Privacy Aware K-Means Clustering with High Utility

Thanh Dai Nguyen(✉), Sunil Gupta, Santu Rana, and Svetha Venkatesh

Center for Pattern Recognition and Data Analytics,
Deakin University, Geelong 3216, Australia
{thanh,sunil.gupta,santu.rana,svetha.venkatesh}@deakin.edu.au

Abstract. Privacy-preserving data mining aims to keep data safe, yet useful. But algorithms providing strong guarantees often end up with low utility. We propose a novel privacy preserving framework that thwarts an adversary from inferring an unknown data point by ensuring that the estimation error is almost invariant to the inclusion/exclusion of the data point. By focusing directly on the estimation error of the data point, our framework is able to significantly lower the perturbation required. We use this framework to propose a new privacy aware K-means clustering algorithm. Using both synthetic and real datasets, we demonstrate that the utility of this algorithm is almost equal to that of the unperturbed K-means, and at strict privacy levels, almost twice as good as compared to the differential privacy counterpart.

1 Introduction

Data mining is transforming the world. The scope is enormous. Not only do institutions collect data, people too “exhume” data - from black-boxes in their cars, to Fitbits they wear, to posts on Facebook. Personal data, however, cannot be accessed freely. But we could quickly learn about dangerous road conditions if we could utilize the data from each car. In another scenario, we could give early warnings of a heart attack if we could access and integrate data from various sources such as Fitbit and hospital Electronic Medical Records data. For decades, we have protected sensitive data by barricading it. This has choked potential benefits available from data utilization. Privacy preserving data mining offers a way to be able *to utilize all the data safely.*

Privacy preserving data mining has become an active research area. There are different ways to achieve privacy. For example, Agrawal and Srikant [1] developed a privacy preserving decision tree by *perturbing data.* Another way to protect privacy is anonymization. In this approach, sensitive information like name, date of birth, social security number are removed from data. However, Sweeney showed that if an adversary has access to auxiliary information, these frameworks may be revealing [2]. She also proposed a k-anonymity framework [2] where some attributes of the data are removed such that if a record is in the database, there are at least k-1 identical records in the database. This reduces the risk of

J. Bailey et al. (Eds.): PAKDD 2016, Part II, LNAI 9652, pp. 388–400, 2016.
DOI: 10.1007/978-3-319-31750-2_31

revealing up to k records. This framework gave rise to a number of privacy preserving methods (see survey in [3]). Privacy can also be achieved using additive noise, data swapping or synthetic data [4]. These methods aim to retain useful statistical information about data while changing individual records.

An important task in data mining is *clustering*, where similar records are grouped together. Clustering has enormous applications for data explorations [5], data organization [6] and retrieval [7]. One of the most popular clustering algorithm is K-means. The need to perform clustering in a privacy aware manner has prompted researchers to develop privacy preserving K-means algorithms. Vaidya and Clifton [8] propose one such algorithm for vertically partitioned data using secure multiparty computation. A similar algorithm for horizontally partitioned data was proposed by Inan et al. [9]. In a more general work, Jagannathan and Wright extended these works for both the horizontally and vertically partitioned data [10]. All these works assume that the adversary does not have access to auxiliary information. In real world, when such assumptions do not hold, these methods may not protect privacy leading to distrust among the users about the system.

Recently, differential privacy [11] has emerged as a strong privacy preserving framework. It protects the data privacy even when an adversary has access to auxiliary information. Several machine learning and data mining models using this framework have been explored such as logistic regression [12], decision tree learning [13] and matrix factorization [14]. Differentially private K-means clustering algorithms are proposed in [15–17]. In [15], Blum et al. proposed SuLQ framework, which releases noisy answer for a query. They used K-means algorithm as an example to demonstrate the SuLQ framework. Similarly, PINQ system was proposed by McSherry [16], who provided a programming interface for privacy preserving analysis. K-means clustering has been implemented using PINQ as an example of data analysis algorithm. Su et al. [17] proposed a differentially private K-means under different settings where the learner is distrusted. Although differential privacy provides a strong guarantee on privacy, it often perturbs the output of algorithms so much that their utility drops to unacceptable levels. *The problem of developing a privacy framework that provides high utility under strong privacy guarantees is therefore still open.*

Inspired from a recent private random forest model [18], we propose a new privacy preserving framework that provides strong guarantee on privacy of each data point in the database ensuring high utility. This framework can handle arbitrary amounts of auxiliary knowledge about the database, that is, even if an adversary has access to all but a one data point, the framework still thwarts the adversary from inferring the unknown data point. We achieve this by randomizing the output of the algorithm using a well known statistical estimation technique known as bootstrap aggregation. Exploiting the randomness offered by bootstrap, our framework ensures that the variance of the error in the adversary's estimation does not reduce significantly due to the participation of a data point in the database. By ensuring that the error in estimation by the adversary is almost invariant to the inclusion/exclusion of the data point in the database, the adversary is defeated. Our framework significantly departs from differential privacy in the manner that in presence/absence of a data point, differential privacy preserves the *likelihood* of algorithm output while our framework preserves the *error variance*. By focusing directly on the estimation error for the data point, our framework is able to use significantly *smaller perturbation* in the algorithm output compared to the differential privacy.

Using our new privacy framework, we construct a novel, privacy preserving K-means algorithm. The key idea is to perturb the cluster centroids before their release. We do this by using bootstrap aggregation to compute the cluster centroids. We analyze our method theoretically, and derive bounds on the size of bootstrap ensemble to ensure the stipulated privacy. We consider two cases - when the cluster a data point belongs to is either *known* or *unknown* to the adversary. Using both synthetic and real datasets, we compare our algorithm against baselines - the conventional, non-private K-means and differentially private K-means. The results are remarkable - *at high levels of privacy, the utility of our method is almost the same as the non-private K-means*, and *at least twice as good as the differential privacy counterpart*. This is because for the same privacy level, we need to add significantly lower levels of noise compared to differential privacy - as example, the noise in our framework is almost 20 times lower for high privacy stipulated by leakage parameter ϵ less than 0.1.

In summary, our contributions are:

- A new privacy preserving framework;
- A novel privacy preserving K-means algorithm with high utility using the proposed privacy framework;
- Theoretical analysis of the proposed K-means algorithm and a derivation of the upper bound on the size of bootstrap ensemble to guarantee the requisite privacy;
- Illustration and validation of the usefulness of the proposed K-means through experiments on both synthetic and real datasets.

2 The Proposed Solution

In this section, we present a new privacy framework where our goal is to provide strong guarantee on privacy of every data point in the database while ensuring that utility of algorithms remain high. The proposed framework is capable of handling the arbitrary amount of auxiliary knowledge about the database in the sense that even if an adversary has access to all but one data point, the framework still thwarts an adversary from inferring the unknown data point. We use this new framework of privacy to develop a privacy preserving K-means clustering algorithm that has high clustering performance.

2.1 A New Privacy Framework

Let us denote by $D_N = \{x_1, x_2, ..., x_N\}, x_i \in R^d$ a dataset with N data points. Further denote by $D_{N\setminus r}$ a dataset that all the data points of D_N except a data point x_r. Next assume that $f(D_N)$ and $f\left(D_{N\setminus r}\right)$ are the *randomized* answers of a system for a statistical query about the dataset D_N and $D_{N\setminus r}$ respectively. Inspired by the strong guarantees of differential privacy framework [11,19], we demand our framework to protect the privacy of a data point x_r even when an adversary has access to data points in $D_{N\setminus r}$. Specifically, our proposed framework controls the level of privacy leakage for the data point x_r based on a pre-specified leakage parameter ϵ. In particular, the adversary's estimation of x_r derived using $f(D_N)$ is guaranteed to be only "ϵ-fraction better" than a estimate that is derived using $f\left(D_{N\setminus r}\right)$. Thus the presence of the data point x_r in the database brings only *negligible* risk on its privacy for a small value of ϵ. Assume that the variance of the error in the adversary's estimate of j-th attribute of x_r using $f(D_N)$, which is computed with data points including x_r, is denoted as $\mathcal{E}_{\text{inc}}(\hat{x}_{rj})$. Similarly, assume that the variance of the error in the adversary's estimate of j-th attribute of x_r using $f\left(D_{N\setminus r}\right)$, which is computed using all data points except x_r, is denoted as $\mathcal{E}_{\text{exc}}(\hat{x}_{rj})$. Formally, our proposed framework ensures the inequality

$$\frac{\mathcal{E}_{\text{inc}}(\hat{x}_{rj})}{\mathcal{E}_{\text{exc}}(\hat{x}_{rj})} \geq \exp(-\epsilon)\,. \tag{1}$$

In the above inequality, when the value of ϵ is 0, the *strongest level of privacy* is offered. In other words, adversary can not estimate x_r any better than an estimate that is obtained without x_r's participation in the database. As the value of ϵ is increased, the level of privacy drops. We refer to this framework as ***Error Preserving Privacy*** **(EPP)**.

2.2 Privacy Preserving K-Means Clustering

Given the dataset D_N, the K-means clustering algorithm aims to partition D_N into K disjoint sets $\{C_1, C_2, ..., C_K\}$ by minimizing the following cost function:

$$\min_{C_1,\ldots,C_K} \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - m_k\|^2 \tag{2}$$

where m_k is the centroid of cluster C_k. The most popular algorithm for K-means clustering is due to Lloyd [20]. This algorithm first randomly picks K data points and uses them to initialize the centroids $m_1, m_2, ..., m_K$. Using these centroids, the algorithm assigns a data point x_i to cluster C_k if m_k is the nearest centroid. After this assignment, each centroid m_k is re-computed by averaging all data points that belong to cluster C_k. The algorithm is iterated between these two steps until it converges or exceeds the maximum number of iterations.

We propose a new privacy preserving K-means algorithm that can cluster the data while maintaining the data privacy under our proposed privacy framework in (1). The key to achieving privacy is to use a randomization in the answer of the query such that the inequality in (1) is satisfied. In doing so, our effort should be to use a mechanism for the randomization that does not degrade the utility of the answer for intended tasks. Motivated by this idea, we use a mechanism that is based on bootstrap sampling [21] of data points. The proposed mechanism not only offers the desired randomness but also retains the high utility of the original algorithm.

Similar to the Lloyd's algorithm, our algorithm iterates between the two steps of data assignment to cluster centroids and centroid re-computation until no improvement can be made. However, in the *last iteration* of our algorithm, the centroids are estimated using bootstrap aggregation (bagging) [21]. For each cluster, it generates a bag of data points through bootstrap sampling, *i.e.* uniformly randomly sampling of data points with replacement. The number of data points in each bag remains same as that in the original cluster. For each bag, the centroid is estimated by averaging the data points. A total of B such bags are generated and the aggregate centroid is computed by averaging the centroid estimates of all B bags. A step-by-step summary of our proposed algorithm is provided in Algorithm 1.

In the following analysis, we present a theoretical analysis of our algorithm showing that as long as the number of bags B in the bootstrap aggregation are smaller than a certain upper bound, the privacy of the algorithm is maintained under the framework of (1). This means given the bootstrap-perturbed cluster centroids and the data points except x_r, the adversary can not estimate x_r significantly better than an estimate made by using the centroids that were computed without x_r. We refer to this model as ***Error Preserving Private K-means* (EPP-KM).**

2.3 The Analysis of Privacy Preserving K-Means Algorithm

Due to the randomness of bootstrapping, the adversary's estimate of unknown data point x_r is perturbed. In this section, we theoretically analyze the proposed model in the light of the adversary estimation of the unknown point. In general, we have the *two possible cases*: 'the adversary knows which cluster the unknown data point belongs to' *or* 'otherwise'.

Case-1 (The adversary knows which cluster x_r belongs to): Let us assume that the adversary knows that $x_r \in C_k$. Let us denote by N_k the number of data points in the cluster C_k and let x_{ij} be the j-th attribute value of data point $x_i \in C_k$. Using the centroid m_k and other data points of C_k, the best estimate of x_{rj} is given by:

$$\hat{x}_{rj} = N_k \times m_{kj} - \sum_{x_i \in C_k \setminus x_r} x_{ij}. \tag{3}$$

where m_{kj} is the j-th attribute of the centroid m_k. When the m_{kj} is estimated using bagging, it is a random variable. We will show that this randomness is used to preserve the privacy of x_{rj}. In (3), N_k and the sum of attributes are already known. Thus, the variance of the estimation error of $\hat{x}_{rj}$ is given by:

$$\mathcal{E}_{\text{inc}}(\hat{x}_{rj} | D_{N \setminus r}, m_k, z_r = k) = N_k^2 var(m_{kj} | D_{N \setminus r}), \tag{4}$$

where the cluster indicator variable $z_r = k$ encodes the knowledge $x_r \in C_k$. Because of the bagging ensemble used in our privacy preserving algorithm, m_{kj} is given by:

$$m_{kj} = \frac{1}{B} \times \frac{1}{N_k} \times \sum_{x_r \in C_k} \alpha_r x_{rj},$$

where α_r denotes the number of times x_r is sampled in B bags of bootstrap during the computation of m_k. Clearly, α_r is a random variable following a binomial distribution with mean B and variance $B(1 - \frac{1}{N_k})$. Therefore, the conditional variance of m_k is:

$$var(m_{kj}|D_{N\backslash r}) = \frac{var(\alpha_r)}{B^2 N_k^2}\left(\sum_{x_r \in C_k} x_{rj}^2\right) = \frac{1}{BN_k^2}(1 - \frac{1}{N_k})\left(\sum_{x_r \in C_k} x_{rj}^2\right). \quad (5)$$

Plugging (5) in (4), we have $\mathcal{E}_{\text{inc}}(\hat{x}_{rj}|D_{N\backslash r}, m_k, z_r = k) = \frac{1}{B}(1 - \frac{1}{N_k})\left(\sum_{x_r \in C_k} x_{rj}^2\right)$. To ensure that this estimation error variance follows the privacy framework in 1, the number of bootstrap bags B has to satisfy

$$B \leq \frac{(1 - \frac{1}{N_k}) \times \left(\sum_{x_r \in C_k} x_{rj}^2\right)}{\mathcal{E}_{\text{exc}}(\hat{x}_{rj}) \times \exp(-\epsilon)}. \quad (6)$$

The above bound is applicable to protect the j-th attribute of the data point x_r. Since the framework is required to protect all the attributes of all the data points in the cluster, the following needs to be satisfied

$$B \leq \min_j \frac{(1 - \frac{1}{N_k}) \times \left(\sum_{x_r \in C_k} x_{rj}^2\right)}{\mathcal{E}_{\text{exc}}(\hat{x}_{rj}) \times \exp(-\epsilon)} \quad (7)$$

We refer to this case as **EPP-KM (1).**

Case-2 (The adversary doesn't know which cluster x_r belongs to): In this case, the adversary does not have the information of the cluster membership of x_r. The unavailability of this information creates a bias in his estimation. To see this, consider the adversary model in (3). Assuming that x_r truly belongs to cluster k', the expectation of the adversary estimate is given as

$$E(\hat{x}_{rj}) = E_{z_r}(E(\hat{x}_{rj} \mid z_r)) = \pi_{k'} x_{rj}$$

where z_r is a random variable and $z_r = k$ implies that x_r belongs to cluster C_k. We use $\pi_{k'}$ to denote the probability that x_r belongs to the cluster $C_{k'}$. The probability $\pi_{k'}$ can be approximately estimated using the partition of data $D_{N\backslash r}$. Clearly, the estimate $\hat{x}_{rj}$, in this case, is biased as $E(\hat{x}_{rj}) \neq x_{rj}$. The variance of the error 2 in the estimation can be derived by *the law of total variance* as below

$$\begin{aligned}
&\mathcal{E}_{\text{inc}}(\hat{x}_{rj}|D_{N\backslash r}, m_{1:K}) \\
&= E_{z_r}\left[\text{var}(\hat{x}_{rj}|z_r, D_{N\backslash r}, m_{1:K})\right] + \text{var}_{z_r}\left[E(\hat{x}_{rj}|z_r, D_{N\backslash r}, m_{1:K})\right] \\
&= \sum_{k=1}^{K}\left[\frac{\pi_k}{B}(1 - \frac{1}{N_k})\left(\sum_{x_r \in C_k} x_{rj}^2\right)\right] + \pi_{k'}(1 - \pi_{k'}) x_{rj}^2
\end{aligned}$$

To satisfy the privacy framework in 1, the number of bootstrap bags B has to satisfy

$$B \le \frac{\sum_{k=1}^{K}\pi_k\left(1-\frac{1}{N_k}\right)\left(\sum_{x_r\in C_k}x_{rj}^2\right)}{\mathcal{E}_{\text{exc}}(\hat{x}_{rj})\times\exp\left(-\epsilon\right)-\pi_{k'}\left(1-\pi_{k'}\right)x_{rj}^2} \tag{8}$$

Once again, since the above bound should be applicable to protect all the attributes of all the data points in the cluster, the following needs to be satisfied

$$B \le \min_{j,r}\frac{\sum_{k=1}^{K}\pi_k\left(1-\frac{1}{N_k}\right)\left(\sum_{x_r\in C_k}x_{rj}^2\right)}{\mathcal{E}_{\text{exc}}(\hat{x}_{rj})\times\exp\left(-\epsilon\right)-\pi_{k'}\left(1-\pi_{k'}\right)x_{rj}^2} \tag{9}$$

We refer to this case as **EPP-KM (2).**

Algorithm 1. Error Privacy Preserving K-means algorithm

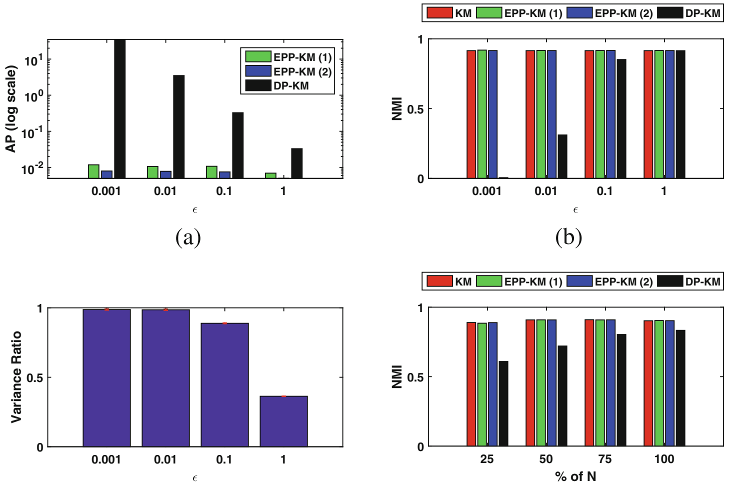
```
Input: Dataset $D = \{x_1, ..., x_N\}, x_i \in R^d$, number of clusters $K$.
Output: The bootstrap estimated cluster centroids: $m_1, ..., m_K$.
Initialization: Randomly initialize the cluster centroids $m_1, ..., m_K$.
repeat
    for each point $x_i$ do
        if $x_i$ is the closest to $m_k$ out of all centroids $m_1, ..., m_K$ then
            Assign $x_i$ to $C_k$
        end if
    end for
    for $k = 1$ to $K$ do
        Compute $m_k$ by averaging all $x_i \in C_k$
    end for
until clustering converges
for $k = 1$ to $K$ do
    Calculate the value of $B$ using (7) or (9) depending on if the adversary knows the cluster membership of data points or not.
    Compute $m_k$ using aggregation of $B$ bootstrap samples.
end for
```

3 Experiments

We experiment with a total of *three* clustering datasets: one synthetic and two real datasets. Experiments with the synthetic data illustrate the behavior of our proposed model in a controlled setting. Experiments with the real datasets show the effectiveness of our model for clustering under privacy constraints.

Baselines Methods. To evaluate the efficacy of our model, we compare its performance with the following baseline methods:

- **The Original K-means (Non-Private)**: This algorithm is the standard K-means algorithm. We note that this method does not protect privacy of database. We refer to this method as **KM**.
- **Differentially Private K-means**: This algorithm is a variant of K-means that protects the privacy of database under the framework of differential privacy [22]. In this algorithm, the j-th element of k-th K-means centroid is made ϵ-differential private by adding to it a noise η_{kj} that follows a Laplacian distribution with mean zero and standard deviation S_{kj}/ϵ where S_{kj} is the sensitivity of the j-th element of the k-th centroid. The sensitivity S_{kj} with respect to the presence/absence of any data point is approximately $\frac{1}{N_k}\max_r x_{rj}$, where N_k is the number of data points in the k-th cluster. We refer to this method as **DP-KM**.



(a) (b)

(c) (d)

Fig. 1. Results using Synthetic dataset with $N = 180, K = 3$. (a) Average perturbation in cluster centroid with respect to ϵ, (b) NMI with respect to ϵ, (c) Ratio of variance for estimation errors $\mathcal{E}_{\text{inc}}$ and $\mathcal{E}_{\text{exc}}$, (d) NMI for varying number of data points at $\epsilon = 0.1$.

Performance Measures. We use four different metrics for performance evaluation: Normalized Mutual Information (NMI) [23], Rand Index [23] and Purity [23] to evaluate the clustering performance, and Average Perturbation (AP) of privacy-preserving models to evaluate how much noise a model adds to the cluster centroids before releasing them for end use. The first three measures are widely used in clustering literature. The last evaluation measure is a normalized version of mean absolute error (MAE). Given K clusters with the original centroids $\{\mathbf{m}_k\}_{k=1}^{K}$and the perturbed centroids $\{\mathbf{m}'_k\}_{k=1}^{K}$, the average perturbation is calculated as $AP = \frac{1}{K}\sum_k \frac{\|\mathbf{m}_k - \mathbf{m}'_k\|}{\|\mathbf{m}_k\|}$.

Experimental Setting. For both synthetic and real data experiments, the clustering performance of each algorithm is studied with respect to varying privacy levels (ϵ) and the number of data points in the database. For the experiments showing clustering performance with respect to ϵ, we average the performance of each algorithm for 30 random centroid initializations for each value of ϵ. For the experiments showing clustering performance with respect to varying number of data points (N), we vary N from 25 % to 100 % of the data set size at a step of 25 %. The average performance is reported over 40 different random subsamples of size N and 20 random centroid initializations. To demonstrate the privacy guarantee of the proposed model, we estimate every data point in the database using the perturbed means and the adversary model in Eq. (3). We report the ratio of the estimation errors made by the adversary under presence/absence of the data points in the database as per our EPP framework (see Eq. (1)).

3.1 Experiments with Synthetic Data

We generate a synthetic data with 3 clusters in a 2-dimensional space. The centroids of these clusters are at $[0, 0]$, $[5, 0]$ and $[4, 4]$. For each cluster, we generate 60 random data points from a bi-variate Gaussian distribution with its mean at the cluster centroid and a standard deviation of 1 along each dimension. Our goal is to illustrate the behavior of the proposed model in terms of its clustering utility and privacy guarantees.

Figure 1 shows the experimental results for the synthetic dataset. Figure 1a compares the two cases of the proposed model with DP-KM in terms of average perturbation. As seen from the figure, DP-KM has much higher amount of perturbation compared to both EPP-KM (1) and EPP-KM (2) when ϵ is small. Figure 1b compares the proposed models with original K-means (KM) and DP-KM in terms of NMI score with respect to increasing values of ϵ. The NMI score of KM is the highest. This is not surprising as this method does not perturb the centroids and thus does not offer any privacy. However, it is interesting to note that the NMI scores of EPP-KM methods are not very different from that of KM in spite of the strong privacy guarantees offered by EPP-KM. On the other hand, DP-KM performs poorly as its NMI scores are significantly lower compared to the other methods. This poor performance of DP-KM is evident from the high levels of perturbations made by this algorithm to the cluster centroids. In Fig. 1c, we demonstrate the privacy guarantee offered by EPP-KM models. As seen from the figure, the variance of the error in an adversary's estimation for any data point changes by a factor of only $\exp(-\epsilon)$ due to its participation in the database. We can see that for low values of ϵ, e.g. when $\epsilon = 0.001$, the ratio of the error variance in the adversary's estimation is around 1, meaning that no extra reduction in uncertainty is achieved by the adversary. At the other values of ϵ, the plot follows the EPP framework of Eq. (1). We also study the effect of the number of data points in the database on the clustering performance. Figure 1d compares the NMI score of the proposed models with KM and DP-KM. For this experiment, the privacy parameter ϵ is fixed at 0.1. The performance of all the algorithms improve with the number of data points due to reduction in the perturbation. The NMI scores of EPP-KM variants are close to that of KM. Once again the performance of DP-KM is poor in the beginning as it needs high perturbations due to small cluster size.

3.2 Experiment with Real Data

We use the following datasets from UCI machine learning repository[1]:

- **Seeds dataset:** This dataset consists of 210 data points of three wheat types: *Kama*, *Rosa* and *Canadian*. Each data point has 7 geometric attributes of wheat kernels: area, perimeter, compactness, length of kernel, width of kernel, asymmetry coefficient, length of kernel groove. Our task is to use these attributes to cluster the data points in 3 different categories.
- **User Knowledge Modeling dataset (UKM):** The dataset is about student's knowledge level about a subject of Electrical DC Machines. There are 4 levels of knowledge: Very Low, Low, Middle, High. The UKM dataset has 258 data points and each data point has 5 attributes: STG, SCG, STR, LPR, PEG. Our task is to use these attributes to cluster the data points in 4 different categories.

Experimental Results. The experimental results with the Seeds dataset and the UKM dataset are shown in Figs. 2 and 3 respectively. The results follow similar patterns as in the Synthetic dataset. As seen from Figs. 2a and 3a, the average perturbations used in the centroids by both the proposed EPP-KM variants are quite small. In contrast, the average perturbation by DP-KM is extremely high for small values of ϵ. The NMI performance of the proposed EPP-KM models with respect to ϵ is approximately 0.7 and 0.3, which is close to that of KM (see Figs. 2b and 3b) while the performance of DP-KM is extremely poor at small values of ϵ and only improves at higher values of ϵ. Similar to the Synthetic dataset, Figs. 2c and 3c demonstrate that the adversary gains almost no extra information about any data point at small values of ϵ (at strict privacy).

We also study the effect of the number of data points in the database on the clustering performance. From Figs. 2d and 3d we can see that the NMI score of both EPP-KM variants are almost same as that of KM. On the contrary, the performance of DP-KM is quite poor as when using 25 % fraction of data points, NMI score of DP-KM drops to as low as 0.54 and 0.17 for Seeds and UKM dataset respectively.

A more complete set of results showing other clustering measures, in particular, Purity and Rand Index are reported in Table 1. As seen from the Table, both EPP-KM variants consistently achieve high level of clustering performance in terms of all three evaluation metrics. At times, we observed that the performances of EPP-KM (2) were slightly better than even KM. After further investigation, we found that this happens due to the robustness of bootstrap sampling to outliers [24].

[1] available at URL https://archive.ics.uci.edu/ml/datasets.html.

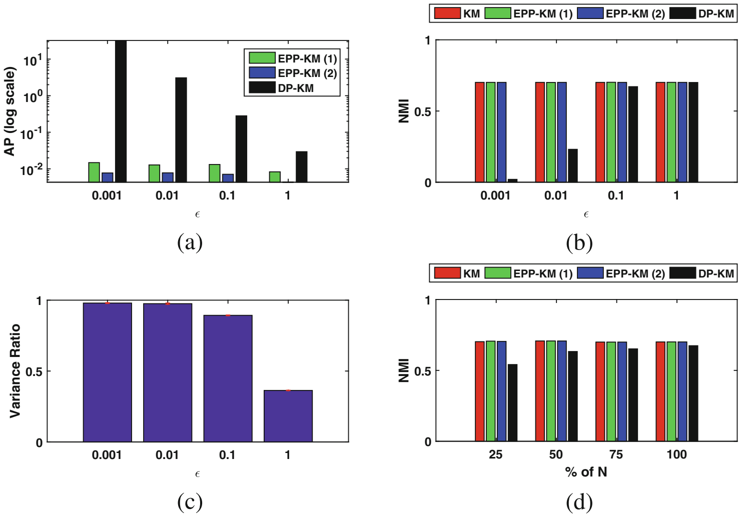


Fig. 2. Results using Seeds dataset with $N = 210, K = 3$, (a) Average perturbation in cluster centroid with respect to ϵ, (b) NMI with respect to ϵ, (c) Ratio of variance for estimation errors $\mathcal{E}_{\text{inc}}$ and $\mathcal{E}_{\text{exc}}$, (d) NMI for varying number of data points at $\epsilon = 0.1$.

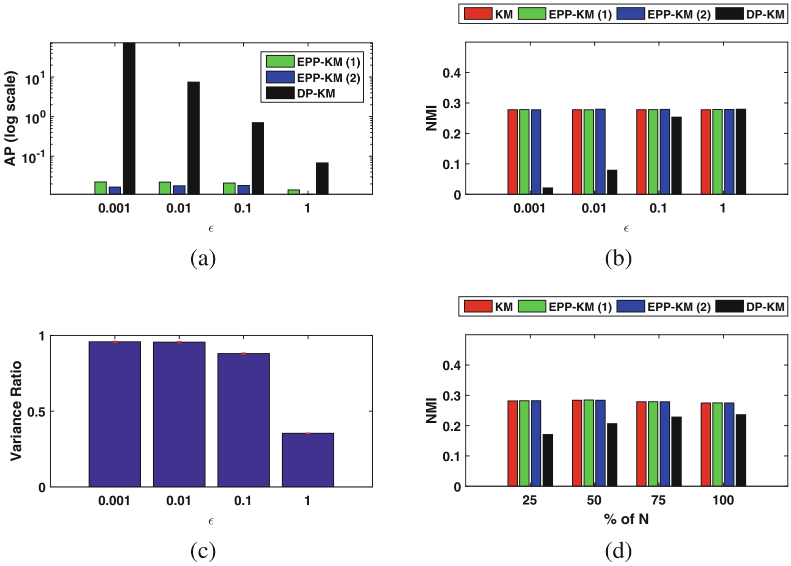


Fig. 3. Results using UKM dataset with $N = 258, K = 4$, (a) Average perturbation in cluster centroid with respect to ϵ, (b) NMI with respect to ϵ, (c) Ratio of variance for estimation errors $\mathcal{E}_{\text{inc}}$ and $\mathcal{E}_{\text{exc}}$, (d) NMI for varying number of data points at $\epsilon = 0.1$.

Table 1. Comparison with the baselines in terms of various metrics at $\epsilon = 0.1$. Average results over 30 random centroid initializations are reported with the standard errors in parenthesis. The bold face indicates the best results among private algorithms.

		Synthetic	Seeds	UKM
NMI	KM	0.9152 (0.0128)	0.7010 (0.0014)	0.2778 (0.0107)
	EPP-KM (1)	0.9160 (0.0121)	**0.7017** (0.0016)	0.2781 (0.0109)
	EPP-KM(2)	**0.9162** (0.0128)	0.7010 (0.0014)	**0.2790** (0.0106)
	DP-KM	0.8514 (0.0192)	0.6709 (0.0064)	0.2534 (0.0108)
Purity	KM	0.9707 (0.0112)	0.8933 (0.0004)	0.5683 (0.0078)
	EPP-KM (1)	0.9709 (0.0110)	**0.8938** (0.0004)	0.5686 (0.0080)
	EPP-KM (2)	**0.9711** (0.0112)	0.8933 (0.0004)	**0.5691** (0.0078)
	DP-KM	0.9446 (0.0128)	0.8759 (0.0047)	0.5536 (0.0062)
Rand index	KM	0.9669 (0.0093)	0.8732 (0.0003)	0.6819 (0.0033)
	EPP-KM (1)	0.9672 (0.0090)	**0.8736** (0.0003)	0.6820 (0.0033)
	EPP-KM (2)	**0.9674** (0.0093)	0.8732 (0.0003)	**0.6823** (0.0033)
	DP-KM	0.9368 (0.0115)	0.8559 (0.0041)	0.6642 (0.0046)
Average perturbation	EPP-KM (1)	0.0108 (0.0007)	0.0131 (0.0008)	0.0210 (0.0008)
	EPP-KM (2)	**0.0075** (0.0004)	**0.0071** (0.0004)	**0.0183** (0.0007)
	DP-KM	0.3275 (0.0227)	0.2811 (0.0176)	0.7059 (0.0433)

4 Conclusion

We proposed a novel framework for privacy preserving data mining and developed a K-means clustering algorithm under this framework. The proposed framework provides strong privacy guarantees even when an adversary has access to auxiliary knowledge about the database. Our private K-means algorithm calculates cluster centroids using bootstrap aggregation, which introduces just enough perturbation to ensure that privacy of every data point is maintained. We theoretically analyze our method and derive bounds on the size of bootstrap ensemble, which ensures the privacy under the proposed framework. The experimental results clearly show that our algorithm has high utility with strong privacy guarantees.

References

1. Agrawal, R., Srikant, R.: Privacy-preserving data mining. ACM SIGMOD Rec. **29**(2), 439–450 (2000). ACM
2. Sweeney, L.: k-anonymity: a model for protecting privacy. Int. J. Uncertainty Fuzziness Knowl. Based Syst. **10**(05), 557–570 (2002)
3. Ciriani, V., di Vimercati, S.D.C., Foresti, S., Samarati, P.: k-anonymous data mining: a survey. In: Aggarwal, C.C., Yu, P.S. (eds.) Privacy-Preserving Data Mining. Advances in Database Systems, vol. 34, pp. 105–136. Springer, US (2008)
4. Malik, M.B., Ghazi, M.A., Ali, R.: Privacy preserving data mining techniques: current scenario and future prospects. In: ICCCT 2012, pp. 26–32. IEEE (2012)

5. Begelman, G., Keller, P., Smadja, F., et al.: Automated tag clustering: improving search and exploration in the tag space. In: Collaborative Web Tagging Workshop at WWW2006, pp. 15–33 (2006)
6. Fred, A.L., Jain, A.K.: Data clustering using evidence accumulation. In: ICPR 2002, vol. 4, pp. 276–280. IEEE (2002)
7. Zeng, H.-J., He, Q.-C., Chen, Z., Ma, W.-Y., Ma, J.: Learning to cluster web search results. In: ACM SIGIR 2004, pp. 210–217 (2004)
8. Vaidya, J., Clifton, C.: Privacy-preserving k-means clustering over vertically partitioned data. In: KDD 2003, pp. 206–215. ACM (2003)
9. Inan, A., Kaya, S.V., Saygın, Y., Savaş, E., Hintoğlu, A.A., Levi, A.: Privacy preserving clustering on horizontally partitioned data. Data Knowl. Eng. **63**(3), 646–666 (2007)
10. Jagannathan, G., Wright, R.N.: Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In: KDD 2005, pp. 593–599. ACM (2005)
11. 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)
12. Chaudhuri, K., Monteleoni, C.: Privacy-preserving logistic regression. In: NIPS 2009, pp. 289–296 (2009)
13. Jagannathan, G., Pillaipakkamnatt, K., Wright, R.N.: A practical differentially private random decision tree classifier. In: ICDMW 2009, pp. 114–121. IEEE (2009)
14. Hua, J., Xia, C., Zhong, S.: Differentially private matrix factorization. In: IJCAI (2015)
15. Blum, A., Dwork, C., McSherry, F., Nissim, K.: Practical privacy: the sulq framework. In: PODS 2005, pp. 128–138. ACM (2005)
16. McSherry, F.D.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: ACM SIGMOD International Conference on Management of Data (2009)
17. Su, D., Cao, J., Li, N., Bertino, E., Jin, H.: Differentially private k-means clustering. CoRR, abs/1504.05998 (2015)
18. Rana, S., Gupta, S., Venkatesh, S.: Differentially private random forest with high utility. In: IEEE International Conference on Data Mining (2015)
19. Dwork, C.: Differential privacy: a survey of results. In: Agrawal, M., Du, D.-Z., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 1–19. Springer, Heidelberg (2008)
20. Lloyd, S.P.: Least squares quantization in PCM. IEEE Trans. Inf. Theor. **28**(2), 129–137 (1982)
21. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. CRC Press, Boca Raton (1994)
22. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006)
23. Manning, C.D., Raghavan, P., Schütze, H., et al.: Introduction to Information Retrieval, vol. 1. Cambridge University Press, Cambridge (2008)
24. Salibian-Barrera, M., Zamar, R.H.: Bootstrapping robust estimates of regression. Ann. Stat. **30**, 556–582 (2002)