A Hierarchical-based Frequent Itemset Mining Method under Local Differential Privacy

Deming Kong, Jian Wang

Nanjing University of Aeronautics and Astronautics, College of Computer Science and Technology, Nanjing, 210016, China

Abstract

Frequent itemset mining aims at finding the sets of items that occur together frequently in many transactions, which can lay the foundation for further association rule mining. In this paper, to deal with the problem of massive amount of data, we adopt a specific model which expresses the attributes of item hierarchically. Based on the layering attributes, we propose a novel frequent itemset mining method under local differential privacy which deals with the frequent itemset mining problem layer by layer and it greatly improves the search efficiency. Experimental analyses based on real and synthetic datasets show that our method outperforms the state-of-the-art in terms of the computation cost.

Keywords

local differential privacy, frequent itemset mining, layering attribute

1. Introduction

Frequent Itemset Mining (FIM) is a key problem in data mining. By mining frequent itemset, merchants can discover the links between products, and thus predict customers' buying habits and improve their service quality, which is beneficial both for their own interests and the economic development of society. However, in the process of collecting users' information, there is a risk that some sensitive personal information such as users' purchase history, browsing preferences may leak and users do not want to expose their privacy [15]. In recent years, in order to protect privacy, local differential privacy [1] has been widely used in data analysis tasks as a secure and advanced privacy-preserving technique without the need for trusted third party and it can provide a better balance between users' privacy and data utility.

There has been some research on frequent itemset mining of set-valued data under local differential privacy [5,6,13]. However, there exits some problems in the process of frequent itemset mining under local differential privacy, e.g., the excessive computation cost and low data utility. To deal with the massive amount of data, we devise a novel framework based on dividing the category attributes of an item into layers and step-by-step processing. The hierarchical expression of attributes makes full use of the correlation between different layers of attributes. Specifically, the correlation here mainly refers to the inclusion relationship of category attributes from different layers. Layer-division strategy of attributes increases the available information dimension of the model and demonstrates the idea of divide and rule [4]. In addition, step-by-step processing greatly reduces the complexity of problem and improves the search efficiency.

In real world, people may have complex personal requirements. Fortunately, based on the layering structure of category attributes, our scheme can perfectly meet the complex requirements through mining frequent itemset from high layer to low layer. Thus, users are able to obtain the frequent itemset of different layers, rather than a single layer. Compared with other methods, our model is more flexible.

The main contributions of this paper are as follows.

ICBASE2022@3rd International Conference on Big Data & Artificial Intelligence & Software Engineering, October 21-23, 2022, Guangzhou, China

EMAIL: kongdeming@nuaa.edu.cn (Deming Kong); wangjian@nuaa.edu.cn (Jian Wang);
ORCID: 0000-0002-6202-2921 (Deming Kong); 0000-0002-8376-5898 (Jian Wang);

1) We adopt a hierarchical expression of category attributes to improve the search efficiency. In addition, mining frequent itemset at different layers can meet the complex personal needs of users.

2) Based on the layering attributes, we propose a novel frequent itemset mining method which deals with the problem layer by layer.

3) We introduce a specific model of local differential privacy to the method mentioned above, and our experimental results validate the effectiveness of the proposed method.

The rest of the paper is organized in the following order. Section 2 introduces background knowledge and problem definition of frequent itemset mining, Section 3 presents some previous work, Section 4 gives a complete implementation of our scheme, Section 5 shows experimental results of our scheme and compares it with other methods, and Section 6 concludes our work.

2. Preliminaries and Problem definition

2.1. Local Differential Privacy

Definition 1 (ϵ-Local Differential Privacy， ϵ-LDP) a perturbation mechanism $A: D \rightarrow O$ satisfies ϵ-LDP if and only if for any two users' data $d, d' \in D$, and any possible output $o \in O$ satisfies the following inequality (1).

$$\frac{P(A(d) = o)}{P(A(d') = o)} \leq e^{\epsilon} \qquad (1)$$

Theorem 1 (Sequential Composition Theorem) Let $\mathrm{A}_1, \mathrm{A}_2, \mathrm{A}_3, \ldots, \mathrm{A}_n$ be a series of LDP mechanisms where A_i implements ϵ_i-LDP, when the mechanism A on the data D are run in an independent random on $\mathrm{A}_1(D), \mathrm{A}_2(D), \mathrm{A}_3(D), \ldots, \mathrm{A}_n(D)$ or run them in some sequential primary order, at this point the mechanism A is considered to satisfy the $(\sum_{i=1}^{n} \epsilon_i)$-LDP.

2.2. Unary Encoding

To cope with the problem of input domain size larger than 2, Generalized Random Response (GRR) [15] and Unary Encoding (UE) [3] were proposed. It was later investigated that when the input domain size $d > 3e^{\epsilon} + 2$, the variance of UE is smaller than that of GRR and is therefore more suitable as a perturbation method [3].

Definition 2 (Unary Encoding, UE) first maps the input v as a d bit length vector B where only the bit corresponding to the user input v is set as 1. For the vector after the one-hot encoding B, UE is perturbed separately for each bit as the following Equation (2).

$$P(B'[i] = 1) = \begin{cases} p, & if B[i] = 1 \\ q, & if B[i] = 0 \end{cases} \qquad (2)$$

According to the ϵ-LDP definition, UE can provide $\epsilon = \ln \frac{p.(1-q)}{q.(1-p)}$ for LDP. In order to minimize the variance of UE, Wang [3] et al. proposed Optimal Unary Encoding (OUE), resulting in $p = \frac{1}{2}, q = \frac{1}{e^{\epsilon}+1}$. In the frequency estimation task, OUE has the smallest estimation variance of $Var = n\frac{4e^{\epsilon}}{(e^{\epsilon}-1)^2}$.

2.3. Padding-and-sampling

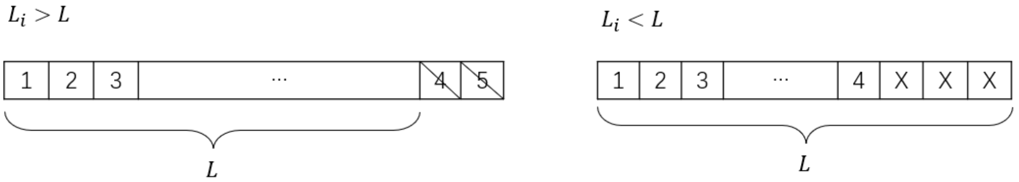


Fig.1. Padding-and-sampling

For the frequency estimation of set-valued data, padding-and-sampling protocol [2,7,8] is proposed to deal with the challenge of users' difference in transaction length.

Wang [8] proposed the Padding-and-Sampling-based Frequency Oracle (PSFO) protocol which is based on the padding-and-sampling. PSFO protocol is specified by three parameters: a positive integer L, a frequency oracle FO, and the privacy budget ϵ. It is composed of a pair of algorithms: $\langle\psi,\phi\rangle$, where ψ is used by each user to perturb his input value, and ϕ is used by the aggregator; ϕ takes as input the reports from all users, and can be queried for the frequency of each value. Hence, PSFO can be defined as the following Equation (3).

$$PSFO(L, FO, \epsilon) = \langle\psi_{FO(\epsilon)}(PS_L), \phi_{FO(\epsilon)}.L\rangle \qquad (3)$$

2.4. Problem definition

In the frequent itemset problem, the data collector knows the set of all items which is referred to as the full set I . There is a total of n users, and the i^{th} user has a set of items $v^i \subseteq I$ and we call this a transaction. Similarly, an itemset X is a collection of items, and the frequency of any $X \subseteq I$ is defined as the number of transactions containing that itemset X, which is also denoted as $f_X = \{v^i \middle| X \subseteq v^i\}$.

Our goal is to devise a secure and efficient scheme that enables data collector to implement frequent itemset mining under local differential privacy. We denote the frequency of top-k frequent item as $f(y)$, for the candidates of top-k frequent itemset Y, our optimization goal is: $f_Y = \underset{Y}{\operatorname{argmax}} \prod_{y \in Y} f(y)$. The top-k frequent itemset refers to an itemset whose frequency is among the k highest for all itemsets.

3. Related work

There has been some researches devoted to mining frequent itemset under local differential privacy. To address the problem of the varying number of items owned by users, Qin [7] proposes a two-phase approach for better frequency estimation. She first pads the set of items to a certain length L by some dummy items, and then samples a random item from it. They propose an algorithm for frequent item mining based on this approach, called LDPMiner. However, this method can only find frequent items while cannot be applied to the frequent itemset mining.

Inspired by LDPMiner, Wang [8] groups users and each of them is involved in an independent query task. This method improves the search accuracy and is known as the SVIM frequent item mining algorithm. Based on SVIM, he then constructs the frequent itemset candidates and proposes the SVSM frequent itemset mining algorithm, which is the first solution to the frequent itemset mining problem under local differential privacy. However, SVSM focuses on items with only a single attribute, which ignores the correlation between items in the real world.

Zhang [16] devised a MRR algorithm to perform frequent itemset mining under local differential privacy. It not only takes into account the multiple attributes of users' data, but also can meet the needs of users' personalized privacy protection. However, the process is inefficient in encoding because items that do not have values in some attributes still consume large computation cost, and the computation of frequent itemset supported by Bayes formula leads to low accuracy. Sharmin [15] proposed a novel method to find frequent itemset on small datasets. The scheme framework is better at maintaining the correlation between items, but the interactions between users and server are too complex to be applied in reality.

4. An efficient FIM method with layering attribute under LDP

In this section, we propose the solution to the problem of mining frequent itemset under local differential privacy. We first introduce the layering structure of category attribute, and then we present the complete implementation of our proposed method under LDP.

4.1. Layering structure of category attribute

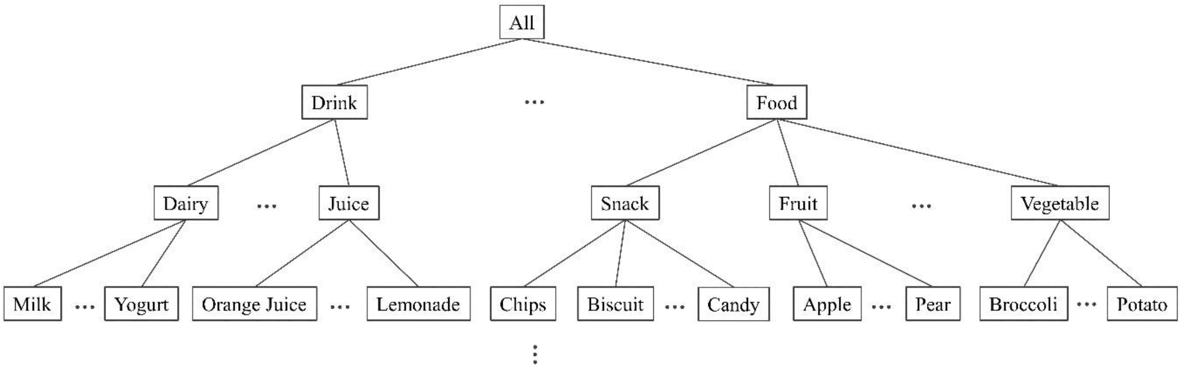


Fig. 2. Hierarchical expression of category attributes

Different from most of the researches nowadays which focus on the items with only a single category attribute, we divide the category attributes into many layers and express them hierarchically. The layering structure is shown in Figure 2. By dividing the attributes into layers, we are able to expand the original single attribute of an item into multi-dimensional category attribute. Hence, the problem of massive amount of data can be effectively solved by taking advantage of the correlation between different layers of attributes [4].

Based on the layering structure of category attribute, in order to improve the search efficiency, we devise a novel scheme of finding frequent itemset step by step, from high layer to low layer. We find the frequent itemset of large category firstly, and then to search the specific frequent itemset of small category within the given frequent itemset of large category, from coarse to fine, to greatly reduce the problem complexity.

If the number of a small category item is increasing, the number of corresponding large category will also increase, it makes sense that the frequent itemset of large category can be used for the further search of small category frequent itemset. The complexity of the frequent 2-itemset problem is O (n^2). e.g., there are ten large categories, and each large category contains 100 small category items. If we find the top three large categories of frequent 2-itemset, the problem scale will decline to O $(3 * 200^2)$, which is much smaller than O (1000^2). Although there is some loss in accuracy, it greatly improves the search efficiency.

4.2. Mining frequent itemset with layering attribute under LDP

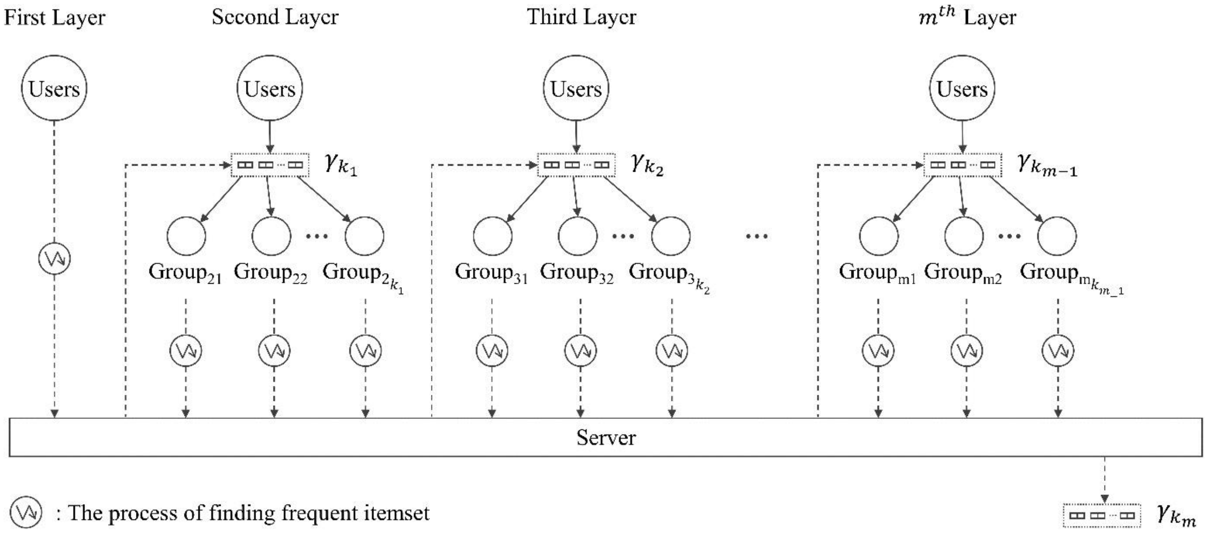


Fig. 3. Top-level framework (γ_{k_i} represents top-k_i frequent itemsets of the i^{th} layer)

We propose a multi-phase scheme framework in which users obtain frequent itemset of each layer after interacting with the server several times. Based on the idea of user pruning, after the itemset frequency estimation of each layer has been performed, only the top-ranked frequent itemsets of the current layer are sent to the next layer for continue mining. The iterative process of the frequent itemset mining for each layer continues and all estimated frequent itemsets of each layer are discovered eventually.

The total privacy budget ϵ is divided into m parts for m layers and each ϵ_i is used in the i^{th} layer. Specifically, we allocate privacy budget ϵ equally, which means each layer has the same average of $\epsilon_i = {}^{\epsilon}/_{m}$ and the whole process satisfies ϵ-LDP according to the sequential composition property.

In LDP, a common approach is to partition users into different groups, each answering one separate question than for all users to answer multiple questions each with part of the total privacy budget ϵ [8]. It is proved that the over division of privacy budget will greatly reduce the data utility and the grouping of users is beneficial to the accuracy of the final result. In our scheme, since there are k_i frequent itemsets of the i^{th} layer, we randomly divide users into k_i groups and each group of users participate in an independent task for the $(i+1)^{th}$ layer frequent itemset mining separately [9]. The top-level framework is shown in Figure 3.

For each group of users with IS from the previous layer, each user first finds the items from his own transaction which is in the range domain of the itemset IS. If a user does not possess any item in the range domain of IS, he will be regarded as a useless participant and does not need to go through the next series of operations. In other words, the frequent itemset is combined of the items only from the range domain, and any item outside the domain is not considered, which demonstrates the idea of user pruning. Algorithm 1 outlines the specific process of each layer.

Step 1: Padding and Sampling. As the number of items in a user transaction varies, existing FO protocols cannot handle this problem. Hence, it is necessary to adopt padding-and-sampling protocol to select one item to report. If the user's transaction length is less than L, the dummy item is padded; if the user's transaction length is larger than L, he randomly draws L items out. Although the dummy item could be sampled, its frequency will be ignored by the server during the frequency estimation process and it definitely could not be considered as a frequent item.

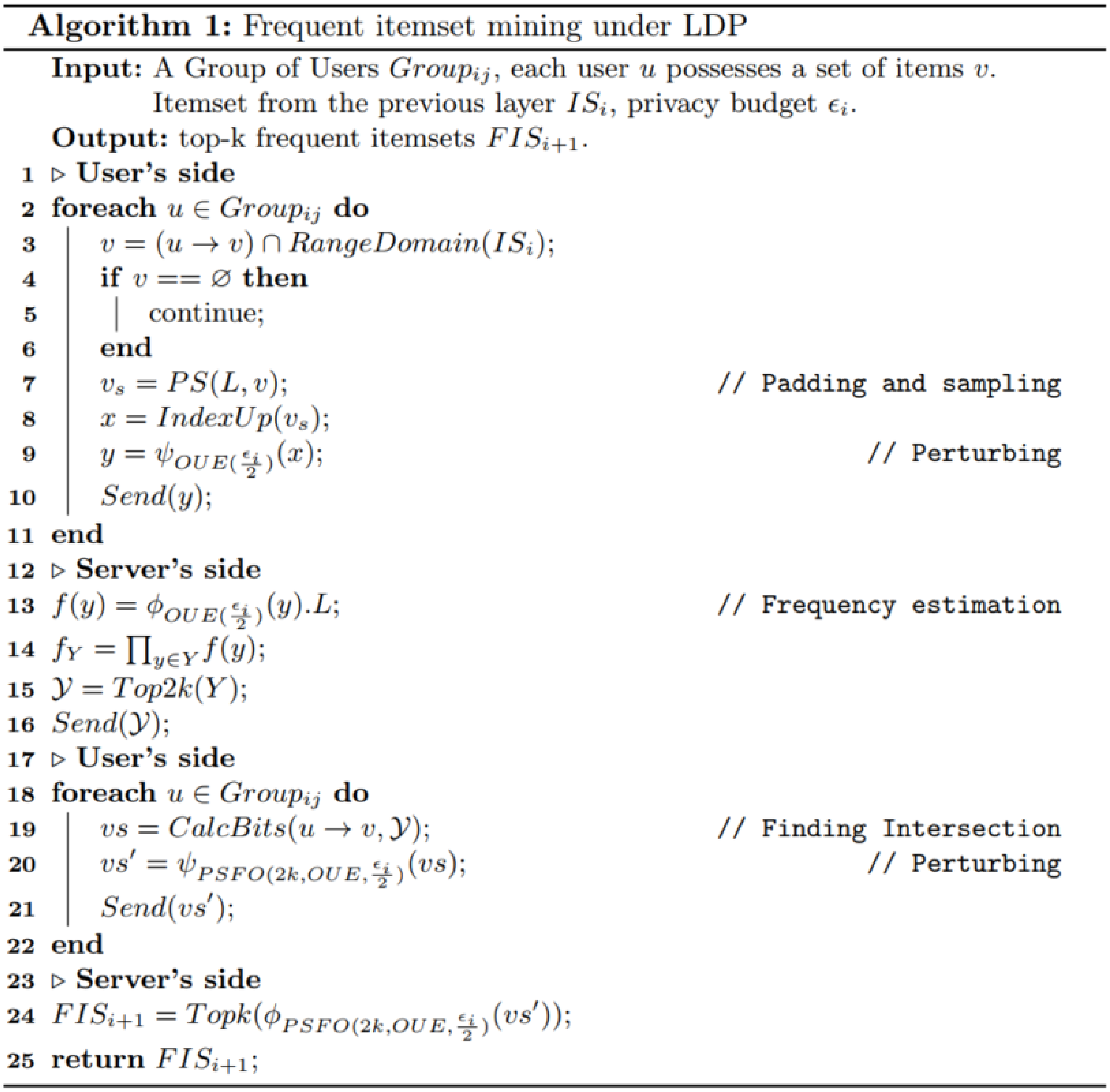
Step 2: Encoding and Perturbing. Assuming that there are N users, each of whom u_j possesses an item obtained by padding and sampling. Then, the sampled item is indexed up to get the value of the i^{th} layer category attribute. Each user adopts OUE algorithm to perturb the value with ${}^{\epsilon_i}/_{2}$. Finally, the perturbed value is submitted to the server for aggregation.

Step 3: Aggregation and Estimation. After receiving the perturbed value from users, the server calculates the frequencies of all category attributes by multiplying the estimated results with L and dropping the dummy item. Then, the server sorts attributes according to the estimated frequencies and combines them two by two to construct the top-2k frequent itemset candidates. The 2k candidates are sent to users.

The following task is quite similar to the original problem, with the difference that the universe of items becomes the 2k frequent itemset candidates, instead of the full set of d items. Specifically, the final top-k frequent itemsets are chosen from the 2k candidates.

Step 4: Candidates Estimation. After receiving 2k frequent itemset candidates, the user selects the set of items he owns in the 2k candidates. He then adopts padding-and-sampling algorithm to randomly select one item from the padded set. Note that at this time the padding length L becomes 2k rather than 90% of the number of users. Afterwards, the user adopts OUE algorithm to perturb the value with ${}^{\epsilon_i}/_{2}$ and sends it to the server.

The server estimates the frequencies of all candidates from the perturbed value and thus top-k frequent itemsets can be obtained.



```
Algorithm 1: Frequent itemset mining under LDP
Input: A Group of Users $Group_{ij}$, each user $u$ possesses a set of items $v$.
       Itemset from the previous layer $IS_i$, privacy budget $\epsilon_i$.
Output: top-k frequent itemsets $FIS_{i+1}$.
▷ User's side
foreach $u \in Group_{ij}$ do
    $v = (u \rightarrow v) \cap RangeDomain(IS_i)$;
    if $v == \varnothing$ then
        continue;
    end
    $v_s = PS(L, v)$;                                  // Padding and sampling
    $x = IndexUp(v_s)$;
    $y = \psi_{OUE(\frac{\epsilon_i}{2})}(x)$;                  // Perturbing
    $Send(y)$;
end
▷ Server's side
$f(y) = \phi_{OUE(\frac{\epsilon_i}{2})}(y).L$;                 // Frequency estimation
$f_Y = \prod_{y \in Y} f(y)$;
$\mathcal{Y} = Top2k(Y)$;
$Send(\mathcal{Y})$;
▷ User's side
foreach $u \in Group_{ij}$ do
    $vs = CalcBits(u \rightarrow v, \mathcal{Y})$;           // Finding Intersection
    $vs' = \psi_{PSFO(2k,OUE,\frac{\epsilon_i}{2})}(vs)$;        // Perturbing
    $Send(vs')$;
end
▷ Server's side
$FIS_{i+1} = Topk(\phi_{PSFO(2k,OUE,\frac{\epsilon_i}{2})}(vs'))$;
return $FIS_{i+1}$;
```

5. Experimental Results

In this section, we experimentally evaluate our method of mining frequent itemset for category data under local differential privacy, and verify that our proposed scheme performs effectively on real and synthetic datasets. Basically, we want to answer the following questions: Firstly, how many frequent itemsets can be effectively identified. Secondly, how much do we improve the search efficiency.

Environment: all algorithms are implemented in Python 3.7 and all the experiments are conducted on an Intel Core i5-9300H 2.4GHz PC with 16GB memory.

Datasets: we use the real dataset (Retail) [15] to show that our proposed method is suitable for real datasets. We show the results of the Retail dataset in Figure 4. We also implement our scheme on a synthetic dataset called norm-user [16] which obeys normal distribution and the result is shown in Figure 5. The synthetic normal distribution has mean of 4 and standard deviation of 0.1.

Our main goal of this experiment is to prove that by taking the correlations between items into account, our scheme can usually obtain frequent itemset results more accurately and efficiently, compared to SVSM [8] and Priv_OA [15] mechanisms.

Evaluation metrics: to compare the accuracy of the results of the mining, we adopt the $\mathrm{F-Score}$ [16] to measure our performance. Let $X_t = \{x_1, x_2, x_3, \ldots, x_k\}$ denotes the ground truth for top-k frequent itemset and X_r represents the set of frequent itemset generated by a specific algorithm. Then

$X_t \cap X_r$ is the set of real top-k itemsets that are identified by the algorithm. In this case, $\mathrm{F-Score}$ represents the accuracy of the search and is defined as the following Equation (4). In addition, we take the program running time as the expression the computation cost.

$$F - Score = 2 * \frac{precision * recall}{precision + recall} = \frac{|X_t \cap X_r|}{|X_t|} \qquad (4)$$

We conduct simulations on two kinds of datasets to compare the performance between three mechanisms. The first algorithm performing is our proposed method, the second one is SVSM algorithm and the last one is called Priv_OA which performs well on small datasets. The privacy budget ϵ we employ ranges from 1.0 to 5.0 with intervals of 1. In addition, we also explore the relationship between the accuracy of the results and the value of k through experiments. The value k ranges from 30 to 70 with intervals of 10.

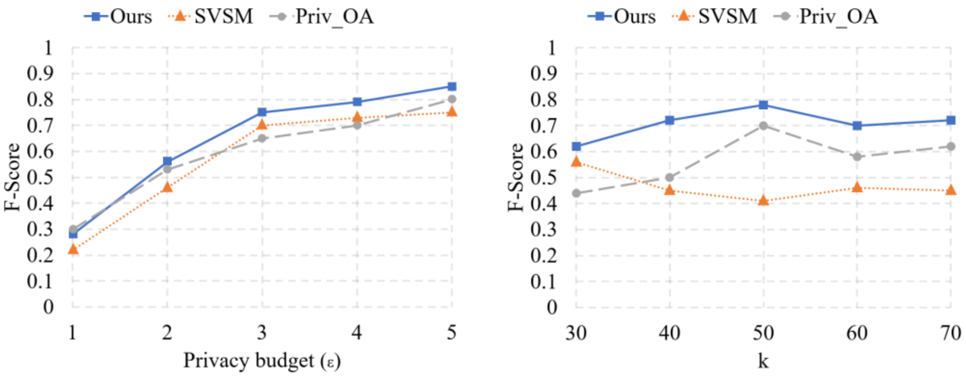


Fig.4. the $\mathrm{F-Score}$ results on Retail dataset

In Figure 4, we evaluate three methods on Retail dataset and plot the $\mathrm{F-Score}$ diagram based on the variation of privacy budget ϵ and the value of k. As can be observed in Figure 4, the identification accuracy which is denoted as $\mathrm{F-Score}$ increases with ϵ, and basically maintain, even with a little fluctuation when the value of k increases. It can be observed that our mechanism has an improvement over the other two methods, and with the increase of privacy budget ϵ and the value of k, that advantage is more obvious. That means the data utility has been improved under our proposed method. We also can find that with the increase of k, $\mathrm{F-Score}$ does not go up that fast. That is probably because the frequency difference of itemset is not that obvious and the error in frequency estimation also increases with k.

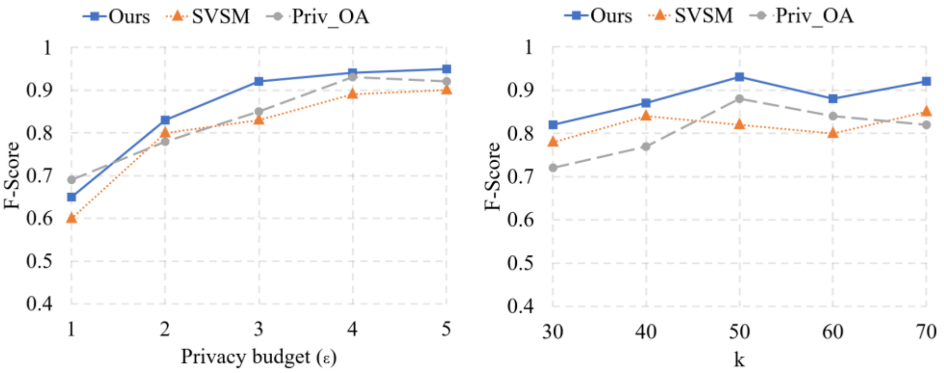


Fig. 5. the $\mathrm{F-Score}$ results on synthetic dataset

In Figure 5, we evaluate three methods on synthetic dataset and plot the $\mathrm{F-Score}$ diagram based on the variation of privacy budget ϵ and the value of k. Similar to the Retail dataset, it can be observed that the scheme we propose outperforms the other two methods. However, it can be noticed that when the privacy budget $\epsilon = 1$, our approach doesn't perform well compared to Priv_OA, probably because data utility is low when the privacy budget is small, and at this time, users' privacy can be better protected.

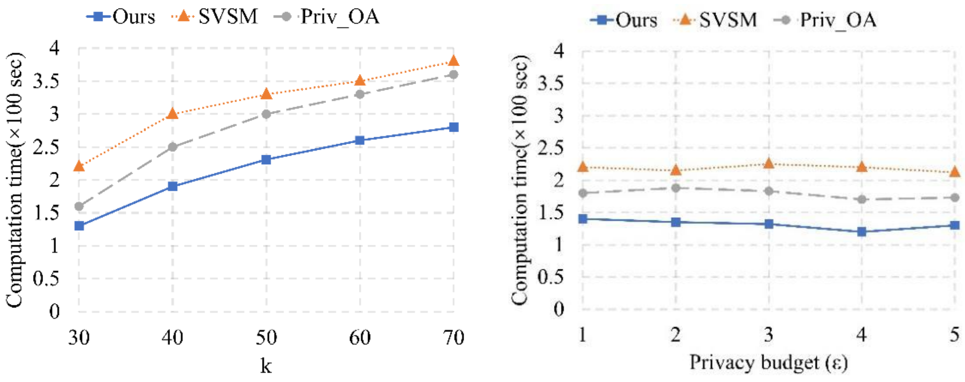


Fig. 6. the computation time on Retail dataset

Furthermore, we also explore the relationship between computation time and variables (the privacy budget ϵ and k) and the result is shown in Figure 6. This suggests that for the set of real top-k itemsets that are identified by algorithms, our proposed method can greatly reduce the computation cost, thus improve the search efficiency.

6. Conclusion

In this paper, a novel method is proposed to mine frequent itemset under local differential privacy. To improve the search accuracy, we divide the category attributes into layers and make full use of the correlation between different layer of attributes. Based on the layering attributes, in order to reduce the complexity of processing, frequent itemset mining is implemented step by step, from high layer to low layer. In addition, we have experimentally verified that our approach performs effectively on both real and synthetic datasets. Due to that we only consider the search of frequent 2-itemset in this paper, in the future, we will continue to do research on frequent multi-itemset mining under LDP. Furthermore, there can be more improvements in terms of different expressions of attribute structures.

7. References

[1] Dwork C. Differential privacy [C]//International Colloquium on Automata, Languages, and Programming. Springer, Berlin, Heidelberg, 2006: 1-12.

[2] Duchi J C, Jordan M I, Wainwright M J. Local privacy and statistical minimax rates[C]//2013 IEEE 54th Annual Symposium on Foundations of Computer Science. IEEE, 2013: 429-438.

[3] Wang T, Blocki J, Li N, et al. Locally differentially private protocols for frequency estimation [C]//26th {USENIX} Security Symposium ({USENIX} Security 17). 2017: 729-745.

[4] Wang N, Xiao X, Yang Y, et al. Collecting and analyzing multidimensional data with local differential privacy [C]//2019 IEEE 35th International Conference on Data Engineering (ICDE). IEEE, 2019: 638-649.

[5] Wang S, Huang L, Nie Y, et al. Privset: set-valued data analyses with locale differential privacy[C]//IEEE INFOCOM 2018-IEEE Conference on Computer Communications. IEEE, 2018: 1088-1096.

[6] Wang S, Qian Y, Du J, et al. Set-valued data publication with local privacy: tight error bounds and efficient mechanisms[J]. Proceedings of the VLDB Endowment, 2020, 13(8): 1234-1247.

[7] Qin Z, Yang Y, Yu T, et al. Heavy hitter estimation over set-valued data with local differential privacy [C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016: 192-203.

[8] Wang T, Li N, Jha S. Locally differentially private frequent itemset mining[C]//2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018: 127-143.

[9] Kairouz P, Bonawitz K, Ramage D. Discrete distribution estimation under local privacy [C]//International Conference on Machine Learning. PMLR, 2016: 2436-2444.

[10] Wang T, Li N, Jha S. Locally differentially private heavy hitter identification [J]. IEEE Transactions on Dependable and Secure Computing, 2019.

[11] Fanti G, Pihur V, Erlingsson Ú. Building a RAPPOR with the Unknown: Privacy-Preserving Learning of Associations and Data Dictionaries[J]. Proceedings on Privacy Enhancing Technologies, 2016, 3: 41-61.

[12] Bassily R, Nissim K, Stemmer U, et al. Practical Locally Private Heavy Hitters [J]. Advances in Neural Information Processing Systems, 2017, 30: 2288-2296.

[13] Wang N, Xiao X, Yang Y, et al. PrivTrie: Effective frequent term discovery under local differential privacy [C]//2018 IEEE 34th International Conference on Data Engineering (ICDE). IEEE, 2018: 821-832.

[14] Q. Xue, Y. Zhu and J. Wang, "Joint Distribution Estimation and Naïve Bayes Categoryification Under Local Differential Privacy," in IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 4, pp. 2053-2063, 1 Oct.-Dec. 2021, doi: 10.1109/TETC.2019.2959581.

[15] Sharmin Afrose, Tanzima Hashem, and Mohammed Eunus Ali. 2021. Frequent Itemsets Mining with a Guaranteed Local Differential Privacy in Small Datasets. 33rd International Conference on Scientific and Statistical Database Management Association for Computing Machinery, New York, NY, USA, DOI:https://doi.org/10.1145/3468791.3468807

[16] Xinyuan Zhang, Liusheng Huang, Peng Fang, Shaowei Wang, Zhenyu Zhu, and Hongli Xu. 2017. Differentially Private Frequent Itemset Mining From Smart Devices in Local Setting. in WASA. 433-44