SCIENCE CHINA
Information Sciences

• LETTER •

May 2020, Vol. 63 159101:1–159101:3
https://doi.org/10.1007/s11432-018-9849-y

# An effective scheme for top-$k$ frequent itemset mining under differential privacy conditions

Wenjuan LIANG[1,2], Hong CHEN[1], Jing ZHANG[1*], Dan ZHAO[1] & Cuiping LI[1]

[1]*Key Lab of Data Engineering and Knowledge Engineering of MOE, Renmin University of China, Beijing* 100872*, China;*
[2]*College of Computer and Information Engineering, Henan University, Kaifeng* 475001*, China*

Received 10 December 2018/Revised 30 January 2019/Accepted 6 March 2019/Published online 26 March 2020

**Citation** Liang W J, Chen H, Zhang J, et al. An effective scheme for top-$k$ frequent itemset mining under differential privacy conditions. Sci China Inf Sci, 2020, 63(5): 159101, https://doi.org/10.1007/s11432-018-9849-y

Dear editor,

Frequent itemset mining (FIM) is important in many data mining applications [1], such as web log mining and trend analysis. However, if the data are sensitive (e.g., web browsing history), directly releasing frequent itemsets and their support may breach user privacy. The protection of user privacy while obtaining statistical information is important. Differential privacy (DP) is a strong and rigorous standard for privacy protection. In this study, we focused on effectively discovering top-$k$ frequent itemsets under DP conditions. By adding a carefully selected amount of noise, DP ensures that the output of a computation is not sensitive to any individual tuple, and thus, user's privacy can be protected. The amount of noise is determined by the privacy budget $\epsilon$ and the sensitivity.

Several studies [2–4] have recently begun to address the issue of performing FIM while satisfying DP. The sensitivity is the size of candidate frequent itemsets, which is very large. According to Laplace mechanism (LM) [5], a large magnitude of noise must be added to the release result. To promote the utility of the release result, a potential solution evaluated in previously published studies was to decrease the dimension of long transactions in a differentially private manner before releasing it. For instance, Ref. [4] employed random sampling to truncate long transactions before releasing them. Random truncation may cause a significant amount of information loss, which also affects its utility. Ref. [2] proposed double standards to reduce the information loss associated with truncation. Ref. [3] proposed the splitting of long transactions instead of truncating them to reduce information loss; however, despite reducing information loss, they were relatively inefficient (Detailed related studies and comparison can be seen in Appendixes C and F). For this reason, we aim to design an effective scheme for FIM under DP conditions.

*Problem definition.* FIM refers to finding a set of patterns the support of which is greater than $\lambda$ (the support threshold, $0 < \lambda < 1$). Top-$k$ FIM is designed to find $k$ patterns the support of which is among the top $k$ in frequent itemsets. The top-$k$ FIM under DP conditions is defined as follows. Let $\widehat{\mathrm{FI}_k}$ denote the private top-$k$ frequent itemsets. After adding a certain amount of noise that satisfies LM or exponential mechanism (EM) [6] to the release process, the probability of outputting the same result for any pair of neighboring databases $(D, D')$ is bounded by $\exp(\epsilon)$, which can be formalized as $\frac{\Pr(\widehat{\mathrm{FI}_k}|D)}{\Pr(\widehat{\mathrm{FI}_k}|D')} \leqslant \exp(\epsilon)$.

*The overall scheme.* Our scheme comprises two processes: the first is splitting the transaction using count estimation, and the second is releasing based on weighted reservoir sampling and EM. To achieve $\epsilon$-differential privacy, $\epsilon$ is divided between the two processes: $\epsilon_1 = \alpha \cdot \epsilon$ $(0 < \alpha < 1)$ is used to split the transaction, and $\epsilon_2 = (1 - \alpha) \cdot \epsilon$ is used to genetate the private release. Our scheme satisfies

[*] Corresponding author (email: Zhang-jing@ruc.edu.cn)

$\epsilon = \epsilon_1 + \epsilon_2$ differential privacy.

*Transaction splitting via count estimation.* In this study, we employed count estimation (CE) to improve the efficiency of transaction spitting, which scans the database only once, and consumes only a small amount of preprocessing time. First the definition of CE is provided as follows.

**Definition 1** (Count estimation)**.** For an $n$-itemset $e$ ($n \geqslant 2$), a set of its subsets $S(e)$, a set of its $m$-subsets $S_m(e)$ and a set of counts for its $m$-subsets $S_m^C(e)$ are formally defined as follows:

- $S(e) = \{\beta | \forall \beta \text{ s.t. } \beta \in 2^e - e \text{ and } \beta \neq \phi\}$.
- $S_m(e) = \{\beta | \forall \beta \text{ s.t. } \beta \in S(e) \text{ and } |\beta| = m\}$.
- $S_m^C(e) = \{C(\beta) | \forall \beta \text{ s.t. } \beta \in S_m(e)\}$, $C(\beta)$ represents the count of $\beta$ that appeared in the transactions.

When all items of $e$ appear together in as many transactions as possible, the count of $e$ is at its maximum. When all items of $e$ appear exclusively in as many transactions as possible, the count of $e$ is at its minimum. Based on this observation, the definitions used for count estimation are defined as follows. The maximum possible count of $e$ is

$$C_{\max}(e) = \min(S_{n-1}^C(e)). \tag{1}$$

For two $(n-1)$-subset $e1$ and $e2$, $e = e1 \cup e2$, the minimum possible count of $e$ can be estimated

$$C_{\min}(e) = \begin{cases} \max(0, C(e1) + C(e2) - C(e1 \cap e2)), \\ \quad \text{if } e1 \cap e2 \neq \phi; \\ \max(0, C(e1) + C(e2) - |D|), \\ \quad \text{if } e1 \cap e2 = \phi. \end{cases} \tag{2}$$

Next, the method for privately splitting long transactions is described.

Privately calculating $\hat{F}$ and $\hat{P}$. First, short transactions are scanned, and the frequent 1-itemsets $\hat{F}$ and 2-itemsets $\hat{P}$ are calculated. Then, these are used in the privately splitting process. In this process, LM noise was added to the support of each itemset in $\hat{F}$ and $\hat{P}$. The budget allocated here is $\epsilon_1$, and $\epsilon_1$ is divided into the two calculations of $\hat{F}$ and $\hat{P}$ evenly. Each calculation is allocated to $\epsilon_1/2$. The sensitivity of the calculation of $i$-itemsets is $C_{l_{\text{opt}}}^i$. According to LM, the noise added to the support of each itemset of $\hat{F}$ is $\text{Lap}(2 \cdot l_{\text{opt}}/\epsilon_1)$, and the noise added to the support of each itemset of $\hat{P}$ is $\text{Lap}(2 \cdot C_{l_{\text{opt}}}^2/\epsilon_1)$.

Privately estimating the frequent patterns of a transaction. Based on the noise count of itemsets in $\hat{F}$ and $\hat{P}$, the frequent patterns (denote by CS) of each long transaction $t_i$ is estimated. First, because $\hat{F}$ and $\hat{P}$ denote the frequent itemsets with lengths less than 2, CS is initialized with $\hat{F}$ and $\hat{P}$. Then, the CS information is used to estimate the frequent patterns of $t_i$, the length of which is greater than 2. Checking all the candidate patterns of $t_i$ is extremely inefficient. It is not necessary to estimate a pattern the length of which is greater than $\delta$ ($\delta$ is the maximal length of the true top-$k$ frequent patterns), because it does not affect the results that are released. Therefore, if $|e|$ is greater than 2 and less than $\delta$, its maximum possible count $C_{\max}(e)$ is estimated based on its $(|e|-1)$ frequent patterns in CS according to (1). If $C_{\max}(e)$ is greater than $\lambda$, $e$ can be identified as a frequent pattern. Its minimum possible count is then estimated based on (2). Then $(e, C_{\max}(e), C_{\min}(e))$ is taken as an element, and added to CS.

Privately splitting a long transaction. When splitting a long transaction, if two frequent itemsets always appear together in each transaction, they should be split into one short transaction to minimize the information loss. The process of generating a short transaction involves finding an optimal short transaction $t_{\text{temp}}$ in the current transaction $t_i$. To measure whether a short transaction is optimal, the weight of $e$ is defined as follows:

$$f(e).\text{weight} = \gamma C_{\max}(e) + (1-\gamma) C_{\min}(e), \tag{3}$$

where $\gamma$ ($0 < \gamma < 1$) is a coefficient that can be used to adjust the proportion of the two estimated values.

**Problem 1** (Finding an optimal short transaction)**.** Given a long transaction $t_i$ and its estimated frequent itemsets CS, finding an optimal short transaction $t_{\text{temp}}$ can be achieved using

$$\begin{cases} \textbf{Objective: } t_{\text{temp}} \wedge \text{MAX}(f(t_{\text{temp}}).\text{weight}); \\ \textbf{Constraints: } (1)\, t_{\text{temp}} \subseteq t_i \wedge |t_{\text{temp}}| \leqslant l_{\text{opt}}; \\ \qquad (2)\, f(t_{\text{temp}}).\text{weight} \\ \qquad\qquad = \displaystyle\sum_{X_j \in \text{CS} \wedge X_j \subseteq t_{\text{temp}}} (f(X_j).\text{weight}). \end{cases}$$

Finding an optimal short transaction is equivalent to finding a short transaction with the maximum weight in the current $t_i$ and that satisfies the above constraints. This problem is a Knapsack problem, which is a typical NP-Hard problem. A greedy strategy is applied: by repeatedly finding the pattern with the largest weight both in $t_i$ and CS, and adding it to $t_{\text{temp}}$ until all the items of $t_i$ are allocated. When finding the pattern with largest weight, the information loss must be reduced. Therefore, the following strategy was developed: for each $e$ in CS, we confirmed the number $p$ of intersecting elements of $t_{\text{temp}}$ and $e$. If $p$ was found to be greater than 0, we increased the weight of $e$ using (4), and selected a pattern $e$ based on its updated weight. $p$ is the number of intersecting elements of $t_{\text{temp}}$ and $e$. $f(e).\text{weight}/|e|$

is the average weight:

$$f(e).\mathrm{weight}=f(e).\mathrm{weight}+(f(e).\mathrm{weight}/|e|)\cdot p. \quad (4)$$

Information loss analysis of transaction splitting. Splitting may cause information loss, because the support of some itemsets decreased after splitting. Suppose the length of a long transaction $t$ is $l$ ($l > l_{\mathrm{opt}}$) and $t$ contains an $i$-itemset $X$. The length constraint on a transaction is $l_{\mathrm{opt}}$. Let $a = l - \lfloor l/l_{\mathrm{opt}} \rfloor$ be the number of items in the short transaction with a length smaller than $l_{\mathrm{opt}}$. After splitting the transaction, the probability that $X$ remains in $\lceil l/l_{\mathrm{opt}} \rceil$ short transactions is as follows:

$$\Pr_{\mathrm{split}(i,l)}(X)=\begin{cases}\dfrac{\lfloor l/l_{\mathrm{opt}}\rfloor\binom{l-i}{l_{\mathrm{opt}}-i}}{\binom{l}{l_{\mathrm{opt}}}}, & \text{if } a<i;\\[2ex] \dfrac{\lfloor l/l_{\mathrm{opt}}\rfloor\binom{l-i}{l_{\mathrm{opt}}-i}}{\binom{l}{l_{\mathrm{opt}}}}+\dfrac{\binom{l-i}{a-i}}{\binom{l}{a}}, & \text{if } a\leqslant i.\end{cases}$$

**Theorem 1** (Information loss)**.** The remaining information rate of $i$-itemset $X$ after splitting is

$$R_m(X) = \sum_{k=i}^{l_{\mathrm{opt}}} \frac{f_k}{\sum_{j=1}^{n} f_j} + \sum_{k=l_{\mathrm{opt}}+1}^{n} \frac{f_k}{\sum_{j=1}^{n} f_j} \cdot \Pr_{\mathrm{split}(i,l)}(X). \quad (5)$$

Detailed analysis can be seen in Appendix B.

*Release based on weighted reservoir sampling and EM.* Based on the split database, a scheme was designed to privately release frequent itemsets by combining weighted reservoir sampling with an EM. When using EM, for each possible element $e$ in the algorithm's output domain, a quality function allocates a certain weight to $e$. The higher the weight for $e$, the more likely $e$ is to be selected as output. The sampling method used in EM is weighted random sampling. By using this method, the sampling set must be traversed twice. If the sampling set is large or the number of elements is uncertain, it is impossible or inefficient to employ this approach. In this case, weighted reservoir sampling, which is characterized by traversing the sampling set only once and using an auxiliary memory to store a valid $k$ sample element at any moment, can be used. This auxiliary memory is called the reservoir.

According to EM, it is necessary to define a quality function to determine the sampling weights of $e$ as follows:

$$e.\mathrm{score} = \exp\left(\frac{\epsilon' \cdot C_r(e)}{2k}\right). \quad (6)$$

The privacy budget allocated here is $\epsilon' = \epsilon_2/2k$. $C_r(e)$ is the updated support of $e$ after offsetting the information loss. Additionally, it is necessary to combine EM with weighted reservoir sampling to promote efficiency. The weighted reservoir sampling method used in our scheme is defined in [7]. The sampling weight of $e$ in our scheme is defined

$$e.\mathrm{sw} = r^{1/e.\mathrm{score}}. \quad (7)$$

Based on the weight in (7), the noisy top-$k$ frequent patterns are selected by traversing the frequent patterns set only once using reservoir sampling method. Before releasing the result patterns in the reservoir, we perturb the support of each result pattern with LM noise. The magnitude of the added noise here can be expressed as $\mathrm{Lap}(2k/\epsilon')$. Detailed algorithms are described in Appendix A.

*Conclusion.* Our scheme can promote the efficiency of FIM under DP conditions. Extensive experiments indicated that our scheme outperforms other state-of-the-art methods. Detailed privacy analysis and experiments are listed in Appendixes D–E.

**Acknowledgements** This work was supported by National Natural Science Foundation of China (Grant Nos. 61532021, 61772537, 61772536, 61702522) and National Key R&D Program of China (Grant No. 2018YFB1004400).

**Supporting information** Appendixes A–F. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

**References**

1 Garofalakis M, Gehrke J, Rastogi R. Querying and mining data streams: you only get one look a tutorial. In: Proceedings of ACM SIGMOD International Conference, Madison, 2002

2 Zeng C, Naughton J F, Cai J Y. On differentially private frequent itemset mining. Proc VLDB Endow, 2012, 6: 25–36

3 Su S, Xu S Z, Cheng X, et al. Differentially private frequent itemset mining via transaction splitting. In: Proceedings of the 32nd IEEE International Conference on Data Engineering, Helsinki, 2016. 1564–1565

4 Wang N, Xiao X K, Yang Y, et al. PrivSuper: a superset-first approach to frequent itemset mining under differential privacy. In: Proceedings of the 33rd IEEE International Conference on Data Engineering, San Diego, 2017. 809–820

5 Dwork C, Mcsherry F, Nissim K, et al. Calibrating noise to sensitivity in private data analysis. In: Proceedings of Theory of Cryptography Conference, New York, 2006. 265–284

6 McSherry F, Talwar K. Mechanism design via differential privacy. In: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, Providence, 2007. 94–103

7 Efraimidis P S, Spirakis P G. Weighted random sampling with a reservoir. Inf Process Lett, 2006, 97: 181–185