Efficient-Secure k-means Clustering Guaranteeing Personalized Local Differential Privacy

Yuling Luo, Zhangrui Wang, Shunsheng Zhang(✉), and Junxiu Liu

School of Electronic and Information Engineering, Guangxi Normal University, Guilin, China
shunszhang@gxnu.edu.cn

Abstract. One highly discussed research topic is user privacy protection and the usability of models in data mining tasks. Currently, the most k-means clustering approach using differential privacy is based on trusted third-party servers. However, malicious servers exist in many applications and cause privacy leakages of user data. The Personalized Local Differential Privacy k-means algorithm (PLDP k-means) is proposed in this paper. To satisfy the PLDP mechanism, a perturbation mechanism is used to perturb the user data at the local side. Then clustering is completed by iteration between the local and server sides. The third-party server remains inaccessible to the real user data and considers the users' personalized privacy demands in the proposed algorithm. In addition, the iterative centroid perturbation algorithm is proposed in this paper for resisting inference attacks and improving the utility of clustering via a privacy budget allocation sequence. Theoretical analysis demonstrates the privacy of the proposed algorithm. Experimental results indicate that the proposed algorithm effectively preserves the utility of clustering while satisfying the PLDP mechanism.

Keywords: Cluster · k-means · Privacy protection · Personalized Local Differential Privacy

1 Introduction

The popularisation of smart devices and the development of big data analytics has led to tremendous growth in the generation, collection, and analysis of personal digital information. The useful information extracted from massive data can bring immeasurable value [1,2]. As a classical data analysis method, clustering is a type of unsupervised learning method. k-means is one of the most popular clustering methods due to its efficiency and simplicity [3]. Although the data analysis has great potential, it also has a risk of leakage of user privacy. Sensitive information such as medical, location, and financial data can directly lead to users' private information leakage. Traditional anonymization methods wipe out identifiers that cannot resist both differential and background knowledge attacks. An attacker can correlate or identify users' private information.

W. Meng et al. (Eds.): ICA3PP 2022, LNCS 13777, pp. 660–675, 2023.
https://doi.org/10.1007/978-3-031-22677-9_35

Therefore, ensuring there is no leakage of users' private information and maintaining a high level of utility in clustering becomes a problem that needs to be solved.

The Differential Privacy (DP) model is currently considered as a reliable model with rigorous and falsifiable privacy guarantees [4]. Compared with traditional protection models such as anonymity and random perturbation, differential privacy has significant advantages in privacy preservation in cluster analysis [5,6]. A differential privacy-based model for cluster analysis, which is referred to as Differential Privacy k-means Algorithm (DP k-means), has been widely applied for its efficiency and privacy preservation [7,8]. DPLloyd-Impr made an improvement on DPLloyd by introducing the concept of sphere packing [9]. DP-KCCM, as a novel algorithm, is effective when cluster merging and adaptive noise mechanisms are adopted to improve clustering utility [10]. The above work improves DP k-means from data pre-processing, cluster delineation, etc., and is based on trusted third-party servers. The servers can collect real user data, perform clustering and uniformly add noise. However, with the development of cloud computing and the diversification of data analysis demands, the assumption that all third-party servers are trustworthy is not valid, as malicious servers may steal and take advantage of users' private information.

Local Differential Privacy (LDP) [11] was proposed because third-party servers cannot be trusted. LDP has more stringent privacy requirements than DP. It requires users to perturb their data at the local side and sends it to an untrusted server. LDP has also been applied to practical cases to create feasible solutions [12,13]. A k-means algorithm based on LDP was proposed in [14] to protect location data through feature transformation and privacy budget allocation. Although LDP can effectively address the problem of privacy leakage on third-party servers, it still faces the challenge of reduced clustering utility due to excessive noise [15]. Owing to the perturbation of user data at the local side, the noise of LDP is larger compared to that of DP. The influence of noise is further amplified in the clustering iterations. Also, most research on LDP implicit assumption is that there is uniform protection of the private information of all users. However, different users and data often have different privacy requirements. To address the above issue, Personalized Local Differential Privacy (PLDP) was proposed in [16], which allows each user to set the privacy level of their data independently.

Based on the above discussions, the main issue that needs to be addressed is how to take into account the protection of users' private information and the utility of clustering in k-means clustering. A clustering framework based on the PLDP k-means algorithm is proposed in this paper. Firstly, the user can perturb sensitive data at the local side by the PLDP k-means algorithm and send it to the server, which performs high-quality k-means clustering on the perturbed data. Thus, the threat of malicious servers is eliminated while the users' personalized privacy demands are met. In addition, an iterative centroid perturbation algorithm is proposed, which prevents privacy leakage caused by inference attacks by perturbing the centroids in the iterative process. The proposed algorithm also reduces the impact of perturbation on the clustering utility by designing a

privacy budget allocation sequence. The main contributions of this paper are as follows.

1) A clustering framework based on the PLDP k-means algorithm is proposed. In the framework, the server does not access users' private information while ensuring quality clustering and users' personalized privacy demands.
2) Iterative centroid perturbation algorithms are proposed to address the potential leakage of private information during iteration. They help prevent inference attacks and further protect users' private information.
3) Theoretical analysis demonstrates the privacy protection capability of the proposed mechanism, and extensive experiments show that the proposed algorithm has better or similar performance than existing DP k-means algorithms. To the best of our knowledge, this paper is the first attempt at adopting PLDP in k-means clustering.

The rest of this paper is organized as follows. The basic concepts required for this framework and the related technical foundation are introduced in Sect. 2. The proposed approach is present in Sect. 3. The experimental results and analysis are illustrated in Sect. 4. Finally, the paper is concluded in Sect. 5.

2 Preliminaries

In this paper, the concept of personalized local differential privacy is adopted. To make the paper more self-contained, some basics of LDP and PLDP are briefly introduced in this section.

Differential privacy is a privacy-preserving model widely used in data analysis, in which the real data of all users is protected by a trusted data collector. However, the prerequisite of a trusted data collector usually does not hold in real-world applications. LDP is an extension of DP that extends to the local settings. LDP implements data sanitization locally by designing random perturbation algorithms that comply with differential privacy requirements. This way, sensitive data information is protected without relying on trusted third-party collectors. The following is the formal definition of LDP.

Definition 1. *(ε-LDP). A randomized mechanism $F : D \to R$ satisfies ε-LDP iff for any possible output result t^* ($t^* \subseteq R$) on any two records t and t' ($t, t' \subseteq D$) that satisfies Eq. 1.*

$$Pr\left[F(t) = t^*\right] \leq e^{\varepsilon} \times Pr\left[F\left(t'\right) = t^*\right]. \tag{1}$$

The parameter ε is the privacy budget, which is public and usually set in [0,2]. The value of ε determines the probability of outputting the same result t^* for any two input values t and t' of the algorithm F. Thus, stronger (weaker) privacy guarantees are provided by smaller (larger) values of ε.

The LDP provides a way to protect private data on the local side of the user, but different privacy protection requirements may exist for different users and data. Therefore, PLDP is adopted to satisfy different privacy requirements

in this paper. Each user in PLDP has a set of optional parameters (G_i, ε_u), ε_u representing the desired strength of privacy protection for that user, i.e., the privacy budget. G_i represents a security range specified by the user containing his real data, where the user data is indistinguishable from other data.

Definition 2 *((G_i, ε_u)-PLDP). Given a set of privacy requirements (G_i, ε_u) to one user n, a randomized mechanism $F : D \to R$ satisfies (G_i, ε_u)-PLDP iff for any possible output result t^* $(t^* \subseteq R)$ on any two records t and t' $(t, t' \subseteq G_i)$ that satisfies Eq. 2.*

$$Pr\left[F(t) = t^*\right] \leq e^{\varepsilon_u} \times Pr\left[F\left(t'\right) = t^*\right]. \tag{2}$$

when G_i is set to the domain D, and all users are unified ε, PLDP is equivalent to LDP.

Differential privacy has two important combinatorial properties: the sequential and parallel combinatorial properties, which are formally defined as follows.

Property 1 (sequence combinability). Given a dataset D and privacy algorithms $\boldsymbol{F} = \{F_1, F_2, \ldots, F_n\}, F_i(1 \leq i \leq n)$ satisfies the ε_i-DP. Then the sequence combination of $\{F_1, F_2, \ldots, F_n\}$ on D satisfies ε-DP, where $\varepsilon = \sum_{i-1}^{n} \varepsilon_i$.

Property 2 (parallel combinability). Given a dataset D, divide it into n disjoint subsets, $\boldsymbol{D} = \{D_1, \ldots, D_n\}$, let F be any privacy algorithm that satisfies ε_i-DP, then the algorithm F satisfies ε_{max}-DP on D.

3 Proposed Approach

In this section, the PLDP k-means clustering algorithm is proposed, and its privacy is demonstrated. Existing privacy issues in clustering analysis are first analyzed. The overall flow of the proposed framework is then described, and the corresponding design of the perturbation mechanism based on PLDP theory is given. Finally, the privacy of the proposed overall system is proved theoretically.

3.1 Overview

The privacy issues faced by DP and LDP k-means clustering model and the solutions are analyzed in this subsection. A third-party data collector collects sensitive data (e.g., location, income, cases, etc.) from many users, processes it using the k-means algorithm, and shares or publishes the model results to partners or public platforms. When users are faced with third-party collectors (e.g., service providers, etc.) asking for their data, protecting their privacy becomes an issue that must be addressed. DP is considered an effective solution to this problem by perturbing the user's data on a third-party server so that neither the attacker nor the subsequent release can cause a leakage of the user's privacy. However, the attacker may be external, or the data collector may be malicious, knowing all the user's real data. LDP protects user data assuming that third-party servers are not trusted. The way solves the problem of malicious data

collectors is that the data is perturbed by the LDP at the local side and then uploaded to the server. The new problem is that due to LDP properties, there are limitations in protecting user data, and the availability of perturbed data is generally considered inferior to that of DP. At the same time, the risk of privacy leakage cannot be completely avoided by simply perturbing the data in clustering. So the problem is to design a model that achieves a better utility of clustering while avoiding the influence of malicious collectors.

A clustering framework based on the PLDP k-means algorithm is proposed in this paper to address the issues mentioned above. A randomized perturbation algorithm satisfying PLDP is used to perturb the user's local data, eliminating the risk of malicious collectors while satisfying personalized privacy requirements and enhancing the utility of clustering. Meanwhile, an iterative clustering centroid perturbation algorithm perturbs the real clustering information locally to prevent privacy leakage due to inference attacks.

3.2 Proposed Framework

A framework based on PLDP k-means that can solve the above problem is proposed, and its overall framework is shown in Fig. 1. The clustering model has a user set $U = \{u_0, u_1, \ldots, u_{n-1}\}$, an attribute set $A = \{a_0, a_1, \ldots, a_{d-1}\}$. Each user has a d-dimensional data vector $S_i = \{s_0, s_1, \ldots, s_{d-1}\}$. where $0 < i < n$, $0 < j < d$ and $d = |S_i|$ is the number of attributes. s_j corresponds to a numerical value of a_j. The target of k-means is to classify the user data into k clusters $C = \{c_0, c_1, \ldots, c_{k-1}\}$.

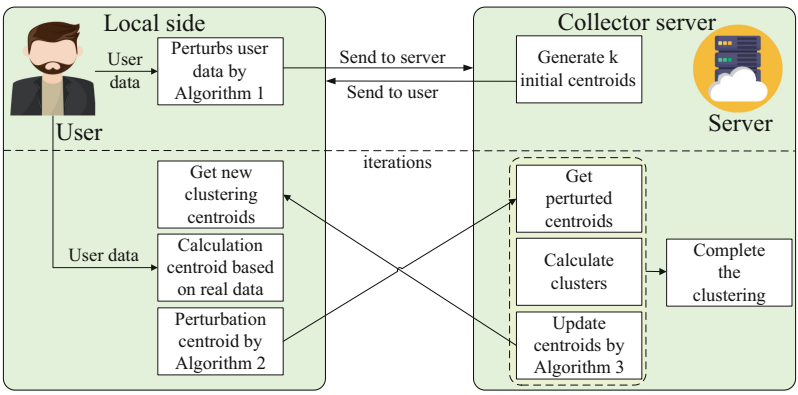

Fig. 1. Cluster privacy-preserving framework based on PLDP k-means.

As shown in Fig. 1, a clustering privacy-preserving framework based on PLDP is proposed. The proposed framework consists of two parts: the local side and the collector server. The local side describes how the user data is perturbed by the PLDP perturbation algorithm. The collector server describes how k-means is

performed based on the perturbed data. The user data S_i is perturbed to get S_i^* by Algorithm 1 at the local side and then sent to the server. The server generates k initial centroids by the initial centroid selection algorithm and attribute set A, then sends them to the local side. The next clustering iteration is performed. The local side calculates the distance between the real user data S_i and each centroid received from the server to find the nearest centroid c_i and the corresponding clusters. The found centroid c_i is then perturbed to get c_i^* by Algorithm 2 and sent to the server. The server updates a new set of centroids based on the received C^* and the perturbed data S_i^* by Algorithm 3, then sends them to the local side. The iterative process is repeated until the results converge.

Local Side Method. As shown in Fig. 1, the local side consists of two core components, user data perturbation, and centroid perturbation.

User Data Perturbation. In contrast to the usual LDP k-means approach of converting the data into binary strings and then perturbing each dimension to obtain the perturbed results before aggregation, this paper normalizes the user data vector S_i to [−1,1] through data pre-processing for the next perturbation process. Because each bit of the binary string has to be equally assigned privacy budget ε, which may cause excessive noise problems when the budget is small, or the number of bits in the string is large.

The Duchi solution [17] is a multidimensional data perturbation scheme based on LDP. Since S_i has already completed data pre-processing to obtain $S_i' = \{s_0', s_1', \ldots, s_{d-1}'\}$, the Duchi-based PLDP mechanism can be used to perturb user data. In the clustering model of this paper, a set of privacy parameters (G_i, ε_u) can be self-selected by each user. ε_u represents the user's selected privacy budget, i.e., the user's requirement for the strength of data protection. Averaging ε_u by user data dimension d to obtain $\varepsilon_d = \frac{\varepsilon_u}{d}$. $G_i = \{g_0, g_1, \ldots, g_{d-1}\}$ represents the user's acceptable security range, and $g_j (0 < j < d)$ represents the security range of the j-th dimensional data. e.g., a set of age data distributed between [1,100]. A user data is 25 years old, and after LDP perturbation, the perturbed data range is between [1,100], representing the user's expectation that their age data is indistinguishable in the range [1,100], a wide privacy requirement that is generally unnecessary. In PLDP, the user needs to choose a security range g_j. The range and size of g_j is user-defined, and the user's real data values must be included within the security range. For example, g_j=[10,40] means that the age is indistinguishable within the range [10,40] to satisfy the user's privacy demands. w_j and m_j are defined as the size and midpoint of g_j. Since a secure region symmetric about 0 needs to be obtained, each user moves the secure range, and the user data points within the range move to $s_j'' = s_j' - m_j$. After processing the $S_i'' = \{s_0'', s_1'', \ldots, s_{d-1}''\}$ representing S_i is obtained. The perturbation mechanism is defined by

$$\Pr\left(s_j^* = x \mid s_j''\right) = \begin{cases} \frac{2 \cdot s_j'' \cdot (e^{\varepsilon_d} - 1) + w_j \cdot (e^{\varepsilon_d} + 1)}{2 \cdot w_j \cdot (e^{\varepsilon_d} + 1)}, & \text{if } x = \frac{w_j}{2} \cdot \frac{e^{\varepsilon_d} + 1}{e^{\varepsilon_u} - 1} + m_j, \\ -\frac{2 \cdot s_j'' \cdot (e^{\varepsilon_d} - 1) + w_j \cdot (e^{\varepsilon_d} + 1)}{2 \cdot w_j \cdot (e^{\varepsilon_d} + 1)}, & \text{if } x = -\frac{w_j}{2} \cdot \frac{e^{\varepsilon_d} + 1}{e^{\varepsilon_d} - 1} + m_j. \end{cases} \tag{3}$$

Since a range move was performed on S_i' before the perturbation, m_j is added to the perturbation result x in Eq. 3 to restore the data. After completing the perturbation, send S_j^* to the server, which gets all the perturbation data and calculates the mean value of each dimension of the data.

The overall process of user data perturbation is shown in Algorithm 1, where S_i^* is obtained according to Eq. 3 perturbation and then sent to the collector server. The privacy proof of the Algorithm 1 is described in Sect. 3.3.

Algorithm 1. User data perturbation.

Require: privacy budget ε_u, security range $G_i = \{g_0, g_1, \ldots, g_{d-1}\}$, user u_i data vector $S_i = \{s_0, s_1, \ldots, s_{d-1}\}, 0 < i < n, 0 < j < d$

Ensure: user u_i data vector after perturbation $S_i^* = \{s_0^*, s_1^*, \ldots, s_{d-1}^*\}$

1: S_i is normalized to obtain $S_i' = \{s_0', s_1', \ldots, s_{d-1}'\}$
2: S_i' range moves to obtain $S_i'' = \{s_0'', s_1'', \ldots, s_{d-1}''\}$
3: **for** $j \leftarrow 0 \ldots d-1$ **do**
4: $\quad w_j = |g_j|$
5: $\quad m_j$ is the centroid of g_j
6: $\quad p \leftarrow$ Bernoulli $\left(\frac{2 \cdot s_j'' \cdot (e^{\varepsilon_d} - 1) + w_j \cdot (e^{\varepsilon_d} + 1)}{2 \cdot w_j \cdot (e^{\varepsilon_d} + 1)}\right)$
7: $\quad$ **if** $p = 1$ **then**
8: $\quad\quad s_j^* = \frac{w_j}{2} \cdot \frac{e^{\varepsilon_d} + 1}{e^{\varepsilon_d} + 1} + m_j$
9: $\quad$ **else**
10: $\quad\quad s_j^* = -\frac{w_j}{2} \cdot \frac{e^{\varepsilon_d} + 1}{e^{\varepsilon_d} + 1} + m_j$
11: **return** S_i^*

Centroid Perturbation. As shown in Fig. 1, the local side enters the iterative process after the user data perturbation is completed. The centroids from the server are first accepted, then iteration centroids are calculated based on the real user data. Although the server cannot infer privacy information from the user data, the clustering information of the user belonging to that cluster, i.e., the iteration centroids sent to the server in each iteration, may reveal user privacy. Because the clusters to which users belong are calculated from real data, over multiple iterations, the server can infer the approximate distribution or exact value of the user data as the clusters to which users belong change, and the iteration centroids are updated. For example, assuming the user data is two-dimensional location data, the user can be positioned in a circular region in each iteration. In multiple iterations, overlapping these circular regions will help the server locate the user's exact location or exact location range.

To address the problem that iterative centroids may cause user privacy leakage, the iterative centroid perturbation algorithm is proposed in this paper to

generate perturbed iterative centroids using a random perturbation mechanism. Also, The centroids in the first few iterations of the clustering change greatly, while the centroids in the last few iterations change only a little. Suppose the privacy budget is distributed equally, i.e., given the same amount of noise in each round. In that case, it will cause the problem of poor clustering utility or failure to converge. A privacy budget allocation mechanism in which the privacy budget for each round is incremented with the number of iterations is proposed in this paper. i.e., a smaller privacy budget is used for the first few rounds to add a larger noise. As the number of iterations increases, the privacy budget is incremented, and the noise is gradually reduced. The iterative centroid perturbation algorithm is described in Algorithm 2.

Algorithm 2. Iterative centroid perturbation algorithm.

Require: privacy budget ε_u, user's clustering centroid c_i, cluster centroid set C, the maximum number of iterations L, number of current iterations l_c, number of centroids k

Ensure: centroid after perturbation c_i^*

1: Generate a privacy budget allocation sequence by $P(n) = 2 \cdot P(n-1)\left(2 < n < L, P(0) = P(1) = \frac{1}{2^{L-1}} \cdot \varepsilon_u\right)$
2: $\varepsilon_n = P\left(l_c\right)$
3: $p \leftarrow \text{Bernoulli}\left(\frac{e^{\varepsilon_n}}{e^{\varepsilon_n}+k-1}\right)$
4: **if** $p = 1$ **then**
5: $\quad c_i^* = c_i$
6: **else**
7: $\quad c_i^* \leftarrow$ random sample from $\{C/c_i\}$
8: **return** c_i^*

As shown in Algorithm 2, the K-Randomized Response (K-RR) is used to perturb the user clustering information. Since K-RR can be applied to multivariate perturbations, there is no need to encode the centroids. The privacy budget allocation algorithm is inspired by the Fibonacci sequence. Since the goal of budget allocation is to construct an allocation scheme that increase by degrees and sums to ε, a privacy budget allocation sequence is constructed in this paper. Assuming that there are L iterations and the recursive formula for the sequence is as follows, $P(n) = 2 \cdot P(n-1)\left(2 < n < L, P(0) = P(1) = \frac{1}{2^{L-1}} \cdot \varepsilon_u\right)$, e.g. L=5, then we have a privacy budget allocation sequence $P = \left\{\frac{1}{16} \cdot \varepsilon_u, \frac{1}{16} \cdot \varepsilon_u, \frac{1}{8} \cdot \varepsilon_u, \frac{1}{4} \cdot \varepsilon_u, \frac{1}{2} \cdot \varepsilon_u\right\}$, the privacy budget for the third iteration is $\varepsilon_3 = \frac{1}{8} \cdot \varepsilon_u$ and the sum is ε_u. The iteration centroid perturbation is shown in the following Eq. 4.

$$\Pr\left[c_i^* = c_i\right] = \begin{cases} p = \frac{e^{\varepsilon_n}}{e^{\varepsilon_n}+k-1} & \text{if } c_i^* = c_i \\ q = \frac{1}{e^{\varepsilon_n}+k-1} & \text{if } c_i^* \neq c_i \end{cases}. \tag{4}$$

where ε_n represents the privacy budget for the current number of iterative rounds, L is the maximum number of iterative rounds, k is the number of centroids and c_i^* represents the iteration centroid after perturbation. The detailed

procedure for the iterative centroid perturbation algorithm is described in Algorithm 2.

Collector Server Method

Initial Centroid Selection. The server randomly generates k d-dimensional initial centroids C based on the S_i^* and sent them to the user.

Aggregation and Centroid Computation. The server groups the user perturbation data S_i^* according to the perturbation centroid $C^* = \left\{c_0^*, c_0^*, \ldots, c_{K-1}^*\right\}$ sent from the local side. In each cluster, the mean of each dimension of S_i^* is calculated separately, and the centroid C is updated in this way.

$$c_i = \frac{1}{|c_i^*|} \cdot \left\{ \sum_{S_i^* \in c_i^*} s_0^*, \sum_{S_i^* \in c_i^*} s_1^*, \ldots, \sum_{S_i^* \in c_i^*} s_{d-1}^* \right\} \tag{5}$$

where c_i is the new centroid updated by the calculation, $|c_i^*|$ is the number of user data belonging to c_i^*, and $\sum_{S_i^* \in c_i^*} s_j^*$ is the intra-class sum of the j-th dimensional data. Send the new centroid to the local side after the calculation is completed. Clustering iterations are performed as described above until the clustering is complete. The main steps of the centroid update are shown in Algorithm 3.

Algorithm 3. Centroid update algorithm.

Require: centroid after perturbation C^*, user u_i data vector after perturbation S_i^*, number of centroids k

Ensure: centroid after update C

1: **for** $i \leftarrow 0 \ldots k-1$ **do**

2: $\quad c_i = \frac{1}{|c_i^*|} \cdot \left\{ \sum_{S_i^* \in c_i^*} s_0^*, \sum_{S_i^* \in c_i^*} s_1^*, \ldots, \sum_{S_i^* \in c_i^*} s_{d-1}^* \right\}$

3: **return** $C = \{c_0, c_1, \ldots, c_{k-1}\}$

3.3 Privacy Analysis

This section proves that Algorithms 1 and 2 satisfy the definition of differential privacy and further proves that the overall framework satisfies the definition of differential privacy.

Theorem 1. *Algorithm 1 provides* (G_i, ε_u)*-PLDP for each uesr* u_i *with* (G_i, ε_u).

Proof. For any two values $s'_{j1}, s'_{j2} \in g_j$ and $s_j^* \in \left\{ \frac{w_j}{2} \cdot \frac{e^{\varepsilon_u}+1}{e^{\varepsilon_u}-1} + m_j, -\frac{w_j}{2} \cdot \frac{e^{\varepsilon_u}+1}{e^{\varepsilon_u}-1} + m_j \right\}$, there is $s''_{j1} = s'_{j1} - m_j$, $s''_{j2} = s'_{j2} - m_j$. Then there is

$$\begin{aligned} \frac{\Pr\left(s_j^* \mid s'_{j1}\right)}{\Pr\left(s_j^* \mid s'_{j2}\right)} &= \frac{\Pr\left(s_j^* \mid s''_{j1}\right)}{\Pr\left(s_j^* \mid s''_{j2}\right)} \\ &= \frac{\frac{2 \cdot s''_{j1} \cdot (e^{\varepsilon_d}-1) + w_j \cdot (e^{\varepsilon_d}+1)}{2 \cdot w_j \cdot (e^{\varepsilon_d}+1)}}{\frac{2 \cdot s''_{j2} \cdot (e^{\varepsilon_d}-1) + w_j \cdot (e^{\varepsilon_d}+1)}{2 \cdot w_j \cdot (e^{\varepsilon_d}+1)}}. \end{aligned} \tag{6}$$

or

$$\frac{\Pr\left(s_j^* \mid s'_{j1}\right)}{\Pr\left(s_j^* \mid s'_{j2}\right)} = \frac{-\frac{2\cdot s''_{j1}\cdot(e^{\varepsilon_d}-1)+w_j\cdot(e^{\varepsilon_d}+1)}{2\cdot w_j\cdot(e^{\varepsilon_d}+1)}}{-\frac{2\cdot s''_{j2}\cdot(e^{\varepsilon_d}-1)+w_j\cdot(e^{\varepsilon_d}+1)}{2\cdot w_j\cdot(e^{\varepsilon_d}+1)}}. \tag{7}$$

Using Eq. 6 as an example,

$$\frac{\Pr\left(s_j^* \mid s'_{j1}\right)}{\Pr\left(s_j^* \mid s'_{j2}\right)} = \frac{2\cdot s''_{j1}\cdot(e^{\varepsilon_d}-1)+w_j\cdot(e^{\varepsilon_d}+1)}{2\cdot s''_{j2}\cdot(e^{\varepsilon_d}-1)+w_j\cdot(e^{\varepsilon_d}+1)}. \tag{8}$$

It can be seen from Eq. 8 that when $s''_{j1} = \frac{w_j}{2}, s''_{j2} = -\frac{w_j}{2}$ $\left(s''_{j1} = -\frac{w_j}{2}, s''_{j2} = \frac{w_j}{2}\right)$, Eqs. 6 and 7 to obtain the maximum value,

$$\frac{\Pr\left(s_j^* \mid s'_{j1}\right)}{\Pr\left(s_j^* \mid s'_{j2}\right)} \leq e^{\varepsilon_d}. \tag{9}$$

Algorithm 1 satisfies (g_j, ε_d)-PLDP by Eq. 9. Since $S'_i = \left\{s'_0, s'_1, \ldots, s'_{d-1}\right\}$, $G_i = \{g_0, g_1, \ldots, g_{d-1}\}$, and $\varepsilon_d = \frac{\varepsilon_u}{d}$, $\sum_{j=0}^{d-1} \varepsilon_d = \varepsilon_u$. According to Property 1 of Sect. 2, differential privacy has the sequence combinability property. So for user u_i with (G_i, ε_u), Algorithm 1 satisfies (G_i, ε_u)-PLDP.

Theorem 2. *Algorithm 2 provides ε_u-LDP for each uesr u_i throughout the clustering process.*

Proof. For any two values $c_{i1}, c_{i2}, c_i^* \in C$ there is

$$\begin{aligned}\frac{\Pr\left[c_{i1} = c_i^*\right]}{\Pr\left[c_{i2} = c_i^*\right]} &= \frac{\frac{e^{\varepsilon_n}}{e^{\varepsilon_n}+k-1}}{\frac{1}{e^{\varepsilon_n}+k-1}} \\ &= e^{\varepsilon_n}.\end{aligned} \tag{10}$$

Algorithm 2 satisfies ε_n-LDP. For all iterations, the Property 1 sequence combinability property of Sect. 2 is applied. Since $\sum_{l_c=1}^{L} \varepsilon_n = \varepsilon_u$, for the whole clustering process, Algorithm 2 satisfies ε_u-LDP.

4 Experimental Evaluation

In this section, experiments are designed to investigate the improvements in the proposed framework compared to the existing DP k-means algorithm and how the relevant parameters influence the utility of the proposed framework.

4.1 Experimental Environment and Datasets

The hardware platform for this experiment uses Intel Core i7-11700 CPU @ 2.50 GHz, and 32.00 GB RAM. The experimental platform uses python 3.7. Two databases from the UCI dataset were used for the experiments. The Blood dataset records 748 individual blood donations from the Blood Transfusion Service Centre in Hsinchu city. Each record has five attributes. The Adult dataset is a dataset extracted from the 1994 census database. There are 488,42 records with 14 attributes per record. In this paper, six numerical attributes are retained for each record.

4.2 Experimental Setup and Evaluation Metrics

This paper focuses on three aspects of experimenting with the proposed framework.

1) Compare the utility of clustering with existing algorithms [9,10] for uniform k values under different ε. To the best of our knowledge, this paper is the first attempt at adopting PLDP in k-means clustering, so the extant advanced DP k-means algorithm was selected for comparison with the algorithm proposed in this paper. The PLDP k-means algorithm proposed in this paper is compared with the DPLloyd-Impr algorithm [9], and the DP-KCCM algorithm [10]. The DPLloyd-Impr algorithm completes the initial centroid selection by an initial centroid selection algorithm and then adds the Laplace noise to each round on average. In the DP-KCCM algorithm, the privacy budget allocation algorithm and the cluster merging algorithm are combined to enhance the clustering utility, and noise is injected through the Laplace mechanism. It is worth noting that both of these algorithms are based on differential privacy mechanisms and do not prevent attacks by malicious servers.
2) Compare the effects of different setting on the utility of clustering. Two sets of experiments are set up to understand the impact of key mechanisms on the utility of clustering. Firstly, the effect of privacy budget allocation methods on clustering utility was explored. Secondly, experiments were conducted on the effect of the iterative centroid perturbation algorithm on the clustering model.
3) Comparing the effect of different parameter distributions on clustering utility. Users can set their privacy budget ε_u and the size of the security range w_j according to their privacy needs in PLDP. All users' privacy parameters cannot be the same in practical applications. To understand the effect of key parameters on clustering utility, three sets of experiments were set up to investigate the effect of different parameters and different distributions on clustering utility.

In this paper, the utility of clustering is assessed using the Normalised Intra-Cluster Variance (NICV) [9]. The essential goal of the k-means algorithm is to divide the data into k clusters based on minimizing the error function, with distance as the evaluation metric. Therefore NICV can directly reflect the utility of clustering, while the NICV value can also reasonably reflect the impact of privacy protection mechanisms on the utility of clustering. The smaller NICV value means the better utility of clustering. NICV is defined as follows,

$$NICV = \frac{1}{N} \sum_{i=1}^{k} \sum_{S_i' \in c_i} \left\| S_i' - c_i \right\|^2 \tag{11}$$

where N represents the total number of users, k represents the number of centroids, S_i' represents the user data S_i normalized to [−1,1], and $S_i' \in c_i$ represents the centroid c_i is the closest centroid to S_i'.

4.3 Experimental Analysis

The results of the experiments are shown below. The first experiment explores the performance of two existing algorithms [9,10] and the PLDP k-means proposed in this paper under different privacy budgets ε. The data were normalized to $[-1,1]$, w_j was set to 0.1, and the maximum number of iterations was set to 12. For comparison purposes, this experiment will unify the privacy budget ε_u and w_j for users. As seen in Fig. 2, PLDP k-means performs better than DPLloyd-Impr [9] and performs similarly to DP-KCCM [10]. However, the algorithm proposed in this paper does not require a trusted third-party server, which means that the PLDP k-means algorithm can obtain a similar or better clustering utility while eliminating the risk of malicious servers.

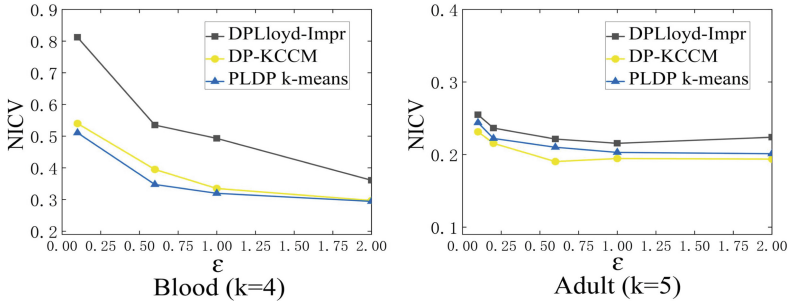

Fig. 2. Performance with respect to ε.

The second experiment explored the effect of different privacy budget allocation methods on the utility of clustering. A privacy budget allocation sequence is designed in this paper. Such that the allocation of privacy budgets in iterations presents a increase by degrees tendency. As shown in Fig. 3, the average privacy budget allocation method and the proposed allocation method were compared. It can be seen that the proposed method in this paper is significantly better than the average method. This demonstrates that the proposed privacy budget allocation algorithm in this paper can further improve the utility of clustering.

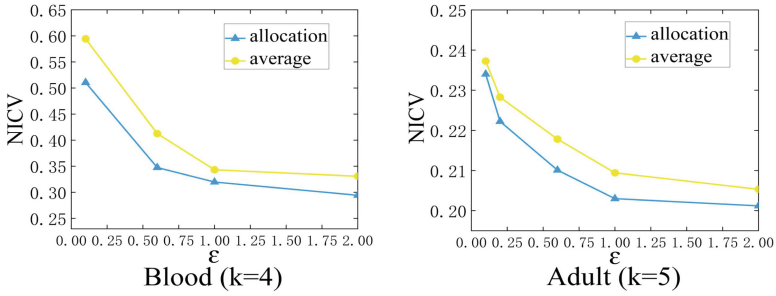

Fig. 3. Performance with respect to privacy budget allocation methods.

Iterative centroid perturbation algorithms are proposed to prevent privacy leakage caused by inference attacks. To evaluate the impact of this algorithm on the utility of clustering, a comparison experiment was conducted between using the iterative centroid perturbation algorithm and using real centroids directly. As shown in Fig. 4, the use of true centroids performed better than the use of the iterative centroid perturbation algorithm. This illustrates that some usability is sacrificed to improve privacy protection.

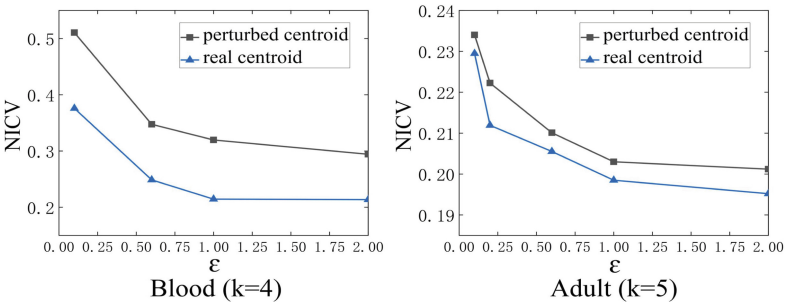

Fig. 4. Performance with respect to iterative centroid perturbation algorithm.

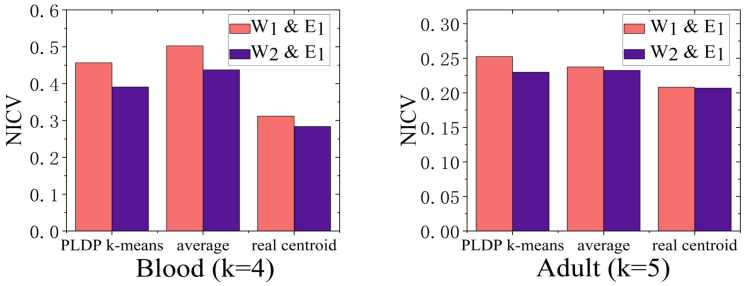

Fig. 5. Performance under the different W and the same E.

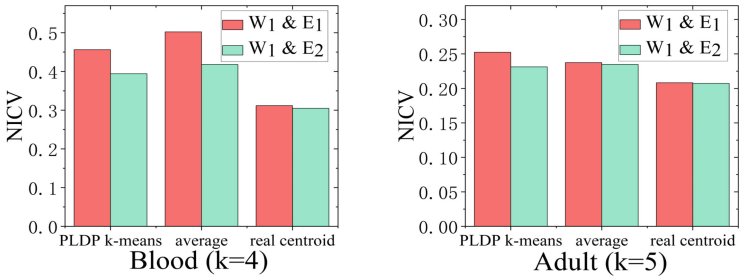

Fig. 6. Performance under the different E and the same W.

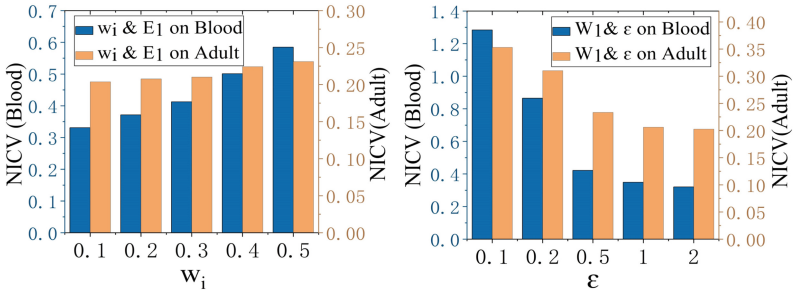

Fig. 7. Performance with respect to w_j and ε under the W_1,E_1.

The effect of the parameters is next explored. A fixed range is specified, $w_j \in[0.1,0.5]$, $\varepsilon_u \in[0.1,2]$. Each user can take their parameters from the range. Suppose the distributions of w_j, ε_u are uniform (W_1, E_1) or normal (W_2, E_2) respectively. W_1 and W_2, E_1 and E_2 have equal means. E_2 and W_2 have standard deviations of 0.3 and 0.1, respectively. PLDP k-means algorithm and the two variants of the algorithm based on this section discussed above, average, real centroid, were used for testing. Figure 5 (Fig. 6) shows the results of the three algorithms at different $W(E)$ and the same $E(W)$ on the two datasets. The control variables method shows that the results for W_2 and E_2 are better than those for W_1 and E_1, respectively. Although the means of the two distributions are equal, the normally distributed data are distributed with a high probability around the mean and a lower probability for smaller ε_u and larger w_j, which leads to better NICV values.

The effects of w_j and ε_u were further explored. The influence of varying w_j, ε_u on the utility of clustering was explored for the W_1,E_1 cases, respectively. As illustrated in Fig. 7, a larger w_j (ε_u) results in the poorer (better) utility of clustering.

Based on the experimental analysis above, the proposed algorithm in this paper improves the utility of clustering while ensuring the strength of privacy protection, and the experiments illustrate that the desired effect is achieved.

5 Conclusion

A clustering framework based on the PLDP k-means and an iterative centroid perturbation algorithms is proposed in this paper. This framework not required trusted third-party servers, and users are allowed to personalize their privacy requirements by the proposed PLDP k-means algorithm. An iterative centroid perturbation algorithm is also proposed that refines the privacy-preserving scheme by perturbing the centroids in the iterative process. Experimental results show that the proposed algorithm in this paper has better or similar performance than the extant DP k-means algorithm. Besides, the PLDP k-means algorithm requires only one upload of perturbation data, unlike the DP k-means algorithm, but the computational and communication costs during the iteration are

still nonnegligible. Future work is to analyze and reduce the computational and communication costs of the PLDP k-means algorithm.

Acknowledgements. This research was supported by the National Natural Science Foundation of China under Grant 61801131, and Guangxi Natural Science Foundation under Grant 2022GXNSFAA035632.

References

1. Wang, X., Yang, L.T., Song, L., Wang, H., Ren, L., Deen, M.J.: A tensor-based multiattributes visual feature recognition method for industrial intelligence. IEEE Trans. Ind. Informatics **17**(3), 2231–2241 (2021)
2. Liu, Q., Tian, Y., Wu, J., Peng, T., Wang, G.: Enabling verifiable and dynamic ranked search over outsourced data. IEEE Trans. Serv. Comput. **15**(1), 69–82 (2022)
3. Wang, S., Sun, Y., Bao, Z.: On the efficiency of k-means clustering: evaluation, optimization, and algorithm selection. In: Proceedings of the VLDB Endowment, pp. 163–175 (2020)
4. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_1
5. Dwork, C.: Differential privacy: a survey of results. In: Agrawal, M., Du, D., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 1–19. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79228-4_1
6. Xiao, Y., Xiong, L.: Protecting locations with differential privacy under temporal correlations. In: Proceedings of the ACM Conference on Computer and Communications Security, pp. 1298–1309. ACM, Denver (2015)
7. Su, D., Cao, J., Li, N., Bertino, E., Lyu, M., Jin, H.: Differentially private k-means clustering and a hybrid approach to private optimization. ACM Trans. Priv. Secur. **20**(4), 1–33 (2017)
8. Nguyen, T.D., Gupta, S., Rana, S., Venkatesh, S.: Privacy aware K-means clustering with high utility. In: Bailey, J., Khan, L., Washio, T., Dobbie, G., Huang, J.Z., Wang, R. (eds.) PAKDD 2016. LNCS (LNAI), vol. 9652, pp. 388–400. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31750-2_31
9. Bertino, E.: Differentially private k-means clustering. In: Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy, pp. 26–37. ACM, New Orleans (2016)
10. Ni, T., Qiao, M., Chen, Z., Zhang, S., Zhong, H.: Utility-efficient differentially private k-means clustering based on cluster merging. Neurocomputing **424**(1), 205–214 (2021)
11. Ye, Q.Q., Meng, X.F., Zhu, M.J., Huo, Z.: Survey on local differential privacy. J. Softw. **29**(7), 1981–2005 (2018)
12. Erlingsson, Ú., Pihur, V., Korolova, A.: RAPPOR: randomized aggregatable privacy-preserving ordinal response. In: Proceedings of the ACM Conference on Computer and Communications Security, pp. 1054–1067. ACM, Scottsdale (2014)
13. Cormode, G., Jha, S., Kulkarni, T., Li, N., Srivastava, D., Wang, T.: Privacy at scale: local differential privacy in practice. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 1655–1658. ACM, Houston, TX, USA (2018)

14. Xia, C., Hua, J., Tong, W., Zhong, S.: Distributed k-means clustering guaranteeing local differential privacy. Comput. Secur. **90**(1), 1–11 (2020)
15. Gu, X., Li, M., Xiong, L., Cao, Y.: Providing input-discriminative protection for local differential privacy. In: International Conference on Data Engineering(ICDE), pp. 505–516. IEEE, Dallas, Texas (2016)
16. Chen, R., Li, H., Qin, A.K., Kasiviswanathan, S.P., Jin, H.: Private spatial data aggregation in the local setting. In: 2016 IEEE 32nd International Conference on Data Engineering, ICDE 2016, pp. 289–300. IEEE, Helsinki, Finland (2016)
17. Duchi, J.C., Jordan, M.I., Wainwright, M.J.: Minimax optimal procedures for locally private estimation. J. Am. Stat. Assoc. **113**(521), 182–201 (2018)