Differentially Private Clustering via Maximum Coverage

Matthew Jones, Huy L. Nguyen, Thy D Nguyen
Northeastern University
{jones.m, hu.nguyen, nguyen.thy2}@northeastern.edu

Abstract

This paper studies the problem of clustering in metric spaces while preserving the privacy of individual data. Specifically, we examine differentially private variants of the k-medians and Euclidean k-means problems. We present polynomial algorithms with constant multiplicative error and lower additive error than the previous state-of-the-art for each problem. Additionally, our algorithms use a clustering algorithm without differential privacy as a black-box. This allows practitioners to control the trade-off between runtime and approximation factor by choosing a suitable clustering algorithm to use.

1 Introduction

Clustering is an important routine in many machine learning tasks, such as image segmentation (Yang et al. 2017; Patel, Van Nguyen, and Vidal 2013), collaborative filtering (McSherry and Mironov 2009; Schafer et al. 2007), and time series analysis (Mueen and Keogh 2012). Thus, improving the performance of private clustering has a great potential for directly improving other private machine learning tasks. Ideally, we are looking to achieve differential privacy, a privacy definition with strong theoretical support which requires that the algorithm be insensitive to small changes in the dataset (Dwork, Roth et al. 2014). As practical evidence of the significance of differentially private clustering, differential privacy has been accepted by several large tech corporations such as Google, Apple, and Microsoft (Álvarez Marañón 2020) and used in several clustering applications including clustering network data (Ni et al. 2018), clustering for facial recognition (Chamikara et al. 2020), and developing intelligent electrical service through user data clustering (Xiong et al. 2018). Hence, the motivation to study clustering in the differential privacy framework has sources in both strong technical and practical reasons. We begin with closely related works and our results, first in k-medians and then in k-means.

The k-medians problem has been studied extensively in the literature. Previous work in Kariv and Hakimi (1979) proved that k-medians is NP-hard. There is a long line of works on approximation algorithms for k-medians without differential privacy (Chrobak, Kenyon, and Young 2006; Arya et al. 2004; Charikar et al. 2002; Jain and Vazirani 2001; Jain et al. 2003; Byrka et al. 2017). The state of the art is a $2.675 + \epsilon$ approximation by Byrka et al. (2017). For the differentialy private k-medians problem, Gupta et al. (2010) proposed a polynomial time $(\epsilon_p, 0)$-differentially private algorithm with cost at most $6\mathsf{OPT} + O(k^2 \log^2 n/\epsilon_p)$. Kaplan and Stemmer (2018) states a variant of the algorithm for (ϵ_p, δ_p)-differential privacy with cost

$$O\Big(\mathsf{OPT} + \frac{\Delta k^{1.5}}{\epsilon_p} \log \frac{n}{\beta} \sqrt{\log n \cdot \log(1/\delta_p)}\Big)$$

where β is the failure probability and $\Delta = \max_{(u,v)\in V^2} d(u,v)$ is the diameter of the metric space. We denote by OPT the optimal objective cost of the non-private clustering problem in question, and n and k denote the number of points available for choice as cluster centers and the number of clusters respectively.

In contrast with the non-private setting, any result for differentially private clustering requires additive error in addition to a multiplicative factor on the optimal clustering cost. In particular for the k-medians problem, Theorem 4.5 in Gupta et al. (2010) shows that any private algorithm needs at least $\Omega(\Delta \cdot k \ln(n/k)/\epsilon_p)$ additive error. In fact, their proof shows there exists a family of k-medians instances such that the optimal objective cost is 0 but every ϵ_p-differentially private algorithm must incur a cost $\Omega(\Delta \cdot k \ln(n/k)/\epsilon_p)$. This lower bound can be extended almost as is to (ϵ_p, δ_p)-DP with $\delta_p = 1/poly(n)$. From a practical point of view, the additive error is the main driver dictating the performance of algorithms as can be seen from the experimental results in Balcan et al. (2017). In Section 7 of Balcan et al. (2017) the experiments show that increasing the number of centers reduces the clustering cost for the non-private k-means algorithm significantly, whereas the clustering cost stays relatively stable for the private counterpart. This behavior is observed in both synthetic and real datasets and both for their algorithm and SuLQ k-means. These results show strong evidence that the additive error is both a limiting factor objectively and an important key to improving private clustering algorithms.

Motivated by the insights from Gupta et al. (2010) and Balcan et al. (2017), our main contributation is a new (ϵ_p, δ_p)-differentially private k-medians algorithm whose time is polynomial and which has a constant multiplicative

factor and improved additive error:

$$O\left(\mathsf{OPT} + \frac{k\Delta}{\epsilon_p}\log n\left(\log\left(\frac{e}{\delta_p}\right)\right)\right).$$

Note that our additive error is linear in k and almost matches the lower bound of Gupta et al. (2010) up to lower order terms. In comparison, the best previous upper bound due to Gupta et al. (2010); Kaplan and Stemmer (2018) has a dependence of $k^{1.5}$. By using the non-private algorithm of Byrka et al. (2017) as a subroutine, our multiplicative factor is $6.35 + \epsilon$, which is slightly worse than 6 from Gupta et al. (2010).

The k-means problem has also been studied extensively with a long line of works on privacy-preserving approximation algorithms (Blum et al. 2005; Nissim, Raskhodnikova, and Smith 2007; Feldman et al. 2009, 2017; Balcan et al. 2017; Kaplan and Stemmer 2018). As an extension of our techniques, we also provide an (ϵ_p, δ_p)-differentially private algorithm for the Euclidean k-means problem with constant multiplicative error and better additive cost compared to previous work (see Table 1). Our algorithm has cost:

$$O\big(\mathsf{OPT} + k\log n + d^{0.51}(\log\log n)^{2.53}\,(k\log k)^{1.01}\,\big)$$

Again, in this setting our additive error is almost linear in k as opposed to $k^{1.5}$ in previous work (Kaplan and Stemmer 2018).

In addition to improved performance guarantee, our k-medians and Euclidean k-means algorithms use a non-private clustering algorithm as a black-box. This allows practitioners to control the trade-off between the approximation guarantee and the runtime of the clustering algorithm and even use heuristics with good empirical performance. This is especially important since most applications use Lloyd's algorithm for k-means, which has superior practical performance compared with other methods despite its $O(\log k)$ approximation factor. For example, in the experiments of Balcan et al. (2017) they found that adding several Lloyd's iteration to their algorithms improves the experimental result, supporting the importance of such heuristics on practical performance.

Our technique is a novel approach based on Maximum Coverage, in contrast to previous approaches in differentially private clustering such as center swapping in Gupta et al. (2010); Balcan et al. (2017) and locality sensitive hashing in Nissim and Stemmer (2018); Balcan et al. (2017). The algorithms for k-medians and Euclidean k-means include two main steps. In the first step, we iterate through distance thresholds from small to large and at each threshold apply a differentially private Maximum Coverage algorithm to select centers that cover almost as many points as are covered by the optimal solution at those thresholds. For the second step, we create a new dataset based on those centers and apply a non-private clustering algorithm on this dataset by moving each demand point to its nearest center and then applying the Laplace mechanism (Dwork et al. 2006b) to report the number of points at each center. This makes sure that privacy is preserved for the new dataset and we incur no additional privacy cost when we apply the non-private clustering algorithm in the final step.

2 Related Works

Table 1 summarizes the performance of previous works in comparison with our algorithms. In the table, only the work by Gupta et al. (2010) is for $(\epsilon_p, 0)$-differential privacy, while the others are for (ϵ_p, δ_p)-differential privacy. As mentioned above, Gupta et al. (2010) gave the first private k-medians algorithm with a constant multiplicative approximation and additive error polynomial in the number of centers and logarithmic in the number of points. The algorithm uses the local search approach of Arya et al. (2004). Our algorithm, on the other hand, can be used with any non-private k-medians algorithm.

For the Euclidean k-means problem, Balcan et al. (2017) proposes the strategy of first identifying a set of potential centers with low k-means cost, then applying the techniques of Gupta et al. (2010) to find the final centers among the potential centers. However, their potential centers are only guaranteed to contain a solution with multiplicative approximation $O\left(\log^3 n\right)$. This result was improved by Kaplan and Stemmer (2018), which can construct a set of potential centers containing a solution with constant multiplicative approximation.

Another approach for the k-means problem is via the 1-cluster problem. Given a set of input points in $\mathbb{R}^d$ and $t \leq n$, the goal is to find a center that covers at least t points with the smallest radius. The work of Feldman et al. (2017) shows that the k-means problem can solved by running the algorithm for the 1-cluster problem multiple times to find several balls to cover most of data points with $O(k\log n)$ multiplicative error. Nissim and Stemmer (2018) proposed an improved algorithm for the 1-cluster problem, resulting in a differentially private k-means algorithm with $O(k)$ multiplicative error.

3 Preliminaries

Differential Privacy

Differential privacy is a privacy framework for computations run against sensitive input data sets. Its requirement, informally, is that the computation behaves similarly on two input datasets that are nearly identical. Formally,

Definition 1. *(Dwork et al. 2006a) A randomized algorithm M has δ_p-approximate ϵ_p-differential privacy, or (ϵ_p, δ_p)-differential privacy, if for any two input sets A and B with a symmetric difference which has a single element and for any set of outcomes $S \subseteq Range(M)$*

$$\Pr[M(A) \in S] \leq \exp(\epsilon_p) \times \Pr[M(B) \in S] + \delta_p.$$

If $\delta_p = 0$, we say that M is ϵ_p-differentially private. An algorithm with $(\epsilon_p, 0)$-differential privacy ensures that the output $M(A)$ is (almost) equally likely to be observed on neighboring datasets, whereas in (ϵ_p, δ_p)-differential privacy the value δ_p dictates the probability that ϵ_p-privacy fails to hold (Dwork et al. 2006a). In this way, (ϵ_p, δ_p)-differential privacy is a relaxation of ϵ_p-differential privacy. We use an error parameter ϵ for utility and we use ϵ_p and δ_p to denote parameters for differential privacy (or ϵ_s and δ_s for Algorithm 1, to differentiate privacy parameters to different algorithms).

Reference	Objective	Multiplicative Error	Additive Error
(Gupta et al. 2010), (Kaplan and Stemmer 2018)	k-medians	O(1)	$O(k \log n)^2$ $O(k \log n)^{1.5}$
Ours	k-medians	$O(1)$	$O(k \log n)$
(Feldman et al. 2017)	Euclidean k-means	$O(k \log n)$	$O\left(k\sqrt{d}\log(nd) \cdot 9^{\log^*(\lvert X\rvert\sqrt{d})}\right)$
(Balcan et al. 2017)	Euclidean k-means	$O\left(\log^3 n\right)$	$O\left(\left(k^2+d\right)\log^5 n\right)$
(Kaplan and Stemmer 2018)	Euclidean k-means	$O(1)$	$O\big((k \log(n \log k))^{1.5}$ $+d^{0.51}(\log\log n)^{2.53}(k \log k)^{1.01}\big)$
Ours	Euclidean k-means	$O(1)$	$O\big(k \log n$ $+d^{0.51}(\log\log n)^{2.53}(k \log k)^{1.01}\big)$

Table 1: Comparison of our clustering algorithms with prior works, omitting dependence on ϵ_p, δ_p.

One basic construction for differentially private algorithms is the Laplace mechanism.

Definition 2. *(L_1 sensitivity) A function $f : \mathbb{N}^{|X|} \to \mathbb{R}^k$ has L_1 sensitivty Δf if $\|f(A) - f(A')\|_1 \leq \Delta f$ for all A, A' with a symmetric difference which has a single element.*

Theorem 3. *Laplace mechanism (Dwork et al. 2006b): Let function $f : \mathbb{N}^{|X|} \to \mathbb{R}^k$ have L_1 sensitivty Δf and $\epsilon_p > 0$. Mechanism M that on input A outputs $f(A) + Lap(\frac{\Delta f}{\epsilon_p})$ is $(\epsilon_p, 0)$-differentially private, where $Lap(\frac{\Delta f}{\epsilon_p})$ denotes a random variable following Laplace distribution with scale parameter $b = \frac{\Delta f}{\epsilon_p}$.*

Another tool for construction of differentially private algorithms which we use in this work is the exponential mechanism. This construction is parameterized by a query function $q(A, r)$ mapping a pair of input data set A and candidate result r to a real value. With q and a privacy value ϵ_p, the mechanism selects an output which favors higher score values using

$$Pr[\mathcal{E}_q^\epsilon(A) = r] \propto \exp(\epsilon_p q(A, r)). \tag{1}$$

Theorem 4. *(McSherry and Talwar 2007) The exponential mechanism when used to select an output $r \in R$ gives $2\epsilon_p\Delta$-differential privacy and, letting R^* be the subset of R achieving $q(A, r) = \max_r q(A, r)$, ensures that*

$$\Pr[q(A, \mathcal{E}_q^\epsilon(A)) < \max_r q(A, r) - \ln\left(\frac{|R|}{|R^*|}\right)/\epsilon_p - t/\epsilon_p] \leq \exp(-t).$$

Maximum Coverage

Our differentially private k-medians algorithm solves the Maximum Coverage problem as a subproblem. The Maximum Coverage problem is defined as follows: on a universe $\mathcal{U}$ of items, a family $\mathcal{S}$ of subsets of $\mathcal{U}$, and a parameter z, the goal is to select z sets in $\mathcal{S}$ to cover the maximum number of elements in $\mathcal{U}$. Formally, we are looking to find

$$\arg\max_{\mathcal{C}\subseteq\mathcal{S}, |\mathcal{C}|=z} \Big|\bigcup_{c\in\mathcal{C}} c\Big|.$$

Our approach for solving private Maximum Coverage is based on the Unweighted Set Cover algorithm in Gupta et al. (2010). To preserve privacy, Algorithm 1 chooses sets using the exponential mechanism, with probability related to the improvement in coverage caused by choosing the set.

Algorithm 1: Maximum Coverage

Input: Set system $(U, \mathcal{S})$, a private set $R \subset U$ to cover, ϵ_s, δ_s, m

$i \leftarrow 1$, $R_i = R$, $\mathcal{S}_i \leftarrow \mathcal{S}$. $\epsilon' \leftarrow \epsilon_s/2\ln(\frac{e}{\delta_s})$.

for $i = 1, 2, \ldots, m$ **do**

  Pick a set S from $\mathcal{S}_i$ with probability proportional to $\exp(\epsilon'|S \cap R_i|)$.

  Output set S.

  $R_{i+1} \leftarrow R_i \setminus S$, $\mathcal{S}_{i+1} \leftarrow \mathcal{S}_i - \{S\}$.

Assume that there exists a selection of z sets that covers $\mathcal{U}$. A classic fact for the maximum coverage problem is that if we build the family $\mathcal{C}$ by always selecting the item in $\mathcal{S} \setminus \mathcal{C}$ that covers the largest number of uncovered elements in $\mathcal{U}$, then after z iterations, $|\cup_{c\in\mathcal{C}} c| \geq (1 - 1/e)|\mathcal{U}|$. Here we show an observation that will be useful for our algorithm later. The proof is in the appendix.

Lemma 5. *For $\epsilon > 0$, if we always select a set that covers at least half as many uncovered elements as the set that covers the most uncovered elements, then after $2z \ln 1/\epsilon$ iterations, $\left|\bigcup_{c\in\mathcal{C}} c\right| \geq (1 - \epsilon)|\mathcal{U}|$.*

4 Private k-medians

Given a set of points V, a metric $d : V \times V \to \mathbb{R}$, a private set of demand points $D \subseteq V$, and a value k where $k < |V| = n$, recall that the objective of the k-medians problem is to select a set of points (centers) $F \subset V$, $|F| = k$ to minimize $\text{cost}(F) = \sum_{v\in D} d(v, F)$, where $d(v, F) = \min_{f\in F} d(v, f)$. Also, recall that we use ϵ as the approximation parameter for maximum coverage problem in Lemma 5 and we use ϵ_p and δ_p as privacy parameters. Let $B_r(v)$ be the ball of radius r centered at v, i.e. the set of all points in the metric space within distance r from v.

Our approach is based on the Maximum Coverage problem. One way to compute the clustering cost is by computing for every distance threshold t the number of points within distance t from the centers and integrating the counts from 0

to the maximum distance. Thus, if for every threshold t the number of points farther than t from our solution's centers is not much more than the number of points farther than t from the optimal centers, then our cost is not much larger than the optimal cost. Thus, our algorithm goes through distance thresholds from small to large and tries to "cover" as many points as possible using fresh centers every time. For each threshold, we aim to cover $(1-\epsilon)$ times the number of points the optimal solution can cover. The result is that our clustering cost is not much larger than the optimal cost, albeit using more centers. Since we use exponentially growing thresholds, we only use a multiplicative factor $O(\log n)$ extra centers in the final set C, which results in a small error due to privacy noise. The full algorithm is described in Algorithm 2.

Algorithm 2: The k-medians algorithm

Input: a set of points V, a private set of demand points $D \subseteq V$, a metric d, ϵ, ϵ_p, δ_p

1 $V' = D, C = \emptyset, r = \lceil 1 + \log_{1+\epsilon} n \rceil$
2 **for** i *from* 1 *to* r **do**
3 Set $t_i = (1+\epsilon)^{i-1}\Delta/n$
4 Run algorithm 1 for the set system $\mathcal{U} = V, \mathcal{S} = \{B_{t_i}(v) \cap V' : v \in V\}$, with $\epsilon_s = \frac{\epsilon_p}{2}, \delta_s = \delta_p$ for $m = 2k\ln(1/\epsilon)$ iterations to get C_i
5 $V_i = \bigcup_{v \in C_i} B_{t_i}(v) \cap V'$
6 $V' = V' \setminus V_i$
7 $C = C \cup C_i$
8 Assign each point in D to its closest point $c \in C$
9 Let n_c be the total number of points assigned to c for each $c \in C$
10 For each $c \in C$, set $n'_c = n_c + Y_c$ where $Y_c \sim \text{Lap}\left(\frac{2}{\epsilon_p}\right)$
11 Run a k-medians algorithm to select and return k centers from V, with demand points at each $c \in C$ with multiplicity n'_c.

The algorithm begins with the discretization of distance thresholds. The goal is similar to our discussion above. We apply Algorithm 1 to select a set of points that covers a large set of demand points across different distance thresholds. Note that the objective cost of any set of centers following the discretization scheme is not too far from the actual costs, as we will show in Lemma 8. Thus, the set of many centers in C that we find across different thresholds t should also have cost similar to OPT.

Our final step is to obtain the final set of centers with privacy. To preserve privacy for this step, we create a new dataset similar to the original one with some privacy noise. In this new dataset, every demand point is shifted towards the closest center from C, and we apply the Laplace mechanism to the assigned number of demands points of each point in C to preserve privacy. At this point, we can use a k-medians algorithm on this dataset to output the final k centers. Although the objective in the new problem changes because of shifting and the Laplace mechanism, the set of available choices for centers V is the same. Thus, the cost of the centers returned in the final step is at most the cost of this new objective plus the total shifting distance. In the following sections, we will first formally analyze the privacy and then the utility of this algorithm.

Privacy Analysis

We first show that this algorithm is (ϵ_p, δ_p)-differentially private. To show this, we first show that the entirety of the for loop is $(\epsilon_p/2, \delta_p)$-differentially private, and then take advantage of composition and apply Theorem 3 at line 10 to obtain the final result. Note that the analysis of the loop very closely follows the proof for privacy of Unweighted Set Cover in Gupta et al. (2010); their algorithm selects sets in a particular order to form a cover, while our algorithm selects candidate centers with increasing distance thresholds where a center is assumed to cover all demand points within its distance threshold. The significant difference in the two proofs is that our algorithm could select the same center twice with different distance thresholds while a set will never be chosen twice in set cover in Gupta et al. (2010). This re-selection of the same center will not affect the differential privacy of the algorithm, because the privacy analysis hinges on which demand points have been covered, not which centers have been selected. As a result of this, we save an additional $\log n$ factor on privacy, which removes a $\log n$ term from the additive error in Lemma 9 which carries through the additive error in the utility. We include the proof in the appendix and omit it here, due to its close similarity to Gupta et al. (2010).

Lemma 6. *The for loop in Algorithm 2 preserves $(\epsilon_p/2, \delta_p)$-differential privacy.*

The function affected by the Laplace mechanism in line 11 returns a vector of the counts n_c. In the case of sets A, A' as in Theorem 3, the difference between $f(A)$ and $f(A')$ is exactly one for one item in this vector, and therefore $\|f(A) - f(A')\| = 1$, so the function has L_1 sensitivity 1. Thus, line 10 is $(\epsilon_p/2, 0)$-differentially private by Theorem 3. By composition, this fact and Lemma 6 yield the following lemma:

Lemma 7. *Algorithm 2 is (ϵ_p, δ_p)-differentially private.*

Utility Analysis

We define $t_0 = 0, t_1 = \frac{\Delta}{n}, t_2 = \frac{\Delta(1+\epsilon)}{n}, ..., t_r = \Delta$ as shorthand for the thresholds. Also, let o_i be the number of points at distance in the range $[t_{i-1}, t_i)$ from their center in the optimal solution (which we denote by OPT) and let a_i be the the number of points at distance in the range $[t_{i-1}, t_i)$ from their closest point in C after the for-loop in Algorithm 2. To bound the performance of our solution, we first show in Lemma 8 that discretizing the distance thresholds at t_i's instead of integrating from 0 to Δ introduces negligible error to the cost of the optimal solution. Next, for each distance threshold, Lemma 9 uses the approximation guarantee of maximum coverage to show that we are efficiently covering demand points using not many more centers than OPT. Crucially, Lemma 10 shows that by covering almost as well

as OPT at every distance threshold, our solution has cost not much more than that of OPT.

We begin by bounding the discretized cost of OPT.

Lemma 8. $\sum_{i=1} o_i t_i \leq (1+\epsilon)\mathsf{OPT} + \Delta$

Proof. On all $u \in D$ and set of centers F, define $d'(u, F)$ as the minimum distance threshold t_i which is larger than $d(u, F)$. If $d(u, \text{OPT}) > \frac{\Delta}{n}$ then $d'(u, \text{OPT}) \leq (1+\epsilon)d(u, \text{OPT})$ since thresholds t_i scale geometrically with factor $1+\epsilon$. If $d(u, \text{OPT}) \leq \frac{\Delta}{n}$, then $d'(u, \text{OPT}) = \frac{\Delta}{n} \leq d(u, \text{OPT}) + \frac{\Delta}{n}$. Summing over $d \in D$ yields the bound. □

Before we begin using this bound, we will first prove the effectiveness of Algorithm 1 in the context of Algorithm 2, to show that we are effectively covering demand points. For each distance threshold t_i, the following lemma shows that the algorithm covers almost as many points as OPT.

Lemma 9. *Consider iteration i of the for-loop and let M_i be the maximum coverage of k centers with radius t_i over points in V'. With high probability, at line 5:*

$$|V_i| \geq (1-\epsilon)M_i - \frac{24k \ln n \ln\left(\frac{e}{\delta_p}\right)}{\epsilon_p}$$

Proof. The items in the family $\mathcal{S}$ in line 4 are exactly one-to-one with the points in V, and the set $s_v \in \mathcal{S}$ corresponding to $v \in V$ is exactly the set of points which are not within distance t_{i-1} of an existing center in C but are within distance t_i of v. Therefore, the items covered by the centers in C_i are all within distance t_i of their closest centers, so the change in the coverage of C is at least the size of the set coverage from C_i.

Our analysis is similar to Gupta et al. (2010). The main difference is that instead of covering all points that OPT can cover, we aim to cover a $(1-\epsilon)$ portion within an additive error, hence we run $2k\log(1/\epsilon)$ iterations instead of $2k \log n$. Consider $|R_i|$ to be the number of remaining elements yet to be covered, and define $L_i = \max_{S \in \mathcal{S}} |S \cap R_i|$, the largest number of uncovered elements covered by any set in $\mathcal{S}$.

By Theorem 4, the exponential mechanism, when selecting a set, ensures that with probability at most $1/n^2$ that we select a center with coverage less than $L_i - \frac{3\ln n}{\epsilon'}$. When $L_i > \frac{6\ln n}{\epsilon'}$, we are guaranteed to choose a center that covers at least $L_i/2$ points. Based on Lemma 5, for each iteration as long as $L_i > \frac{6\ln n}{\epsilon'}$ we always take the greedy option and are guaranteed to have $|C_i| \geq (1-\epsilon)M_i$ with probability at least $1 - \frac{1}{n}$. When $L_i \leq \frac{6\ln n}{\epsilon'}$, although we are not guaranteed to take the greedy action there are at most $\frac{6k\ln n}{\epsilon'}$ points yet to be covered by the algorithm compared to OPT at radius r. Thus, the algorithm loses at most $\frac{6k\ln n}{\epsilon'}$ points. □

The next lemma relates the cost of our solution and that of OPT given that we cover almost as well as OPT at every distance threshold. Specifically, we bound the discretized cost of our coverage in terms of the discretized cost of the coverage of OPT and the additive error resulting from Algorithm 1.

Lemma 10.

$$\sum_{i=1}^{r} a_i t_i \leq \frac{1-\epsilon}{1-\epsilon-\epsilon^2}\sum_{i=1} o_i t_i + \frac{24\Delta k \ln n \ln\left(\frac{e}{\delta_p}\right)}{\epsilon_p(1-\epsilon-\epsilon^2)}$$

Proof. Let $O_i = \sum_{j=i}^{r} o_j$, $A_i = \sum_{j=i}^{r} a_j$, and $E = \frac{24k\ln n \ln\left(\frac{e}{\delta_p}\right)}{\epsilon_p}$. Given a threshold t_i, we know that the centers in OPT cover $n - O_{i+1}$ points with distance at most t_i. At threshold t_i, Algorithm 2 has already covered $n - A_i$ points so we know that there is a solution covering an additional $(n - O_{i+1}) - (n - A_i) = A_i - O_{i+1}$ points. By the guarantee of the greedy set cover algorithm in Lemma 9, we cover $a_i \geq (1-\epsilon)(A_i - O_{i+1}) - E$ new points on the next iteration. By substituting $A_i = a_i + A_{i+1}$, we have $a_i \geq \frac{(1-\epsilon)}{\epsilon}(A_{i+1} - O_{i+1}) - \frac{E}{\epsilon}$. Rearranging the last inequality to isolate A_{i+1}, notice that

$$\begin{aligned}
\sum_{i=1}^{r} a_i t_i &= \sum_{i=1}^{r} A_i(t_i - t_{i-1}) \\
&\leq \sum_{i=1}^{r}\left(\frac{\epsilon a_{i-1}}{1-\epsilon} + O_i + \frac{E}{1-\epsilon}\right)(t_i - t_{i-1}) \\
&= \sum_{i=1}^{r}\frac{\epsilon a_{i-1}}{1-\epsilon}(t_i - t_{i-1}) + \sum_{i=1}^{r} o_i t_i + \frac{\Delta E}{1-\epsilon}.
\end{aligned}$$

The last equality holds because of telescoping sums. Also, notice that

$$\begin{aligned}
\sum_{i=1}^{r}\frac{\epsilon a_{i-1}}{1-\epsilon}(t_i - t_{i-1}) &= \sum_{i=1}^{r-1}\frac{\epsilon a_i}{1-\epsilon}(t_{i+1} - t_i) \\
&\leq \sum_{i=1}^{r-1}\frac{\epsilon^2}{1-\epsilon} a_i t_i.
\end{aligned}$$

The last inequality holds because for all $1 \leq i \leq r-1$, $t_{i+1} \leq (1+\epsilon)t_i$ by definition. We are also able to drop the term $i = 0$ from the last two sums because $a_0 = 0$.

Thus,

$$\begin{aligned}
\sum_{i=1}^{r} a_i t_i - \sum_{i=1}^{r-1}\frac{\epsilon^2}{1-\epsilon} a_i t_i &= \frac{1-\epsilon-\epsilon^2}{1-\epsilon}\sum_{i=1}^{r-1} a_i t_i + a_r t_r \\
&\leq \sum_{i=1} o_i t_i + \frac{\Delta E}{1-\epsilon}
\end{aligned}$$

which implies that

$$\sum_{i=1}^{r} a_i t_i \leq \frac{1-\epsilon}{1-\epsilon-\epsilon^2}\sum_{i=1} o_i t_i + (1-\epsilon-\epsilon^2)\Delta E.$$

□

Combining the results of Lemmas 8 and 10, we see that

$$\sum_{i=1}^{r} a_i t_i \leq \frac{1-\epsilon}{1-\epsilon-\epsilon^2}\left((1+\epsilon)\mathsf{OPT} + \Delta\right) + (1-\epsilon-\epsilon^2)\Delta E$$

which gives us a bound on the cost of snapping points in D to points in C. This will be included in our final cost, which we show in the next lemma.

Lemma 11. *Consider the k-medians problem in the last line of Algorithm 2, where demand points in D are shifted to points in C and Laplace noise is applied. With high probability, the optimal objective cost of this new k-medians problem is at most*

$$\mathsf{OPT} + \sum a_i t_i + \frac{4\Delta k \ln(1/\epsilon)}{\epsilon_p}\left(\frac{\ln(n)}{\ln(1+\epsilon)} + 2\right).$$

Proof. After assigning every point in D to the closest point in C, we run a k-medians algorithm on a multiset defined by C where each element $c \in C$ has multiplicity n'_c as in line 10 of Algorithm 2. Recall that the absolute value of a random variable following a Laplace distribution with parameter b follows an exponential distribution with parameter $\frac{1}{b}$. Also note that the sum of exponential variables will be less than twice the expectation with high probability, with probability of failure decreasing exponentially in the number of summands. For the sake of completeness, we include the proof of this fact in the appendix. It is also significant that since we call Algorithm 1 a total $\left\lceil \log_{(1+\epsilon)} n + 1 \right\rceil = \left\lceil \frac{\ln n}{\ln(1+\epsilon)} + 1 \right\rceil$ times with $m = 2k\ln(1/\epsilon)$, we select at most $|C| \leq 2k\ln(n)\ln(1/\epsilon)/\ln(1+\epsilon) + 2k\ln(1/\epsilon)$ centers before calling the black-box k-medians algorithm. With all this preliminary information, we begin to prove the claim.

The last term in the bound is obtained by the Laplace mechanism, where noise is applied to the counts of each center $c \in C$. Each of the centers in C has Laplacian noise applied to it with parameter $2/\epsilon_p$. Therefore, with high probability at most $\frac{4k\ln(1/e)}{\epsilon_p}\left(\frac{\ln(n)}{\ln(1+\epsilon)} + 2\right)$ demand points are "added" by line 10 of the algorithm and each of these points is at distance at most Δ from their closest center, which yields the last term in the bound.

The first two terms come from the fact that we shift points in line 8 of Algorithm 2 and the triangle inequality. For each of the original demand points $v \in D$, let the cluster center in OPT closest to v be OPT_v, and let the point in C closest to V be denoted c_v. We see that

$$d(c_v, OPT_v) \leq d(v, c_v) + d(v, OPT_v)$$

by the triangle inequality. Since OPT of the original k-medians problem is a candidate solution for the new k-medians problem, the objective cost of using OPT upper bounds the optimal cost of the new problem. Summing over all $v \in D$, we see that the objective cost of shifted demand points is therefore bounded as

$$\sum_{v \in D} d(c_v, OPT_v) \leq \sum a_i t_i + \mathsf{OPT}$$

since $\sum a_i t_i$ is an upper-bound approximation of the cost of shifting the demand points to centers in C, and the sum of $d(v, OPT_v)$ yields exactly OPT. Thus, the first two terms come from the cost of shifting the real demand points, and the last term comes from the Laplacian noise. □

Note that there may be a better solution than the original k-medians' OPT to this new k-medians problem, but this is consistent with the bound by inequality.

Using a non-private approximation algorithm for the k-medians problem with approximation factor M in the last step of Algorithm 2, the entire cost gains a multiplicative factor M. Therefore, we can summarize the utility of Algorithm 2 into the following lemma and even simpler theorem:

Lemma 12. *With high probability, Algorithm 2 preserves (ϵ_p, δ_p)-differential privacy and solves the k-medians problem with cost*

$$O\left(M(1+\epsilon)\right)\mathsf{OPT} + O\left(\frac{Mk\Delta}{\epsilon_p}\ln n\left(\ln\left(\frac{e}{\delta_p}\right) + \frac{\ln(1/\epsilon)}{\ln(1+\epsilon)}\right)\right)$$

where the black-box k-medians algorithm used in the last step of Algorithm 2 has approximation factor M and ϵ is a small, positive constant.

Proof. Combining the results of Lemmas 8, 10, and 11, we see that with high probability the optimal cost of the k-medians problem at the final line of Algorithm 2 is given by at most

$$\left(\frac{2-\epsilon-2\epsilon^2}{1-\epsilon-\epsilon^2}\right)\mathsf{OPT} + \Big(\frac{4\Delta k\ln(1/\epsilon)}{\epsilon_p}\left(\frac{\ln(n)}{\ln(1+\epsilon)} + 2\right) + \frac{24k\ln n\ln\left(\frac{e}{\delta_p}\right)}{\epsilon_p(1-\epsilon-\epsilon^2)} + \frac{1-\epsilon}{1-\epsilon-\epsilon^2}\Big)\Delta.$$

For simplicity, we will use big-O notation going forward, so this is the same as

$$(2+O(\epsilon))\mathsf{OPT} + O\left(\frac{k\Delta}{\epsilon_p}\left(\ln(n)\frac{\ln(1/\epsilon)}{\ln(1+\epsilon)} + \ln n\ln\left(\frac{e}{\delta_p}\right)\right)\right).$$

Since the k-medians algorithm used at the last line has approximation factor M, the objective cost of the shifted k-medians problem is M times that result. To obtain the objective cost for the original problem, we see that for any demand point $v \in D$, the distance between v and a center is at most $d(c_v, v)$ and the distance from c_v to a center, by triangle inequality. Therefore, the new objective cost only adds the shifting cost on top of the modified k-medians problem's objective cost, which only slightly affects the factor in front of OPT,

$$(2M+1+(M+1)O(\epsilon))\mathsf{OPT} + O\left(\frac{Mk\Delta}{\epsilon_p}\left(\ln(n)\frac{\ln(1/\epsilon)}{\ln(1+\epsilon)} + \ln n\ln\left(\frac{e}{\delta_p}\right)\right)\right),$$

which simplifies to the claim. □

Corollary 13. *By using a constant-approximation non-private algorithm for k-medians, there is a (ϵ_p, δ_p)-differentially private algorithm for the k-medians problem that, with high probability, outputs a solution with objective cost*

$$O\left(\mathsf{OPT} + \frac{k\Delta}{\epsilon_p}\ln(n)\left(\ln\left(\frac{1}{\delta_p}\right) + \frac{\ln(1/\epsilon)}{\ln(1+\epsilon)}\right)\right).$$

Note that our algorithm allows for any k-medians algorithm to be used at the last step. One can choose a preferred trade-off between runtime and performance to select a suitable algorithm. This is in contrast to the approach in Gupta et al. (2010), where the algorithm builds on the k-median algorithm in Arya et al. (2004).

5 Application to Euclidean k-means

In the Euclidean k-means problem, instead of having a discrete set of demand points, V is defined to be all of $\mathbb{R}^d$. We wish to select a set of points (centers) $F \subset \mathbb{R}^d$, $|F| = k$ to minimize $\text{cost}(F) = \sum_{v \in D} d(v, F)^2$. In this section, we will apply our result to improve additive error in the approach in Kaplan and Stemmer (2018).

The strategy in Kaplan and Stemmer (2018) is to first identify a polynomial set of candidate centers such that it contains a subset of k candidate centers with low k-means cost. Then, the algorithm uses a private discrete k-means algorithm to select the final k-centers with low cost from the set of candidate centers. More concretely, the algorithm in Kaplan and Stemmer (2018) is guaranteed to output a (ϵ_p, δ_p)-private set of candidate centers Y of size at most $\epsilon_p n \log(\frac{k}{\beta})$ such that with probability at least $1 - \beta$, there exist a subset of size k centers with constant multiplicative error and additive error of $O(T^{\frac{1}{1-a-b}} \cdot w^{\frac{1}{1-a-b}} \cdot k^{\frac{1}{1-a-b}})\Delta^2$, where a, b are small constant parameters of the Locality Sensitive Hashing algorithm used in Kaplan and Stemmer (2018), $T = \Theta(\log \log n)$ and

$$w = O\left(\frac{\sqrt{d}}{\epsilon_p} \cdot \log \log n \cdot \log\left(\frac{k}{\beta}\right) \sqrt{\log \frac{\log \log n}{\delta_p}}\right).$$

Note that a, b can be chosen to be arbitrarily small at the cost of making the multiplicative approximation factor a larger constant. The work of Kaplan and Stemmer (2018) focuses on the regime where a, b are small and $1/(1-a-b) \leq 1.01$. The resulting additive error for identifying candidate centers is $\tilde{O}_{\epsilon_p, \delta_p}(k^{1.01} d^{.51})$.

The performance bottleneck of the Euclidean k-means is in the algorithm to select the final k centers from a candidate set. We can apply our algorithm on the potential centers returned by Kaplan and Stemmer (2018) to improve the algorithm's performance. Note that our algorithm can be applied to solve the k-means objective by passing the correct distance function. Although the square of Euclidean distance is not a metric, we can still apply the same algorithm and similar analysis to get the bound for the k-means objective. The only algorithmic difference is at the last step, where instead of running a k-medians algorithm we run a k-means algorithm to get final centers.

Rather than replicate the entire analysis section, we will only review the proofs which are directly affected by the change in the distance function. Furthermore, we will extend the proof to all distance functions d^p for any natural number p where d^2 is the distance function in the Euclidean k-means objective.

Notably, the privacy analysis is independent of the distance function, and is therefore unaffected. In fact, the only steps in the proofs of Section 4 which involve the distance function are Lemma 8, 11 and 12. For the distance function d^p, Lemma 8 is amended as follows:

Lemma 14. $\sum_{i=1} o_i t_i \leq (1+\epsilon)^p \mathsf{OPT} + \Delta$

Proof. In the case where $(d(u, \text{OPT}))^p > \frac{\Delta}{n^2}$, now $(d'(u, \text{OPT}))^p \leq (1+\epsilon)^p (d(u, \text{OPT}))^p$. Otherwise, the proof is functionally identical to that of Lemma 8. □

For Lemmas 11 and 12, we directly address and resolve the main issue the distance function faces here, which is that the triangle inequality does not hold when $p > 1$. However, we can use the following lemma, which we prove in the appendix:

Lemma 15. *In any metric space and $p \geq 1$, $(d(a,b))^p \leq 2^{p-1}((d(a,c))^p + (d(b,c))^p)$.*

Using this lemma, we see that when the triangle inequality would be applied, we gain an additional 2^{p-1} constant. This affects the leading constant of the approximation factor and additive error, but does not affect the asymptotic cost.

Therefore, the only changes to Algorithm 2 necessary to make it a functional k-means algorithm are to use the proper input metric d and to run a black-box k-means algorithm as the last step rather than a black-box k-medians algorithm. Then, if we are trying to minimize the objective function using the distance function d^p and we use a black-box algorithm for this objective function at the last step of Algorithm 2, the proofs in section 4 using Lemma 14 instead of Lemma 8 yields the following lemma and corollary:

Lemma 16. *Given a problem equivalent to k-means but with distance function d^p, and a non-private algorithm for that problem with approximation factor M, there exists an (ϵ_p, δ_p)-differentially-private algorithm for that problem which, with high probability, has objective cost at most*

$$O\left(2^{p-1} M (1+\epsilon)^p\right) \mathsf{OPT} + O\left(\frac{Mk\Delta^p}{\epsilon_p}\left(\ln n + 2^{p-1} \ln |Y| \ln\left(\frac{e}{\delta_p}\right)\right)\right).$$

Corollary 17. *There is a (ϵ_p, δ_p)-differentially private algorithm for the Euclidean k-means problem that with probability at least $1 - \beta$ returns a solution with a constant multiplicative factor and an additive error of*

$$O\left(\Delta^2\left(T^{\frac{1}{1-a-b}} \cdot w^{\frac{1}{1-a-b}} \cdot k^{\frac{1}{1-a-b}}\right) + \frac{\Delta^2 k}{\epsilon_p}\left(\ln n + \ln\left(\epsilon_p n \log\left(\frac{k}{\beta}\right)\right) \ln\left(\frac{e}{\delta_p}\right)\right)\right).$$

Note that our algorithm results in a better additive term compared to applying Gupta et al. (2010) on the potential centers. Specifically, the second additive term is almost linear in k instead of $k^{1.5}$, making the entire additive error nearly linear in k.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1909314 and 1750716. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Ethics Statement

Clustering has many applications in machine learning, such as image segmentation (Yang et al. 2017; Patel, Van Nguyen, and Vidal 2013), collaborative filtering (McSherry and Mironov 2009; Schafer et al. 2007), and time series analysis (Mueen and Keogh 2012). Privacy is a major concern when input data contains sensitive information. Differential privacy (Dwork et al. 2006b) has become a rigorous framework for ensuring privacy in algorithms. Thus, differentially private algorithms for clustering problem would ensure for each individual in the input a robust privacy guarantee.

Our improved utility guarantee will perhaps encourage adoption of privacy-preserving algorithm as a replacement for the non-private counterpart. Furthermore, our approach allows for usage with another clustering algorithm as a black-box. We believe this further improves the applicability of private clustering algorithms, making it easier to incorporate the privacy guarantee into existing clustering frameworks. The limitations of the work are that the privacy guarantee requires that the range of the data is bounded and the utility guarantee has additive error that is only meaningful when the dataset has a large enough number of participants. When applying the algorithm, the curator has to ensure that the assumptions hold to protect the privacy of the participants.

References

Arya, V.; Garg, N.; Khandekar, R.; Meyerson, A.; Munagala, K.; and Pandit, V. 2004. Local search heuristics for k-median and facility location problems. *SIAM Journal on computing* 33(3): 544–562.

Balcan, M.-F.; Dick, T.; Liang, Y.; Mou, W.; and Zhang, H. 2017. Differentially private clustering in high-dimensional euclidean spaces. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 322–331. JMLR. org.

Blum, A.; Dwork, C.; McSherry, F.; and Nissim, K. 2005. Practical privacy: the SuLQ framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 128–138.

Byrka, J.; Pensyl, T. W.; Rybicki, B.; Srinivasan, A.; and Trinh, K. 2017. An Improved Approximation for *k*-Median and Positive Correlation in Budgeted Optimization. *ACM Trans. Algorithms* 13(2): 23:1–23:31. doi:10.1145/2981561. URL https://doi.org/10.1145/2981561.

Chamikara, M.; Bertok, P.; Khalil, I.; Liu, D.; and Camtepe, S. 2020. Privacy Preserving Face Recognition Utilizing Differential Privacy. *Computers & Security* 97: 101951. ISSN 0167-4048. doi:https://doi.org/10.1016/j.cose.2020.101951. URL http://www.sciencedirect.com/science/article/pii/S0167404820302273.

Charikar, M.; Guha, S.; Tardos, É.; and Shmoys, D. B. 2002. A constant-factor approximation algorithm for the k-median problem. *Journal of Computer and System Sciences* 65(1): 129–149.

Chrobak, M.; Kenyon, C.; and Young, N. 2006. The reverse greedy algorithm for the metric k-median problem. *Information Processing Letters* 97(2): 68–72.

Dwork, C.; Kenthapadi, K.; McSherry, F.; Mironov, I.; and Naor, M. 2006a. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 486–503. Springer.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006b. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 265–284. ISBN 3540327312. doi:10.1007/11681878_14.

Dwork, C.; Roth, A.; et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9(3–4): 211–407.

Feldman, D.; Fiat, A.; Kaplan, H.; and Nissim, K. 2009. Private coresets. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 361–370.

Feldman, D.; Xiang, C.; Zhu, R.; and Rus, D. 2017. Coresets for differentially private k-means clustering and applications to privacy in mobile sensor networks. In *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 3–16. IEEE.

Gupta, A.; Ligett, K.; McSherry, F.; Roth, A.; and Talwar, K. 2010. Differentially private combinatorial optimization. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, 1106–1125. SIAM.

Jain, K.; Mahdian, M.; Markakis, E.; Saberi, A.; and Vazirani, V. V. 2003. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *Journal of the ACM (JACM)* 50(6): 795–824.

Jain, K.; and Vazirani, V. V. 2001. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM (JACM)* 48(2): 274–296.

Kaplan, H.; and Stemmer, U. 2018. Differentially private k-means with constant multiplicative error. In *Advances in Neural Information Processing Systems*, 5431–5441.

Kariv, O.; and Hakimi, S. L. 1979. An algorithmic approach to network location problems. I: The p-centers. *SIAM Journal on Applied Mathematics* 37(3): 513–538.

McSherry, F.; and Mironov, I. 2009. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 627–636.

McSherry, F.; and Talwar, K. 2007. Mechanism Design via Differential Privacy. In *Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE. URL https://www.microsoft.com/en-us/research/publication/mechanism-design-via-differential-privacy/.

Mueen, J. Z. A.; and Keogh, E. 2012. Clustering time series using unsupervised-shapelets. In *International Conference on Data Mining (ICDM)*.

Ni, L.; Li, C.; Wang, X.; Jiang, H.; and Yu, J. 2018. DP-MCDBSCAN: Differential Privacy Preserving Multi-Core DBSCAN Clustering for Network User Data. *IEEE Access* 6: 21053–21063. doi:10.1109/ACCESS.2018.2824798.

Nissim, K.; Raskhodnikova, S.; and Smith, A. 2007. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, 75–84.

Nissim, K.; and Stemmer, U. 2018. Clustering Algorithms for the Centralized and Local Models. In *Algorithmic Learning Theory*, 619–653.

Patel, V. M.; Van Nguyen, H.; and Vidal, R. 2013. Latent space sparse subspace clustering. In *Proceedings of the IEEE international conference on computer vision*, 225–232.

Schafer, J. B.; Frankowski, D.; Herlocker, J.; and Sen, S. 2007. Collaborative filtering recommender systems. In *The adaptive web*, 291–324. Springer.

Xiong, J.; Ren, J.; Chen, L.; Yao, Z.; Lin, M.; Wu, D.; and Niu, B. 2018. Enhancing privacy and availability for data clustering in intelligent electrical service of IoT. *IEEE Internet of Things Journal* 6(2): 1530–1540.

Yang, B.; Fu, X.; Sidiropoulos, N. D.; and Hong, M. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 3861–3870. JMLR. org.

Álvarez Marañón, G. 2020. What Differential Privacy Is and Why Google and Apple Are Using It with Your Data. *Think Big* URL business.blogthinkbig.com/differential-privacy-google-apple-using-it-with-your-data/.