2024 International Conference on Networking and Network Applications (NaNA) | 979-8-3503-7677-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/NaNA63151.2024.00029

HiDS Data clustering algorithm based on differential privacy

Shuhui Fang
School of Mathematics and Computer Scienc
Yunnan Minzu University
Kunming, China
e-mail: 1030046225@qq.com

Xuejun Wan
Scientific and technological information detachment
Yuxi Public Security Bureau
Yuxi, China
e-mail: xuejunwan@sina.com

Jun Wang
School of Mathematics and Computer Scienc
Yunnan Minzu University
Kunming, China
e-mail: 1972094448@qq.com

Lin Chai
School of Mathematics and Computer Scienc
Yunnan Minzu University
Kunming, China
e-mail: 2365077185@qq.com

Wenlin Pan
School of Mathematics and Computer Scienc
Yunnan Minzu University
Kunming, China
e-mail: 876219184@qq.com

Wu Wang *
School of Mathematics and Computer Scienc
Yunnan Minzu University
Kunming, China
e-mail: wwkmyn@139.com

***Abstract*—In recent years, the proliferation of high-dimensional sparse (HiDS) data has posed significant challenges to both data analysis and privacy protection. Traditional k-means algorithms often carry the risk of privacy leakage when applied to HiDS data, while differential privacy mechanisms offer effective protection for data privacy. Addressing such issues, this paper proposes a novel differential privacy k-means clustering algorithm. This algorithm first projects the data into a low-dimensional space, then privately generates a candidate center set, and finally performs privacy-preserving clustering on the candidate set. The algorithm proposed achieves the dual objectives of privacy protection and clustering analysis for HiDS data.**

Keywords-differential privacy; cluster; k-means; high-dimensional sparse data

I. *INTRODUCTION*

With the continuous development of the Internet and the widespread use of various devices, sensors, and applications, a vast amount of data is growing explosively. Within this data lies rich information and value, making the extraction of useful insights from it a pressing challenge for people to address. Clustering algorithms, particularly *k*-means clustering, have long been indispensable tools in data analysis, enabling the organization of unlabeled data into meaningful clusters based on similarity criteria. As an effective data analysis and processing tool, *k*-means clustering algorithm has been widely used in data analysi s tasks, such as recommendation system [1], database system [2], image processing [3], data mining [4] and other fields. However, as the proliferation of data continues unabated, concerns regarding data privacy have become increasingly prominent. Traditional clustering algorithms like *k*-means, while effective in data analysis tasks, often operate without regard for the privacy of individuals whose data is being analyzed. At present, there are many applications using *k*-means clustering algorithm, unfortunately, most of these applications do not take into account the disclosure of sensitive information, which may bring immeasurable threats to users [5].

The inherent vulnerability of *k*-means and similar algorithms lies in their reliance on explicit data points and centroids for clustering, which can inadvertently expose sensitive information. For instance, in scenarios involving personal data, such as healthcare or finance, the use of *k*-means clustering may inadvertently reveal private details about individuals or groups. Therefore, the research on clustering algorithm for privacy protection is of great practical significance for solving the problem of data privacy leakage [6].

For the privacy problems mentioned above, the concept of differential privacy has emerged as a promising solution [7]. Differential privacy offers a principled framework for ensuring that the inclusion or exclusion of any individual's data does not significantly impact the outcome of data analysis, thereby safeguarding individual privacy. Differential privacy as a powerful privacy protection technology has been widely used [8].

In recent years, there has been considerable research in the field of privacy protection for *k*-means clustering. For instance, the authors of [9] proposed a *k*-means algorithm

*Corresponding author

979-8-3503-7677-7/24/$31.00 ©2024 IEEE
DOI 10.1109/NaNA63151.2024.00029

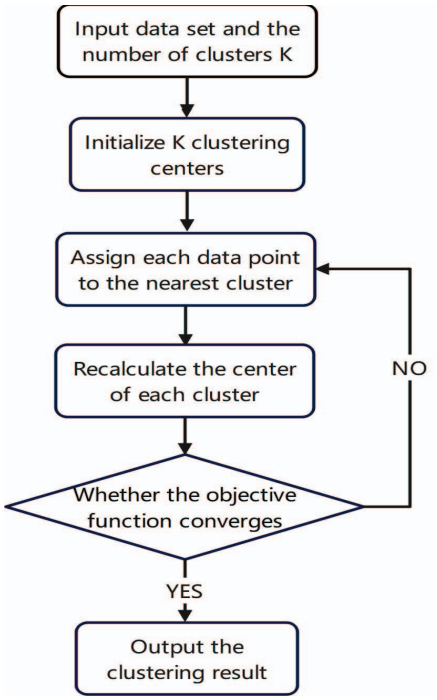

Figure 1. *K*-Means clustering algorithm flow diagram

with outlier removal and differential privacy, which selects initial centroids based on the density distribution of data points and adds Laplace noise to the original data to protect privacy. Reference [10] developed techniques to analyze the empirical error behavior of existing interactive and non-interactive methods, leading to an improved DPLloyd algorithm. However, these algorithms still suffer from efficiency issues due to the addition of significant amounts of noise. Therefore, reference [11] introduced a differential private *k*-means clustering algorithm based on cluster merging (DP-KCCM), which initially partitions the data into n × k clusters with differential privacy and then merges these clusters into the desired k cluster. Although a lot of research has been done on privacy clustering [12], there are still many basic problems that have not been solved. One of the long-standing challenges is to design a private clustering algorithm suitable for high-dimensional sparse data with small clustering losses. While some previous work has explored the possibility of non-private clustering in the context of sparse data [13], there are still many unknowns in the context of privacy.

Privacy clustering algorithm for high-dimensional sparse data sets is a challenging task, because traditional clustering methods may expose individual privacy information, while high-dimensional sparse data sets often contain a large number of features and a small number of non-zero elements, which increases the complexity of privacy protection. This paper combines differential privacy with *k*-means clustering algorithm to propose a new clustering algorithm, which aims to reduce the risk of privacy disclosure while maintaining the effectiveness of cluster analysis. This approach enables secure data analysis in privacy-sensitive applications by injecting controlled noise into clustering algorithms to ensure

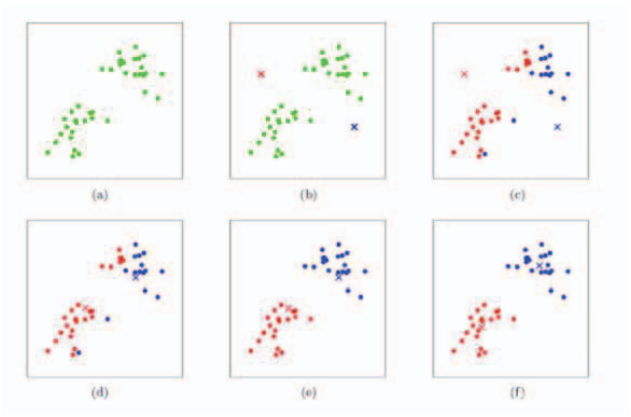

Figure 2. Diagram of clustering process

that sensitive information about individual data points is protected.

II. TECHNICAL PRELIMINARIES

A. K-Means Clustering

The *k*-means clustering algorithm is one of the classic partition-based clustering algorithms, known for its high efficiency and strong interpretability. It serves as the foundation for numerous clustering studies and is widely applied in data analysis, particularly in the data preprocessing stage. The basic idea of the *k*-means algorithm is to partition a dataset D with n samples and m dimensions into k clusters, ensuring that each cluster contains at least one data sample. Fig. 1 shows the flow diagram of *k*-means algorithm, the algorithm involves the following steps: initializing the number of clusters, selecting cluster centroids, calculating the distance from each data point to the centroids, assigning each data point to the cluster whose centroid is nearest based on the principle of minimum distance, iterating until the optimal result is achieved, and ending the process.

The *k*-means clustering algorithm utilizes the Euclidean distance between sample data points as the similarity measure. The Euclidean distance between two data points x_i and x_j is expressed as follows:

$$d\left(x_i, x_j\right) = \sqrt{\sum_{r=1}^{m}\left(x_{ir} - x_{jr}\right)^2}\,. \tag{1}$$

The cluster centroids are generated by calculating the average of the data points within each cluster. Let

The cluster centroids are generated by calculating the average of the data points within each cluster. Let $Z_i = \{x_1, x_2, \cdots, x_n\}$ be a cluster set, where x_i is a sample point within the cluster set. The formula for calculating the centroid C_i of this cluster set is:

$$C_i = \frac{\sum_{i=1}^{n} x_i}{n}\,. \tag{2}$$

The ideal objective of the clustering algorithm is to minimize the intra-cluster distance while maximizing the inter-cluster distance. Hence, the objective function of *k*-means clustering is typically based on this goal. Assuming a dataset X contains n data points to be partitioned into k clusters, represented by the set C, the sum of squared errors (SSE) is defined as:

$$SSE = \sum_{j=1}^{K} \sum_{i=1}^{n} \left\| x_i - C_j \right\|^2 . \quad (3)$$

The goal of clustering is to minimize the SSE value. If the SSE value remains unchanged before and after an iteration, it indicates that the cluster assignments no longer change, and the algorithm has converged. In other words, clustering is an iterative process, and the ideal termination condition is when the cluster assignments and cluster centroids no longer change. Additionally, loop count or change error can be set as termination conditions. In Fig. 2, it can be observed that the assignment schemes and cluster centroids remain unchanged in both (e) and (f) diagrams, indicating the end of the algorithm.

It should be noted that because minimizing the objective function (3) exactly is NP-hard, in this paper, we convert it to finding an approximate solution with an objective function of at most $\alpha \times OPT + \beta$, where OPT represents the optimal objective. Unlike non-private algorithms, all privacy algorithms must satisfy $\beta > 0$.

The *k*-means clustering algorithm is widely used due to its simplicity and efficiency. However, in practical applications, it has gradually revealed privacy leakage issues. Attackers can exploit the knowledge of cluster centroids and distances between samples and centroids obtained during two iterations to deduce sample data. Moreover, attackers can use background knowledge to infer the true values of sample data points.

B. Differential Privacy

The main idea of differential privacy [14] is that when a dataset D undergoes any counting, summing, or other query operations, resulting in a query result $f(D)$, adding or removing a data point from dataset D and then performing the same query operations on the dataset still yields the same query result $f(D)$. This ensures that whether a particular data point exists in the dataset or not, the final query result is hardly affected by its variation. Consequently, attackers are prevented from extracting the true data information through existing information.

Definition 1: A randomized algorithm M satisfies ε − differential privacy if for any neighboring databases D and D' (i.e., differing by only one record), and for any subset S of outputs of M, there exists a nonzero positive real number ε such that M satisfies the following inequality:

$$\Pr[M(D) \in S] \le e^{\varepsilon} \cdot \Pr[M(D') \in S], \quad (4)$$

where $\Pr[\cdot]$ denotes probability, and ε is referred to as the differential privacy parameter, which measures the resilience of the randomized algorithm M against attacks. A smaller ε value indicates stronger privacy protection provided. Therefore, the judicious selection of the ε parameter holds significance for both the privacy and utility of the data.

The differential privacy protection model employs data distortion to ensure data privacy, with a rigorous mathematical definition enabling quantitative assessment of privacy protection levels. This model can be applied to queries on massive datasets, safeguarding sensitive data by adding noise with specific characteristics. The amount of noise added is determined by the privacy parameter and sensitivity, independent of the dataset size.

C. Exponential Mechanism

The exponential mechanism [15] stands out as a prevalent technique in crafting algorithms for differential privacy, accommodating both numerical and non-numerical quandaries. It serves as a robust method for safeguarding individual privacy during the release of query outcomes by injecting randomness to obscure precise individual details. This approach ensures that sensitive information remains protected, making it a pivotal tool in data privacy preservation across various domains, including but not limited to healthcare, finance, and personal data analysis.

Definition 2: In the exponential mechanism, given a quality function $q(D,r)$, where $q(r)$ represents the quality of element r, and Δq denotes the sensitivity of the quality function, the mechanism M selects and outputs an element $r \in R$ with probability proportional to $\exp(\frac{\varepsilon q(D,r)}{2\Delta q})$.

In other words, the probability of selecting an element is directly related to its quality, with higher-quality elements having a higher probability of being chosen.

Definition 3: The sensitivity of an exponential mechanism is defined as:

$$\Delta q = \max_{r \in R} \max_{D,D' \left\| D - D' \right\|_1 \le 1} \left| q(D,r) - q(D',r) \right|, \quad (5)$$

where $q(D,r)$ represents the quality function, and r is an effective output of the exponential mechanism.

The exponential mechanism introduces noise into query results to balance accuracy and privacy. For a given query, it determines the probability of releasing each potential outcome based on its "privacy loss". Outcomes with lower privacy loss are favored, ensuring privacy protection while maintaining data accuracy.

D. Laplace Mechanism

The Laplace mechanism is a commonly used privacy-preserving technique employed in data or query result release processes by introducing noise. The basic idea of this mechanism is to add random noise following the Laplace distribution (also known as the double exponential distribution) to the computed results, thus achieving privacy

protection. Specifically, for a given query or computational task requiring differential privacy, such as calculating averages or sums, the Laplace mechanism adds Laplace noise to the result to obscure it to a certain extent, thereby safeguarding individual privacy.

Definition 4: For any function $f: \mathbf{N}^{|x|} \to \mathbf{R}^k$, the Laplace mechanism is defined as:

$$M_L(x, f(\cdot), \varepsilon) = f(x) + (\mathbf{Y}_i, \mathbf{Y}_2, \cdots, \mathbf{Y}_k), \tag{6}$$

where Y_i is a random variable independently and identically distributed (i.i.d.) sampled from $Y_i \sim Lap(\Delta f / \varepsilon)$.

The Laplace mechanism is built on a solid mathematical foundation, with its noise addition process strictly controlled by the Laplace distribution, enabling mathematical proof of privacy protection. Moreover, by adjusting the scale parameter, one can flexibly control the magnitude of noise, thereby adjusting the level of privacy protection. Larger parameter values introduce more noise, providing higher privacy protection, while smaller parameter values reduce noise, enhancing data accuracy.

III. A K-MEANS CLUSTERING ALGORITHM BASED ON DIFFERENTIAL PRIVACY

This paper proposes a novel *k*-means clustering algorithm based on differential privacy (DP-KMC) to address the issue of privacy leakage in high-dimensional sparse data clustering. The algorithm takes a high-dimensional sparse dataset as input and utilizes effective data summarization techniques to generate k cluster centers. This approach stands out for its utilization of both the exponential mechanism for selecting k suitable cluster centers and the Laplace mechanism for safeguarding the source data.

Specifically, the proposed scheme consists of two main steps: in the first step, the data is projected to a lower dimensional space using Johnson-Lindenstrauss(JL) lemma [16], and then the space is recursively subdivided to construct a candidate center set from each subregion and its center of mass; in the second step, the exponential mechanism is used to recursively exchange candidate centers, thereby focusing privacy from candidate centers to select the final k good centers. The above is the overall framework of HiDS Data clustering algorithm based on differential privacy, and its pseudo-code is shown in *Algorithm 1*. Below we will introduce the specific details of the algorithm.

A. Generate Privacy Candidate Sets

In this section, an effective algorithm will be introduced to construct a candidate center set with privacy in low-dimensional space, serving as a building block for privacy-preserving clustering. This algorithm consists of two main steps: recursively partitioning the space and constructing a privacy-preserving candidate center set. Initially, the data is projected to a lower-dimensional space using the JL lemma, and a novel technique is employed to construct a set of candidate centers with privacy in the projected space. Specifically, the lower-dimensional space is recursively

Algorithm 1: DP-KMC algorithm

Input: $D \leftarrow$data set, k, ε, $\delta \leftarrow$hyperparameters

Output: $c_1, c_2, \cdots, c_k \leftarrow$ cluster centers

1 Set latent p and number of trials T
2 **for** all $t = 1:T$ **do**
3 Randomly transform D from $R^m \to R^p$ using JL lemma
4 Initialize and update clusters to obtain clustering results C
5 Obtain cluster assignments for each data point $\{u_1, \cdots, u_k\}$
6 Use the Laplace mechanism to add noise and obtain $z^{(t)}$
7 **end**
8 Choose $c_1, c_2, \cdots, c_k$ from $z^{(t)}$ with exponential mechanism

partitioned, with each subdivided subregion and its centroid termed as a candidate centroid, whose probability depends on the number of points in the region and the expected privacy value. The algorithm iteratively partitions the space into multiple cubes until each cubes contains few points. Finally, the algorithm outputs the center of each cubes as a candidate center.

To ensure the smooth execution of the aforementioned steps, the following points need to be addressed: (1) output results are privacy-preserving, (2) the algorithm is computationally efficient, (3) the candidate set has a high (α, β) approximation ratio, meaning it contains a subset of size k with clustering loss at most $\alpha \times OPT + \beta$, where α and β are small. For the achievement of objective 1, we randomly decide whether to partition a cubes. For objective 2, we make the probability of further partitioning cubes negligible, and the upper bound on the size of the candidate set is the size of a simple partition tree, i.e., poly(n, k). For objective 3, it is sufficient to ensure capturing each cluster center within its own radius. As a large number of data points will converge around an optimal center, we can place candidate centers on cubes containing multiple points during partitioning. Random shifting and repetition are employed to avoid worst-case scenarios.

In summary, the algorithm in this section is primarily divided into two steps. The first step involves recursively partitioning the space. Initially, the high-dimensional space is projected to a lower-dimensional space using the JL lemma. Then, the lower-dimensional space is partitioned. We start from a cubes containing all data points and randomly decide whether to partition the current cubes based on the number of data points it contains, with the termination condition being that each cubes contains few points. It can be observed that the algorithm does not partition an empty cubes with high probability, thus ensuring computational efficiency, and appropriate stop probabilities are set during this process to protect the privacy of our algorithm. The second step involves generating candidate center sets. Based on the results generated by the first step of recursively partitioning algorithm, we consider each subdivided region and its centroid as candidate centroids, thus generating candidate center sets for subsequent algorithm use.

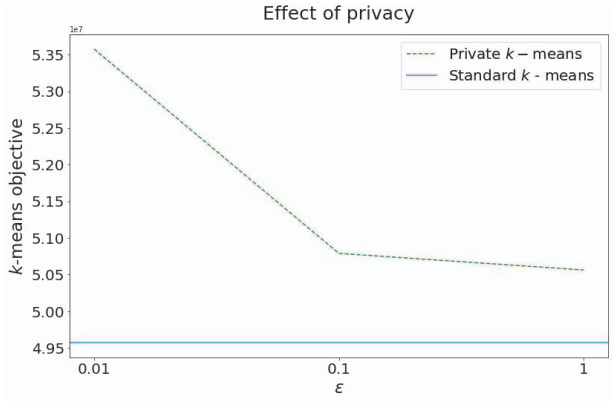

Figure 3. Comparison of privacy *k*-means and standard *k*-means

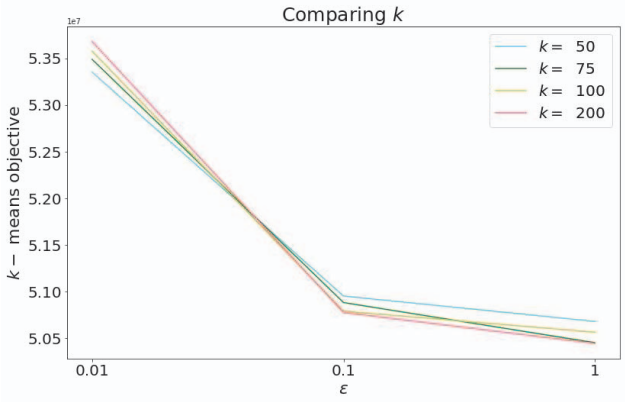

Figure 4. Comparing the effects of different k values on the algorithm

B. Perform Privacy Clustering on the Candidate Sets

In this section, we propose an efficient privacy-preserving clustering algorithm based on a candidate center set constructed in a low-dimensional space. The method consists of two main steps: privacy clustering in a low-dimensional space and privacy recovery in the original high-dimensional space. Our privacy clustering algorithm follows the technique of local exchanges on a privacy candidate center set, where we construct k pairs of exchange points, take the average of their gains, and relate it to the optimal loss OPT. Finally, we recover the private centers of the original high-dimensional space. Considering the low sensitivity of centers in the projection space, we can obtain a private center for each cluster by taking a noise mean of the clustering.

With the initial candidate centers obtained from the previous algorithm, we then assign data points to each cluster center and obtain the cluster assignment of each data point. After obtaining the cluster assignment of each data point, we introduce a process that extracts noise from the Gumbel distribution, with a probability density function of $p(y;b) = \frac{1}{b}\exp(-(y/b + e^{-y/b}))$, adds it to the mean of the centroids, and takes the top s items, where s represents the number of non-zero items in the dataset. This paper proposes a novel high-dimensional sparse recovery algorithm. The

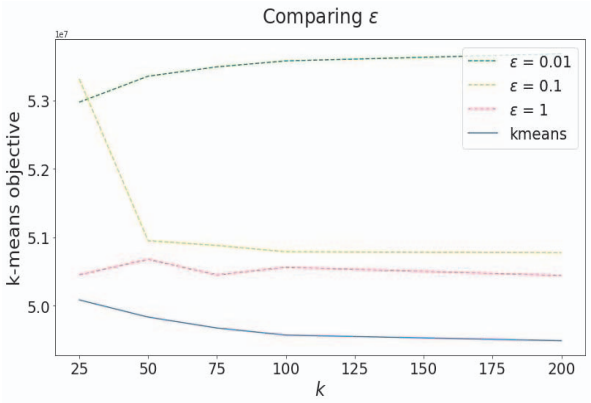

Figure 5. Comparing the effects of different ε values on the algorithm

Gumbel technique adds noise to the top-s items to prevent leakage of items with large values but does not protect the data itself. Therefore, in this algorithm, Laplace-distributed noise is added to the original data to protect the data itself and enhance privacy. Finally, we use the exponential mechanism to select the final cluster centers. Specifically, we select a candidate point with a probability proportional to $\exp(\frac{\varepsilon q(D,r)}{2\Delta q})$ as the final cluster center.

IV. SIMULATION

A. Dataset

To evaluate the effectiveness of the algorithm proposed in this paper on high-dimensional sparse datasets, this section uses synthetic data to evaluate the algorithm. We generate a synthetic dataset to simulate discrete processes such as rating or counting event occurrences. The relevant matrices are generated as $U \sim N(0,1) \in R^{m\times l}$, $V \sim N(0,1) \in R^{n\times l}$ and $X \sim Pois(\exp(UV^T))$, we set $n = 50000$, $m = 200$, $l = 100$.

B. Experimental process

To assess the quality of the DP-KMC algorithm, we utilize the *k*-means objective, defined as the average Euclidean distance from the original dataset to the closest mean. Formally, the loss $L(z_1,\cdots,z_k;D)$ for dataset $D = \{x_1,\cdots,x_m\}$ and mean $z_1,\cdots,z_k$ is defined as:

$$L(z_1,\cdots,z_k;D) = \sum_{i=1}^{m} \left\| x_i - z_j \right\|^2 . \tag{7}$$

In this experiment, we compare our proposed algorithm with the standard *k*-means algorithm (without privacy protection). From Fig. 3, it can be observed that the objective functions of the private *k*-means (our algorithm) and the standard *k*-means method are fairly close, indicating that our approach can maintain performance degradation within acceptable limits while preserving privacy.

Additionally, we examine the impact of different parameters on the algorithm. Firstly, we compare different values of k, specifically 50, 75, 100, and 200, and analyze

the variation of the *k*-means objective function. As shown in Fig. 4, we observe that larger values of k do not necessarily lead to better performance. For larger values of k , the private *k*-means algorithm repeats centroids rather than overfitting, thereby delaying the minimization of the objective function. Thus, the effect of k variation on the algorithm is not very pronounced.

Furthermore, we compare the impact of different ε values on the algorithm. From Fig. 5, it can be observed that as ε increases, the level of privacy decreases, leading to a decrease in the *k*-means objective. Eventually, it approaches the objective achieved by the standard non-private *k*-means implementation. This indicates that increasing ε can reduce the objective function, implying better algorithm performance.

V. CONCLUSION

This paper presents a differential privacy-based high-dimensional sparse data clustering algorithm, aiming to address the challenges of privacy protection and computational efficiency when clustering large-scale high-dimensional datasets. By integrating differential privacy mechanisms into the clustering process and leveraging sparsity characteristics, this algorithm effectively protects individual privacy information while maintaining high clustering accuracy and reducing computational costs. In the experimental section, we validate the effectiveness and performance advantages of this algorithm through experiments on synthetic datasets. The results indicate that, compared to traditional non-private clustering algorithms, our algorithm achieves comparable or superior clustering performance while preserving privacy.

In future research directions, we can further explore methods to enhance the clustering accuracy and computational efficiency of the algorithm, while considering characteristics and requirements from more practical application scenarios to achieve broader applicability. Additionally, we can also consider applying the differential privacy mechanism to other data mining and machine learning tasks to further advance the application and development of differential privacy in the field of data science.

ACKNOWLEDGMENT

The authors thank all the reviewers and editors for their comments and opinions on this study.

This work was supported by the [Yuxi Public Security Bureau] under Grant [number: Yunsaizhaozi 2023-095)]; [National Natural Science Foundation of China] under Grant [number 62362071].

REFERENCES

[1] F. McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the Netflix prize contenders", Proc. 15th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min. (KDD), pp. 627-636, 2009.

[2] M. Ester, H. Kriegel, J. Sander and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", Proc. 2nd Int. Conf. Knowl. Discov. Data Mining, pp. 226-231, 1996.

[3] H. Zhang, Z. Lin, C. Zhang and J. Gao, "Robust latent low rank representation for subspace clustering", Neurocomputing, vol. 145, pp. 369-373, Dec. 2014.

[4] P. Berkhin, "A survey of clustering data mining techniques" in Grouping Multidimensional Data, Berlin, Germany:Springer, pp. 25-71, 2006.

[5] F. Gao, J. He and X. Wu, "An Approach for Tracking Privacy Disclosure", Proc. 6th International Conference on Networked Computing and Advanced Information Management, pp. 294-299, August 2010.

[6] X. Zhang, H. Yang, Z. Li, F. He, J. Gai and J. Bao, "Differential private location privacy-preserving scheme with semantic location", Computer Science, pp. 300-308, 2021.

[7] C. Dwork, "Differential privacy", Proc. 33rd Int. Conf. Automata Languages Program., pp. 1-12, 2006.

[8] C. Dwork, "Differential Privacy: A survey of results", Proc. Theory Appl. Models Comput.: 5th Int. Conf, pp. 1-19, 2008.

[9] Q. Yu, Y. Luo, C. Chen and X. Ding, " Outlier-eliminated k -means clustering algorithm based on differential privacy preservation ", Appl. Intell., vol. 45, no. 4, pp. 1179-1191, Dec. 2016.

[10] D. Su, J. Cao, N. Li, E. Bertino and H. Jin, "Differentially private k-means clustering", Proc. 6th ACM Conf. Data Appl. Secur. Privacy, pp. 26-37, 2016.

[11] T. Ni, M. Qiao, Z. Chen, S. Zhang and H. Zhong, " Utility-efficient differentially private k -means clustering based on cluster merging ", Neurocomputing, vol. 424, pp. 205-214, Feb. 2021.

[12] D. Su, J. Cao, N. Li, E. Bertino, M. Lyu and H. Jin, "Differentially private k-means clustering and a hybrid approach to private optimization", ACM Trans. Privacy Secur., vol. 20, no. 4, pp. 1-33, Oct. 2017.

[13] A. Barger and D. Feldman, "k -means for streaming and distributed big sparse data ", Proc. SDM, pp. 342-350, 2016.

[14] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy", Found. Trends Theor. Comput. Sci., vol. 9, no. 3, pp. 211-407, 2014.

[15] F. McSherry and K. Talwar, "Mechanism design via differential privacy", Proc. IEEE Symp. Found. Comput. Sci. (FOCS), pp. 94-103, 2007.

[16] E. J. Candès and T. Tao, "Decoding by linear programming", IEEE Trans. Inf. Theory, vol. 51, no. 12, pp. 4203-4215, Dec. 2005.