$$

Once we get the probability of making an erroneous report for a single itemset, as the number of itemsets in sub-interval $R_i = (\rho|\Phi(D)|/2^i, \rho|\Phi(D)|/2^{i-1}]$ is at most $\frac{2^\Delta}{\rho/2^i} = \frac{2^{\Delta+i}}{\rho}$ [9]. Then, applying the union bound, the probability of event A_i happening is

$$\begin{aligned} Pr(A_i) &\leq \tfrac{2^{\Delta+i}}{\rho} \cdot 2^{-(i-3)(\Delta+\log\frac{5\eta}{\rho}+5)} \\ &= \tfrac{1}{5\eta} 2^{-(i-4)(\Delta+\log\frac{5\eta}{\rho}+5)+i-5} \\ &\leq \tfrac{1}{5\eta} 2^{-7(i-4)+i-5} \text{As } \Delta \text{ and } \log\tfrac{5h}{\rho} \geq 1 \\ &= \tfrac{1}{5\eta} 2^{-6i+23} = \tfrac{1}{5\eta} 2^{(-5i+20)-i+3} \\ &\leq \tfrac{1}{5\eta} 2^{-(i-3)} \text{ since } i \geq 4. \end{aligned}$$

Now, we can get that if we set the size of KMV synopsis as $K \geq (\frac{2}{\rho}(\Delta + \log\frac{5\eta}{\rho} + 5) + 1)$, the probability of event A_i is bounded by $\frac{1}{5\eta}2^{-(i-3)}$ when $i > 3$, where $\rho = (1 - \epsilon)\theta$. Since event B_1 is a union of a set of independent events $\{A_i | 4 \leq i \leq L\}$, by applying the union bound, we can get that

$$Pr(B_1) = \sum_{i=4}^{L} Pr(A_i) \leq \frac{1}{5\eta} \sum_{i=4}^{L} 2^{-(i-3)} \leq \frac{1}{5\eta}. \qquad \square$$

C PROOF OF LEMMA 4

Proof. $B_2 = \{A_2, A_3\}$ corresponds to the itemsets whose frequency falls into the interval $(\rho|\Phi(D)|/8, \rho|\Phi(D)|/2]$. Let X be an itemset with frequency $\rho|\Phi(D)|/8 < freq(X) \leq \rho|\Phi(D)|/2$. According to Algorithm 2, X is error reported if $\hat{freq}(X) \geq (1-\epsilon/2)\theta|\Phi(D)|$. We can get that $\frac{\hat{freq}(X)}{freq(X)} \geq \frac{2(1-\epsilon/2)\theta}{\rho} = 2 + \frac{\epsilon}{1-\epsilon}$. Let $\delta = 1 + \frac{\epsilon}{1-\epsilon}$, according to the upper tail of Equation (8), we get that

$$\begin{aligned} Pr(X \text{ is reported}) &\leq \exp^{-\frac{\delta^2 freq(X)}{3}} \\ &\leq \exp^{-\frac{|\Phi(D)|\rho}{24}} \text{ As } \delta > 1 \wedge freq(X) > \rho/8. \end{aligned}$$

As the number of itemsets with frequency falling in the interval $(\rho|\Phi(D)|/8, \rho|\Phi(D)|/2]$ is at most $\frac{2^\Delta}{\rho/8}$ [9]. By applying the union bound, the probability of event B_2 occurring is

$$Pr(B_2) \leq \frac{2^{\Delta+3}}{\rho} \exp^{-\frac{|\Phi(D)|\rho}{24}}.$$

By replacing $|\Phi(D)|$ as $\frac{K-1}{U_{(K)}}$ and considering $U_{(K)} \leq 1$, we have

$$Pr(B_2) \leq \frac{2^{\Delta+3}}{\rho} \exp^{-\frac{(K-1)\rho}{24}}.$$

To ensure $Pr(B_2) \leq \frac{1}{5\eta}$, we get $K \geq \frac{24}{\rho}(\Delta + \log\frac{5\eta}{\rho} + 3) + 1$, where $\rho = (1-\epsilon)\theta$. □

D PROOF OF LEMMA 5

Proof. For event $B_3 = \{A_1\}$ corresponds to the itemsets whose frequency falls into the interval $(\rho|\Phi(D)|/2, \rho|\Phi(D)|]$, let X be an itemset whose frequency $\rho|\Phi(D)|/2 < freq(X) \leq \rho|\Phi(D)|$. According to Algorithm 2, X is error reported if $\hat{freq}(X) \geq (1-\epsilon/2)\theta|\Phi(D)|$. Then, we can get $\frac{\hat{freq}(X)}{freq(X)} \geq (1 + \frac{\epsilon}{2(1-\epsilon)})$. Let $\delta = \frac{\epsilon}{2(1-\epsilon)}$, according to the upper tail of Equation (8), we get that

$$\begin{aligned} Pr(X \text{ is reported}) &\leq \exp^{-\frac{\delta^2 freq(X)}{3}} \\ &\leq \exp^{-\frac{\epsilon^2|\Phi(D)|\rho}{24}} \text{ As } \delta > \epsilon/2 \wedge freq(X) > \frac{\rho|\Phi(D)|}{2}. \end{aligned}$$

As the number of itemsets with frequency falling in the interval $(\rho|\Phi(D)|/2, \rho|\Phi(D)|]$ is at most $\frac{2^\Delta}{\rho/2}$ [9]. By applying the union bound, we get the probability of event B_3 happening as follows:

$$Pr(B_3) \leq \frac{2^{\Delta+1}}{\rho} \exp^{-\frac{\epsilon^2|\Phi(D)|\rho}{24}}.$$

By replacing $|\Phi(D)|$ as its unbiased estimation $\frac{K-1}{U_{(K)}}$ and considering that $U_{(K)} \leq 1$, then

$$Pr(B_3) \leq \frac{2^{\Delta+1}}{\rho} \exp^{-\frac{(K-1)\epsilon^2\rho}{24}}.$$

To ensure $Pr(B_3) \leq \frac{1}{5\eta}$, we get the size of KMV synopsis should be $K \geq \frac{24}{\epsilon^2\rho}(\Delta + \log\frac{5\eta}{\rho} + 1) + 1$, where $\rho = (1-\epsilon)\theta$. □

E DEMONSTRATION FOR FIGURE 1

Firstly, let $X = \{x_i | 1 \leq i \leq l\}$ be an l-itemset ($l > 1$). We can estimate its frequency as $freq\hat{(}X)_{ex} = \frac{K_\cap}{k_{ex}} \times \frac{k_{ex}-1}{U_{(k_{ex})}}$ by the existing estimator in Equation (3) and as $freq\hat{(}X)_{our} = \frac{K_\cap}{U_{(k_{our})}}$ by our proposed estimator in Equation (6), where k_{ex} and k_{our} are the sizes of KMV synopses $\mathcal{L}_{\cup ex}$ and $\mathcal{L}_{\cup our}$ constructed for the union set $\cup_{i=1}^{l} D_{x_i}$ under Theorems 1 and 2, respectively. $k_{our} = |L_{\cup our}|$ and

$k_{ex} = \min\{|\mathcal{L}_{x_1}|, |\mathcal{L}_{x_1}|, \ldots, |\mathcal{L}_{x_l}|\}$. $U_{(k_{ex})}$ (resp. $U_{(k_{our})}$) is the k_{ex}th (resp. k_{our}th) smallest hash value of $\mathcal{L}_{\cup ex}$ (resp. $\mathcal{L}_{\cup our}$), respectively.

Secondly, Theorem 1 shows that $U_{(k_{ex})}$ is the k_{ex}th smallest value of $\cup_{i=1}^{l} \mathcal{L}_{x_i}$. Theorem 2 tells us that $U_{(k_{our})}$ is the maximum value of $\cup_{i=1}^{l} \mathcal{L}_{x_i}$. Then, considering the hash values follow uniform distribution, we can get $U_{(k_{ex})} \approx \frac{k_{ex}}{k_{our}} \times U_{(k_{our})}$. Taking this approximation into $\widehat{freq}(X)_{ex}$, we get $\widehat{freq}(X)_{ex} \approx \frac{k_{our}}{k_{ex}} \times \frac{k_{ex}-1}{k_{ex}} \times \frac{K_{\cap}}{U_{(k_{our})}}$. i.e., $\widehat{freq}(X)_{ex} \approx \frac{k_{our}}{k_{ex}} \times \frac{k_{ex}-1}{k_{ex}} \times \widehat{freq}(X)_{our}$. When the size of KMV synopsis L is large, it is safe for us to assume $(k_{ex} - 1) \approx k_{ex}$. Then, $\widehat{freq}(X)_{ex} \approx \frac{k_{our}}{k_{ex}} \times \widehat{freq}(X)_{our}$. As k_{our} is generally much larger than k_{ex} on the three streams when $l > 1$, let $\alpha = \frac{k_{our}}{k_{ex}}$ and so $\alpha \gg 1$. According to the definition of ARE, ARE of $\widehat{freq}(X)_{ex}$ is $\frac{|\widehat{freq}(X)_{ex} - freq(X)|}{freq(X)}$ $\approx \frac{|\alpha \widehat{freq}(X)_{our} - freq(X)|}{freq(X)} = \frac{|(\alpha-1)\widehat{freq}(X)_{our} + \widehat{freq}(X)_{our} - freq(X)|}{freq(X)}$.

Thirdly, as $\alpha \gg 1$, $(\alpha - 1)$ plays a dominant role in ARE calculation for $\widehat{freq}(X)_{ex}$. Meanwhile, both of k_{our} and k_{ex} increase proportionately with the size of KMV synopsis. So $\alpha = \frac{k_{our}}{k_{ex}}$ remains almost flat when the size of KMV synopsis varies. Therefore, the estimation error ARE of $\widehat{freq}(X)_{ex}$, which is dominated by α, does not decrease when the size of KMV synopsis increases.

REFERENCES

[1] Charu C. Aggarwal and S. Yu Philip. 2007. A survey of synopsis construction in data streams. In *Data Streams*. Springer, 169–207.

[2] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, Vol. 1215. 487–499.

[3] Noga Alon, Yossi Matias, and Mario Szegedy. 1996. The space complexity of approximating the frequency moments. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*. Association for Computing Machinery, 20–29.

[4] Noga Alon and Joel H. Spencer. 2015. *The Probabilistic Method*. John Wiley & Sons.

[5] Arvind Arasu and Gurmeet Singh Manku. 2004. Approximate counts and quantiles over sliding windows. In *Proceedings of the 23rd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. Association for Computing Machinery, 286–296.

[6] Rodrigo Barbado, Oscar Araque, and Carlos A. Iglesias. 2019. A framework for fake review detection in online consumer electronics retailers. *Information Processing & Management* 56, 4 (2019), 1234–1244.

[7] Kevin Beyer, Peter J. Haas, Berthold Reinwald, Yannis Sismanis, and Rainer Gemulla. 2007. On synopses for distinct-value estimation under multiset operations. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*. Association for Computing Machinery, 199–210.

[8] Luís Cavique. 2007. A scalable algorithm for the market basket analysis. *Journal of Retailing and Consumer Services* 14, 6 (2007), 400–407.

[9] Venkatesan T. Chakaravarthy, Vinayaka Pandit, and Yogish Sabharwal. 2009. Analysis of sampling techniques for association rule mining. In *Proceedings of the 12th International Conference on Database Theory*. Association for Computing Machinery, 276–283.

[10] Joong Hyuk Chang and Won Suk Lee. 2003. estWin: Adaptively monitoring the recent change of frequent itemsets over online data streams. In *Proceedings of the 12th International Conference on Information and Knowledge Management*. 536–539.

[11] Joong Hyuk Chang and Won Suk Lee. 2003. Finding recent frequent itemsets adaptively over online data streams. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 487–492.

[12] Moses Charikar, Kevin Chen, and Martin Farach-Colton. 2004. Finding frequent items in data streams. *Theoretical Computer Science* 312, 1 (2004), 3–15.

[13] Hui Chen. 2014. Mining top-k frequent patterns over data streams sliding window. *Journal of Intelligent Information Systems* 42, 1 (2014), 111–131.

[14] Hui Chen, LihChyun Shu, Jiali Xia, and Qingshan Deng. 2012. Mining frequent patterns in a varying-size sliding window of online transactional data streams. *Information Sciences* 215, 1 (2012), 15–36.

[15] Yun Chi, Haixun Wang, Philip S. Yu, and Richard R. Muntz. 2004. Moment: Maintaining closed frequent itemsets over a stream sliding window. In *Proceedings of the 4th IEEE International Conference on Data Mining*. IEEE, 59–66.

[16] Edith Cohen. 2015. Stream sampling for frequency cap statistics. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 159–168.

[17] Edith Cohen and Haim Kaplan. 2007. Summarizing data using bottom-k sketches. In *Proceedings of the 26th Annual ACM Symposium on Principles of Distributed Computing*. Association for Computing Machinery, 225–234.

[18] Graham Cormode and Marios Hadjieleftheriou. 2009. Finding the frequent items in streams of data. *Communications of the ACM* 52, 10 (2009), 97–105.

[19] Graham Cormode and S. Muthukrishnan. 2005. An improved data stream summary: The count-min sketch and its applications. *Journal of Algorithms* 55, 1 (2005), 58–75.

[20] Graham Cormode and S. Muthukrishnan. 2005. What's hot and what's not: Tracking most frequent items dynamically. *ACM Transactions on Database Systems* 30, 1 (2005), 249–278.

[21] Youcef Djenouri, Marco Comuzzi, and Djamel Djenouri. 2017. SS-FIM: Single scan for frequent itemsets mining in transactional databases. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 644–654.

[22] Youcef Djenouri, Jerry Chun-Wei Lin, Kjetil Nørvåg, and Heri Ramampiaro. 2019. Highly efficient pattern mining based on transaction decomposition. In *Proceedings of the 2019 IEEE 35th International Conference on Data Engineering*. IEEE, 1646–1649.

[23] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Bay Vo, Tin Truong Chi, Ji Zhang, and Hoai Bac Le. 2017. A survey of itemset mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7, 4 (2017), e1207.

[24] Joao Gama. 2012. A survey on learning from data streams: Current and future trends. *Progress in Artificial Intelligence* 1, 1 (2012), 45–55.

[25] Poonam Goyal, Jagat Sesh Challa, Shivin Shrivastava, and Navneet Goyal. 2017. AnyFI: An anytime frequent itemset mining algorithm for data streams. In *Proceedings of the 2017 IEEE International Conference on Big Data*. IEEE, 942–947.

[26] Poonam Goyal, Jagat Sesh Challa, Shivin Shrivastava, and Navneet Goyal. 2020. Anytime frequent itemset mining of Transactional data streams. *Big Data Research* 21, 1 (2020), 100–146.

[27] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. 2007. Frequent pattern mining: Current status and future directions. *Data Mining and Knowledge Discovery* 15, 1 (2007), 55–86.

[28] Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data Record*, Vol. 29. Association for Computing Machinery, 1–12.

[29] Jochen Hipp, Ulrich Güntzer, and Gholamreza Nakhaeizadeh. 2000. Algorithms for association rule mining – a general survey and comparison. *ACM SIGKDD Explorations Newsletter* 2, 1 (2000), 58–64.

[30] Cheqing Jin, Weining Qian, Chaofeng Sha, Jeffrey X Yu, and Aoying Zhou. 2003. Dynamically maintaining frequent items over a data stream. In *Proceedings of the 12th International Conference on Information and Knowledge Management*. 287–294.

[31] Richard M. Karp, Scott Shenker, and Christos H. Papadimitriou. 2003. A simple algorithm for finding frequent elements in streams and bags. *Transactions on Database Systems* 28, 1 (2003), 51–55.

[32] Guoliang Li, Yang Wang, Ting Wang, and Jianhua Feng. 2013. Location-aware publish/subscribe. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 802–810.

[33] Hua-Fu Li and Suh-Yin Lee. 2009. Mining frequent itemsets over data streams using efficient window sliding techniques. *Expert Systems with Applications* 36, 2 (2009), 1466–1477.

[34] Hua-Fu Li, Suh-Yin Lee, and Man-Kwan Shan. 2004. An efficient algorithm for mining frequent itemsets over the entire history of data streams. In *Proceedings of the 1st International Workshop on Knowledge Discovery in Data Streams*. 1–10.

[35] Hua-Fu Li, Man-Kwan Shan, and Suh-Yin Lee. 2008. DSM-FI: An efficient algorithm for mining frequent itemsets in data streams. *Knowledge and Information Systems* 17, 1 (2008), 79–97.

[36] Yongsub Lim and U. Kang. 2017. Time-weighted counting for recently frequent pattern mining in data streams. *Knowledge and Information Systems* 53, 2 (2017), 391–422.

[37] Chih-Hsiang Lin, Ding-Ying Chiu, Yi-Hung Wu, and Arbee LP Chen. 2005. Mining frequent itemsets from data streams with a time-sensitive sliding window. In *Proceedings of the 2005 SIAM International Conference on Data Mining*. SIAM, 68–79.

[38] Hongyan Liu, Yuan Lin, and Jiawei Han. 2011. Methods for mining frequent items in data streams: An overview. *Knowledge and Information Systems* 26, 1 (2011), 1–30.

[39] Nishad Manerikar and Themis Palpanas. 2009. Frequent items in streaming data: An experimental evaluation of the state-of-the-art. *Data & Knowledge Engineering* 68, 4 (2009), 415–430.

[40] Amit Manjhi, Vladislav Shkapenyuk, Kedar Dhamdhere, and Christopher Olston. 2005. Finding (recently) frequent items in distributed data streams. In *Proceedings of the 21st International Conference on Data Engineering*. IEEE, 767–778.

[41] Gurmeet Singh Manku. 2016. *Frequent Itemset Mining over Data Streams*. Springer, 209–219.

[42] Gurmeet Singh Manku and Rajeev Motwani. 2002. Approximate frequency counts over data streams. In *VLDB*. Elsevier, 346–357.

[43] Ahmed Metwally, Divyakant Agrawal, and Amr El Abbadi. 2005. Efficient computation of frequent and top-k elements in data streams. In *Proceedings of the 10th International Conference on Database Theory*. Springer, 398–412.

[44] Barzan Mozafari, Hetal Thakkar, and Carlo Zaniolo. 2008. Verifying and mining frequent patterns from large windows over data streams. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*. IEEE, 179–188.

[45] Chedy Raïssi and Pascal Poncelet. 2007. Sampling for sequential pattern mining: From static databases to data streams. In *Proceedings of the 7th IEEE International Conference on Data Mining*. 631–636.

[46] Matteo Riondato and Eli Upfal. 2014. Efficient discovery of association rules and frequent itemsets through sampling with tight performance guarantees. *Transactions on Knowledge Discovery from Data* 8, 4 (2014), 25–41.

[47] Matteo Riondato and Eli Upfal. 2015. Mining frequent itemsets through progressive sampling with Rademacher averages. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 1005–1014.

[48] Moshe Shaked. 1975. On the distribution of the minimum and of the maximum of a random number of iid random variables. In *A Modern Course on Statistical Distributions in Scientific Work*. Springer, 363–380.

[49] Bai-En Shie, Vincent S. Tseng, and Philip S. Yu. 2010. Online mining of temporal maximal utility itemsets from data streams. In *Proceedings of the 2010 ACM Symposium on Applied Computing*. Association for Computing Machinery, 1622–1626.

[50] Saurabh Ranjan Srivastava, Yogesh Kumar Meena, and Girdhari Singh. 2020. Itemset mining based episode profiling of terrorist attacks using weighted ontology. In *Proceedings of the International Conference on Advanced Machine Learning Technologies and Applications*. Springer, 337–348.

[51] Xingzhi Sun, Maria E. Orlowska, and Xue Li. 2006. Finding frequent itemsets in high-speed data streams. In *Proceedings of the 2006 SIAM International Conference on Data Mining*.

[52] Syed Khairuzzaman Tanbeer, Chowdhury Farhan Ahmed, Byeong-Soo Jeong, and Young-Koo Lee. 2009. Sliding window-based frequent pattern mining over data streams. *Information Sciences* 179, 22 (2009), 3843–3865.

[53] Daniel Trabold. 2020. *Mining Frequent Itemsets from Transactional Data Streams with Probabilistic Error Bounds*. Ph.D. Dissertation. Rheinische Friedrich-Wilhelms-Universität Bonn.

[54] Daniel Trabold and Tamás Horváth. 2016. Mining data streams with dynamic confidence intervals. In *Proceedings of the International Conference on Big Data Analytics and Knowledge Discovery*. Springer, 99–113.

[55] Daniel Trabold and Tamás Horváth. 2017. Mining strongly closed itemsets from data streams. In *Proceedings of the International Conference on Discovery Science*. Springer, 251–266.

[56] Daniel Trabold, Tamás Horváth, and Stefan Wrobel. 2020. Effective approximation of parametrized closure systems over transactional data streams. *Machine Learning* 109, 6 (2020), 1147–1177.

[57] Luigi Troiano and Giacomo Scibelli. 2014. Mining frequent itemsets in data streams within a time horizon. *Data & Knowledge Engineering* 89, 1 (2014), 21–37.

[58] Jeffrey S. Vitter. 1985. Random sampling with a reservoir. *Transactions on Mathematical Software* 11, 1 (1985), 37–57.

[59] Xiaoyang Wang, Ying Zhang, Wenjie Zhang, and Xuemin Lin. 2014. Efficiently identify local frequent keyword co-occurrence patterns in geo-tagged Twitter stream. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. Association for Computing Machinery, 1215–1218.

[60] Xiaoyang Wang, Ying Zhang, Wenjie Zhang, and Xuemin Lin. 2016. Efficient identification of local keyword patterns in microblogging platforms. *IEEE Transactions on Knowledge and Data Engineering* 28, 10 (2016), 2621–2634.

[61] Raymond Chi-Wing Wong and Ada Wai-Chee Fu. 2006. Mining top-K frequent itemsets from data streams. *Data Mining and Knowledge Discovery* 13, 2 (2006), 193–217.

[62] Jeffery Xu Yu, Zhihong Chong, Hongjun Lu, and Aoying Zhou. 2004. False positive or false negative: Mining frequent itemsets from high speed transactional data streams. In *Proceedings of the 13th International Conference on Very Large Data Bases*. 204–215.

[63] Hui Zheng, Peng Li, Qing Liu, Jinjun Chen, Guangli Huang, Junfeng Wu, Yun Xue, and Jing He. 2020. Dual incremental fuzzy schemes for frequent itemsets discovery in streaming numeric data. *Information Sciences* 514, 10(2020), 15–43.

[64] Yunyue Zhu and Dennis Shasha. 2002. Statstream: Statistical monitoring of thousands of data streams in real time. In *Proceedings of the 28th International Conference on Very Large Data Bases*. 358–369.

Received January 2021; revised March 2021; accepted May 2021