Supplementary Material of *Differentially Private Clustering in High-Dimensional Euclidean Spaces*

Maria-Florina Balcan [1] **Travis Dick** [1] **Yingyu Liang** [2] **Wenlong Mou** [3] **Hongyang Zhang** [1]

1. Additional Related Work

Non-Private Clustering: There is a wide range of prior work on the problem of center-based clustering in the absence of privacy requirement. It is known that exact optimization of objective function in $\mathbb{R}^d$ is not computationally possible (Dasgupta, 2008) even for the problem of 2-means clustering. To avoid the computational obstacle, several approximation algorithms have been developed, e.g., by the local swap (Kanungo et al., 2002; Arya et al., 2004), careful seeding (Arthur & Vassilvitskii, 2007), or enumeration via sample-based loss estimator (Kumar et al., 2010). Another line of research focuses on the recovery of optimal data partition under stability or separation assumption (Balcan et al., 2009; Awasthi & Balcan, 2014).

It is worth noting that most of existing work for clustering in $\mathbb{R}^d$ with reasonable approximation guarantee relies on the construction of a candidate set of centers. In particular, Matoušek (2000) constructed a $(1+\epsilon)$-approximate candidate set via griding argument, which was widely applied in the later work (Kanungo et al., 2002; Makarychev et al., 2015). Kumar et al. (2010) took the average of a randomly sampled subset of data points as the set of candidate centers. However, none of these approaches can be easily adapted to the differentially private settings.

2. Why Direct Extension Failed

It would be natural to ask, if one has read (Matoušek, 2000), why their discretization methods cannot directly extend to private setting, with randomized decision in the sub-division procedure. In the following example, we will see that, the privatized vanilla recursive partition, i.e., direct use of discretization routine in Algorithm 1, will lead to arbitrarily bad performance in our settings.

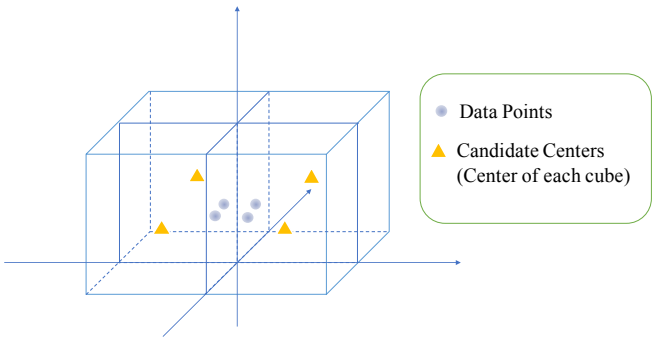

Figure 1. Illustration for the Worst Case for Partition Procedure

Consider a set of n points S in $\mathbb{R}^p$, with $p \geq 1+\log n$. The initial cube we are working on is $[-1,1]^p$, which contains set S.

[1]Carnegie Mellon University, Pittsburgh, PA, USA [2]Princeton University, Princeton, NJ, USA [3]Peking University, Beijing, China. Correspondence to: Wenlong Mou <mouwenlong@pku.edu.cn>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

Let $S = \{x_i\}_{i=1}^n$ and $x_i = [1/2, 0, 0, \cdots, 0]^T + \eta\sigma_i$, where $\eta = \frac{1}{p^2n^2}$ and $\sigma_i \in [0, \pm 1, \pm 1, \cdots, \pm 1]^T$, $\sigma_i \neq \sigma_j, \forall i \neq j$. Apparently S is a small cluster with clustering loss less than 1. Using the hierarchical partition procedure, we will get 2^p cubes corresponding to 2^phyperoctants. According to their definition, each points in S is divided into a separate cube. Due to calculation similar to Lemma 1, none of these cubes will be further divided, with high probability, and the partition procedure ends up returning a set of candidate centers with distance at least $\frac{1}{2} - \eta$ to each of x_i, and the resulted clustering loss becomes at least $\Omega(n)$. In Figure 1, we illustrate the bad case for discretization procedure.

What's wrong with this method? In (Matoušek, 2000), the discretization procedure can carry on until meeting a threshold that guarantees candidate cost at the same scale of optimal clustering loss. However, for privacy reasons, the partition procedure *has to stop*, in order to hide the accurate location of a single data point. Thus we do not want a cluster to be divided into too many parts during the procedure, as in the example above.

Fortunately, this worst case is extremely atypical in high dimensions, and can be avoided via repeated random shift. Using probabilistic arguments in high dimensions, we can make sure that each optimal cluster is *fully contained* in the cube with a proper scale.

3. Omitted Proofs in Section 4

Theorem 1 (Theorem 1 in the Main Body)**.** *The set C generated by Algorithm 1 satisfies $|C| \leq n \log n$, with probability $1 - \delta$.*

Proof. First of all, it is easy to verify that the function $f(\cdot)$ defined in Algorithm 1 satisfies

$$\Pr\left(Q_i \text{ partitioned} | Q_i \text{ empty}\right) = f(0) \leq \frac{\delta}{n^{10}}.$$

Consider a (non-private) tree $\mathcal{T}$ partitioning generated as follows: the root node corresponds to the initial cube Q. For each round, we subdivide the cube in each dimensions evenly, resulting in 2^p cubes. A smaller cube will be active when it contains at least one point, until the depth of this node in the tree grows to be $\log n$. The tree has at most $\log n$ levels, while each level has at most n nodes. So the size of this tree is upper bounded with $n \log n$. Let $\tilde{\mathcal{T}}$ denote the partitioning tree generated by Algorithm 1, we have

$$\begin{aligned}
\Pr\left(\tilde{\mathcal{T}} \not\subseteq \mathcal{T}\right) \leq & \Pr\left(\exists n \in \mathsf{leaf}(\mathcal{T}), n \text{ is partitioned in } \tilde{\mathcal{T}}\right) \\
\leq & \sum_{n \in \mathsf{leaf}(\mathcal{T})} \Pr\left(n \text{ is partitioned in } \tilde{\mathcal{T}}\right) \\
\leq & |\mathcal{T}| f(0) \leq \delta.
\end{aligned}$$

With probability $1 - \delta$, the number of cells generated by Algorithm 1 is no more than $|\mathcal{T}|$, and the size of C is thus bounded. □

Theorem 2 (Theorem 2 in the Main Body)**.** *Algorithm 1 preserves ϵ-differential privacy.*

Proof. Consider a layers of discretization with $a \leq \log n$. By adaptive composition lemma, it suffices to show that given the active cubes and candidate centers constructed in upper layers, the points added to C in this layer preserves $\frac{\epsilon}{\log n}$-differential privacy.

Given current C and $\mathcal{A}$ fixed, modification on a single data point will influence the point counts of at most two active cubes. By definition we only need to show

$$\forall i, \quad \Pr_S\left(\{Q_i^{(l)}\}_{l=1}^{2^d} \subseteq \mathcal{A}\right) \leq \exp\left(\frac{\epsilon}{2\log n}\right) \Pr_{S'}\left(\{Q_i^{(l)}\}_{l=1}^{2^d} \subseteq \mathcal{A}\right).$$

It is easy to verify our construction of $f(\cdot)$ satisfies this bound. □

Theorem 3 (Theorem 3 in the Main Body)**.** *The following event holds with probability at least $1 - \delta$: In Algorithm 1, when a cube Q_i is removed from $\mathcal{A}$ and its subdivided cubes are not inserted to $\mathcal{A}$, then we have either $|Q_i \cap X| \leq \mathsf{O}\left(\gamma \log \frac{n}{\delta}\right)$, or the edge length of Q_i is at most $\frac{\Lambda}{n}$.*

Proof. It is easy to verify that $f(\cdot)$ satisfies

$$\Pr\left(Q_i \text{ partitioned}\middle| |Q_i \cap X| \geq 10\gamma \log\frac{n}{\delta}\right) \geq f\left(10\gamma\log\frac{n}{\delta}\right) \geq 1 - \frac{\delta}{n^{10}}.$$

Consider a (non-private) partition tree $\mathcal{T}'$. The construction of this tree is based on bisection for each dimension, similar to the proof of Theorem 1. However, this time a cell will stop being partitioned when the number of data points inside is less than $10\gamma \log \frac{n}{\delta}$. Apparently we have $|\mathcal{T}'| \leq |\mathcal{T}| \leq 2^p n \log n$, where $\mathcal{T}$ is the tree constructed in the proof of Theorem 1.

Since leaf nodes of $\mathcal{T}'$ satisfy the desired properties of this theorem, it suffices to show that $\mathcal{T}' \subseteq \tilde{\mathcal{T}}$ with probability $1-\delta$. Actually, we have

$$\begin{aligned}\Pr\left(\mathcal{T}' \nsubseteq \tilde{\mathcal{T}}\right) \leq& \Pr\left(\exists n \in \mathsf{internal}(\mathcal{T}'), n \text{ is not partitioned in } \tilde{\mathcal{T}}\right)\\ \leq& \sum_{n\in\mathsf{internal}(\mathcal{T}')} \Pr\left(n \text{ is not partitioned in } \tilde{\mathcal{T}}\right)\\ \leq& |\mathcal{T}'|\left(1 - f\left(10\gamma\log\frac{n}{\delta}\right)\right) \leq \delta.\end{aligned}$$

□

Theorem 4 (Theorem 4 in the Main Body)**.** *Algorithm 2 preserves ϵ-differential privacy.*

Proof. The proof of privacy is simply done by T-fold composition theorem over the T independent trials of the private partition procedure, each of which preserves $\frac{\epsilon}{T}$-differential privacy. □

Theorem 5 (Theorem 5 in the Main Body)**.** *Algorithm 2 outputs an $\left(\mathsf{O}(\log^3 n), \mathsf{O}(k\gamma(\frac{\epsilon}{T})\log\frac{n}{\delta})\right)$ candidate set with probability at least $1-\delta$, where $\gamma(c) = \frac{40}{c}\log\frac{n}{\delta}\log n$, and $T = k\log\frac{n}{\delta}$.*

Proof. Suppose the optimal set of centers are $u_1^*, u_2^*, \cdots, u_k^*$, fixed but unknown. Let S_j^* be the cluster induced by u_j^*, i.e. $S_j = \{i : j = \mathrm{argmin}_l \|x_i - u_l\|\}$. Let $r_j^* = \sqrt{\frac{1}{|S_j^*|}\sum_{i\in S_j^*}\|x_i - u_l\|^2}$.

For $\forall j = 1, 2, \cdots, k$, we say u_j^* is captured by C with factor L iff $\mathcal{B}\left(u_j^*, Lr_j^* + \mathsf{O}(\frac{1}{n})\right) \cap C \neq \varnothing$. If each u_j^* with at least $\gamma(\frac{\epsilon}{T})\log\frac{n}{\delta}$ is captured by C with factor L, let $\tilde{u}_j \in \mathcal{B}\left(u_j^*, Lr_j^* + \mathsf{O}(\frac{1}{n})\right) \cap C$, we have

$$\begin{aligned}\sum_{i=1}^n \min_j \|x_i - \tilde{u}_j\|^2 \leq& \sum_{j=1}^k \sum_{i\in S_j^*} \|x_j - \tilde{u}_j\|^2 \leq \sum_{j=1}^k \left(\sum_{i\in S_j^*}\|x_j - u_j^*\|^2 + |S_j|\cdot\|u_j^* - \tilde{u}_j\|^2\right)\\ =& \mathsf{OPT} + \sum_{u_j^*\in\mathsf{captured}} |S_j|\cdot\|u_j^* - \tilde{u}_j\|^2 + \sum_{u_j^*\notin\mathsf{captured}} |S_j|\cdot\|u_j^* - \tilde{u}_j\|^2\\ \leq& \mathsf{OPT} + L^2\cdot\mathsf{OPT} + \mathsf{O}(\Lambda^2) + \mathsf{O}\left(k\gamma(\frac{\epsilon}{T})\log\frac{n}{\delta}\Lambda^2\right).\end{aligned}$$

Thus, to prove the quality of our candidate set, it suffices to show that each u_j^* with $|S_j^*| \geq k\gamma(\frac{\epsilon}{T})\log\frac{n}{\delta}$ is captured by C with factor $\mathsf{O}(\log^{\frac{3}{2}} n)$, with high probability.
For $\forall j \in \{1, 2, \cdots, k\}$ fixed, since $\frac{1}{|S_j^*|}\sum_{i\in S_j^*}\|x_i - u_l\|^2 = r_j^{*2}$, using Chebyshev Inequality, we have

$$\left|\mathcal{B}(u_j^*, 2r_j^*) \cap S_j^*\right| \geq \frac{1}{2}\left|S_j^*\right|.$$

Given the randomized shift vector v in Algorithm 2 fixed, consider an (infinite) rectangular grid $\mathcal{G}_v$, constructed by recursive bisection on each dimension of Q_v. There must exist a cube $\bar{Q}(j) \in \mathcal{G}_v$, such that $u_j^* \in \bar{Q}(j)$ and the edge length of $\bar{Q}(j)$ is between $2pr_j^*$ and $4pr_j^*$. If any point in $\bar{Q}(j)$ is added during the execution of Algorithm 1, its distance to u_j^* is at most $4r_j^* p\sqrt{p} = r_j^*\cdot\mathsf{O}(\log^{\frac{3}{2}} n)$. Therefore, if $\bar{Q}_j$ is ever active during the discretization, this cluster center must have been captured by C.

According to Theorem 3, conditioned on the event that no failure occurs (which has negligible probability), there can only be two cases for which $\bar{Q}(j)$ does not appear in C:

- The edge length of $\bar{Q}(j)$ is less than $\frac{1}{n}$;
- $\left|\bar{Q}(j) \cap S_j^*\right| \leq \mathsf{O}\left(\gamma(\frac{\epsilon}{T}) \log \frac{n}{\delta}\right)$.

If it is the first case, we can turn to a cube $\bar{Q}'(j) \in \mathcal{G}_v$ that contains $\bar{Q}(j)$ with edge length exactly $\Theta(\frac{1}{n})$. Since our definition of capturing allows $\mathsf{O}(\frac{1}{n})$ additive error on the radius, u_j^* will also be captured if $\bar{Q}'(j)$ becomes active. Since $\bar{Q}'(j)$ has larger size and potentially more points in S_j^*, we only need to show that a cube containing u_j^* with edge length at least $2pr_j^*$ is likely to a large number of points in S_j^* and thus becomes active in Algorithm 1.
Let's now turn to the randomness of v, in terms of captures for u_j^*, shifting Q_v and $\mathcal{G}_v$ uniformly at random is equivalent to shifting $\bar{Q}(j)$ uniformly at random, given the fact that it contains u_j^*.

(The location of the cell of this scale that contains u_j^* may vary in the grid. However, u_j^* is indifferent with "which" cell contains it, its capture only depends on its relative location within this axis-aligned cell.)

Therefore, if $|S_j| \geq \Omega(\gamma(\frac{\epsilon}{T}) \log \frac{n}{\delta})$, we will have

$$\begin{aligned}
\Pr\left(u_j^* \in \mathsf{captured}(v)\right) &\geq \Pr\left(\left|S_j^* \cap \bar{Q}(j)\right| \geq \Omega(\gamma(\frac{\epsilon}{T}) \log \frac{n}{\delta})\right) \\
&\geq \Pr\left(\left|S_j^* \cap \bar{Q}(j)\right| \geq \frac{1}{2}|S_j^*|)\right) \\
&\geq \Pr\left(\mathcal{B}(u_j^*, 2r_j^*) \subseteq \bar{Q}_j\right) \\
&= \left(1 - \frac{2}{p}\right)^p \geq \frac{1}{27}.
\end{aligned}$$

For each $j \in \{1, 2, \cdots, k\}$, we assign $27 \log \frac{n}{\delta}$ independent trials to it, and it is easy to see that it is captured by C with probability at least $1 - \frac{\delta}{n}$. By aggregating the $27k \log \frac{n}{\delta}$ trails together and applying union bound, we conclude that with probability at least $1 - \delta$, we have

$$\forall j \in \{1, 2, \cdots, k\}, \quad |S_j^*| \geq \Omega\left(\gamma(\frac{\epsilon}{T}) \log \frac{n}{\delta}\right) \Rightarrow \exists \tilde{u}_j \in C \text{ captures } u_j^*.$$

So the proof is completed. □

4. Omitted Proofs in Section 5

Theorem 6 (Theorem 6 in the Main Body)**.** *Algorithm 3 preserves ϵ-differential privacy.*

Proof. The privacy guarantee is straightforward using composition theorem over T rounds of the algorithm, and an additional exponential mechanism that selects the best one. It is easy to verify the sensitivity of loss increments $\mathcal{L}(Z - \{x\} + \{y\}) - \mathcal{L}(Z)$ is $8\Lambda^2$, the privacy guarantee of exponential mechanism in each round follows. □

Theorem 7 (Theorem 7 in the Main Body)**.** *With probability $1 - \delta$, the output of Algorithm 3 satisfies*

$$\mathcal{L}(Z) \leq 30\mathsf{OPT} + \mathsf{O}\left(\frac{k^2\Lambda^2}{\epsilon} \log^2 \frac{n|\mathcal{C}|}{\delta}\right).$$

Proof. The proof inspires from (Gupta et al., 2010). Basically, we use the following fact, which is derived from the construction of swap pairs and Lemma 2.2 in (Kanungo et al., 2002):

For clustering centers Z, there exists a set of k swaps $\{(x_i, y_i)\}_{i=1}^k$ such that

$$\sum_{i=1}^{k} \mathcal{L}(Z) - \mathcal{L}(Z - \{x_i\} + \{y_i\}) \geq 3\mathcal{L}(Z) - \mathsf{OPT} - 2\sum_{i=1}^{n} \|x_i - z(o_i)\|^2,$$

where o_i is the optimal cluster to which x_i is assigned, and $z(o)$ is point o's nearest center in Z.

Following the proof for local swap heuristics, we can bound the extra term.

$$\begin{aligned}R =& \sum_{i=1}^{n}\sum_{i=1}^{n}\|x_i - z(o_i)\|^2 \leq 2\mathsf{OPT} + \mathcal{L}(Z) + 2\sum_{i=1}^{n}\|x_i - o_i\| \cdot \|x_i - z_i\| \\ \leq& 2\mathsf{OPT} + \mathcal{L}(Z) + 2\sqrt{OPT \cdot \mathcal{L}(Z)}.\end{aligned}$$

Putting them together, we have that $\exists x \in Z, y \in \mathcal{C}$ such that

$$\begin{aligned}\mathcal{L}(Z) - \mathcal{L}(Z - \{x\} + \{y\}) \geq& \frac{1}{k}\left(3\mathcal{L}(Z) - \mathsf{OPT} - 2R\right) \\ \geq& \frac{1}{k}\left(\mathcal{L}(Z) - 5\mathsf{OPT} - 4\sqrt{\mathsf{OPT} \cdot \mathcal{L}(Z)}\right) \\ \geq& \frac{1}{2k}\left(\mathcal{L}(Z) - 30\mathsf{OPT}\right).\end{aligned}$$

The rest of this proof proceeds just as in (Gupta et al., 2010): exponential mechanism guarantees the bound 4 holds with an additive term $\mathsf{O}\left(\frac{1}{\epsilon}\log\frac{n|\mathcal{C}|}{\delta}\right)$ with failure probability $\frac{\delta}{n}$. T rounds of iteration guarantees the multiplicative term being reduced to constant order, except for the case of $\mathsf{OPT} = \mathsf{O}(\frac{k^2}{\epsilon}\log^2\frac{n|C|}{\delta})$ where all the excess loss goes to the additive term. Combining these facts together using union bound over failure probability we conclude the result. □

Theorem 8 (Theorem 8 in the Main Body)**.** *Under the following assumptions:*

- *Algorithm* `candidate` $(\{x_i\}_{i=1}^n, \epsilon, \delta)$ *preserves* ϵ*-differential privacy for* $\{x_i\}_{i=1}^n$.
- *Given any* C*, Algorithm* `localswap` $(\{x_i\}_{i=1}^n, C, \epsilon, \delta)$ *preserves* ϵ*-differential privacy for* $\{x_i\}_{i=1}^n$.

Algorithm 4 preserves ϵ*-differential privacy.*

Proof. In each of T-rounds, the two sub-routines each preserves $\frac{\epsilon}{6T}$-DP. Given the centers $u_1, u_2, \cdots, u_k$ in projected space fixed, changing the position for one of data points (and also resulting the change in projected space) will affect at most two clustering centers. The noised version of cluster count s_j preserves $\frac{\epsilon}{12T}$-DP, and the recovery procedure also preserves $\frac{\epsilon}{12T}$-DP given the cluster counts known and fixed. At last, the exponential mechanism preserves $\frac{\epsilon}{6}$-DP, putting them together using composition theorem, we have the privacy guarantee. □

Theorem 9 (Theorem 9 in the Main Body)**.** *Instantiated by algorithms that guarantee:*

- *With probability* $\frac{2}{3}$*, algorithm* `candidate` $(\{x_i\}_{i=1}^n, \epsilon, \delta)$ *can output an* $(\alpha, \sigma_1(\epsilon))$*-approximate candidate set.*
- *With probability* $\frac{2}{3}$*, algorithm* `localswap` $(\{x_i\}_{i=1}^n, C, \epsilon, \delta)$ *achieves clustering loss with multiplicative approximation factor* c *and additive term* $\sigma_2(\epsilon)$*, compared with optimal clustering centers in the discrete space.*

Algorithm 4 achieves the following bound with probability $1 - \delta$*:*

$$\mathcal{L}\left(\{z_j\}_{j=1}^k\right) \leq 3c\alpha\mathsf{OPT} + 3C\sigma_1' + 3\sigma_2' + \mathsf{O}\left(\frac{d\Lambda^2\log^3\frac{1}{\delta}}{\epsilon^2}\right),$$

where $\sigma_i' = \sigma_i\left(\frac{\epsilon}{2\log 1/\delta}\right)$ *for* $i = 1, 2$

Proof. With probability $\frac{1}{3}$, the centers $u_1, u_2, \cdots, u_k$ in projected space satisfies

$$\sum_{i=1}^{n}\min_j\|y_i - u_j\|^2 \leq c\mathsf{OPT}_{\mathsf{discrete}} + \sigma_2 \leq c\alpha(\mathsf{OPT}_{\mathsf{projected}} + \sigma_1) + \sigma_2.$$

The JL transform guarantees that with probability $\frac{1}{2}$, the pairwise distances between points uniformly satisfies

$$\forall i, j \in \{1, 2, \cdots n\}, \quad \|y_i - y_j\|^2 \leq \left(1 \pm \frac{1}{2}\right) \|x_i - x_j\|^2.$$

This event is also independent with randomized algorithms in projected spaces.

Since the optimal clustering loss for k-means depends only on the pairwise distances between data points, we have the following fact under the event that JL transform successfully preserves pairwise distances: (C_j denotes optimal clustering assignment)

$$\begin{aligned}\mathsf{OPT}_{\mathsf{projected}} \leq & \sum_{i \in C_j^*} \|u_i - \nu_j^*\|^2 = \sum_{j=1}^{k} \frac{1}{2|C_j^*|} \sum_{i \in C_j^*} \sum_{l \in C_j^*} \|u_i - u_l\|^2 \\ \leq & \frac{3}{2} \sum_{j=1}^{k} \frac{1}{2|C_j^*|} \sum_{i \in C_j^*} \sum_{l \in C_j^*} \|x_i - x_l\|^2 = \frac{3}{2} \mathsf{OPT}.\end{aligned}$$

On the other hand, we have

$$\begin{aligned}\sum_{i \in S_j} \|x_i - \mu_j\|^2 = & \sum_{j=1}^{k} \frac{1}{2|S_j|} \sum_{i \in S_j} \sum_{l \in S_j} \|x_i - x_l\|^2 \\ \leq & 2 \sum_{j=1}^{k} \frac{1}{2|S_j|} \sum_{i \in S_j} \sum_{l \in S_j} \|u_i - u_l\|^2 = 2 \sum_{i \in S_j} \|u_i - \nu_j\|^2,\end{aligned}$$

where μ_j (μ_j^*, resp.) are clustering center for S_j (C_j^*, resp.) in $\mathbb{R}^d$, and ν_j (ν_j^*, resp.) are clustering center for S_j (C_j^*, resp.) in $\mathbb{R}^p$.

The noise added to cluster sizes, as well as the Laplacian mechanism, leads to the additional term. Thus we have the desired bound to hold with constant probability. By repeating it with T independent trials and selecting the best, the failure probability is reduced to δ. The additive loss induced by exponential mechanism in the last step is dominated by previous terms. □

5. Omitted Proofs in Section 6

Theorem 10 (Theorem 10 in the Main Body)**.** *Algorithm 5 preserves ϵ-differential privacy.*

Proof. The privacy proof is straightforward: for $\frac{2s}{\eta}$ rounds of the algorithm, each round is exponential mechanism accompanied by Laplacian mechanism. Using the fact that both parts preserves $\frac{\epsilon}{2T}$-DP, the privacy proof then follows via composition. □

Theorem 11 (Theorem 11 in the Main Body)**.** *The output of Algorithm 5 satisfies the following with probability at least $1 - \delta$:*

$$\sum_{i=1}^{n} \|x_i - v\|^2 \leq \frac{1}{1-\eta} \mathsf{OPT} + \mathsf{O}\left(\frac{\Lambda^2 s^2 \ln \frac{ds}{\eta\delta}}{\eta^2 \epsilon}\right).$$

Proof. Let $\pi(j)$ denote the index of entries in μ with the j-th largest absolute value. Note that removing j entries from $\{1, 2, \cdots d\}$ makes the largest absolute value at least μ_{j+1}, exponential mechanism guarantees that, for the index r_j sampled in j-th round, we have the following with probability at least $\frac{\eta\delta}{2s}$:

$$|\mu_{r_j}| \geq |\mu_{\pi(j)}| - \mathsf{O}\left(\frac{\Lambda s \ln \frac{ds}{\eta\delta}}{\eta \epsilon n}\right), \quad j = 1, 2, \cdots, \frac{s}{\eta}.$$

Let $\tilde{\mu} = \sum_{j=1}^{s/\eta} \mu_{\pi(j)} e_{\pi(j)}$, For $\forall j \in \{1, 2, \cdots, d\}$, let $S_j = \{i \in [n] : x_{ij} \neq 0\}$, $c_j = |S_j|$ and $v_j = \frac{1}{c_j} \sum_{i \in S_j} x_{ij}$.

$$n\|\tilde{\mu} - \mu\|^2 = n \sum_{j=s/\eta+1}^{d} \mu_{\pi(j)}^2 = \sum_{j=s/\eta+1}^{d} \frac{c_{\pi(j)}^2 v_{\pi(j)}^2}{n}.$$

On the other hand, we have

$$\sum_{i=1}^{n} \|x_i - \mu\|^2 = \sum_{j=1}^{d} \left(\frac{c_j(n-c_j)}{n} v_j^2 + \sum_{i \in S_j} |x_{ij} - c_j|^2 \right) \geq \sum_{j=1}^{d} \frac{c_j(n-c_j)}{n} v_j^2.$$

Sort the d entries again according to the value of c_j and let $\tau(j)$ denote the index of entries in μ with the j-th largest c_j, we have

$$n\|\tilde{\mu} - \mu\|^2 = \sum_{j=s/\eta+1}^{d} \frac{c_{\pi(j)}^2 v_{\pi(j)}^2}{n} \leq \sum_{j=s/\eta+1}^{d} \frac{c_{\tau(j)}^2 v_{\tau(j)}^2}{n}.$$

Since $\sum_{j=1}^{d} c_j \leq ns$, Markov inequality implies $|\{j \in [d] : c_j \geq \eta n\}| \leq \frac{s}{\eta}$. Thus we have

$$\begin{aligned}
&\sum_{j=s/\eta+1}^{d} \frac{c_{\tau(j)}^2 v_{\tau(j)}^2}{n} \leq \frac{\eta}{1-\eta} \sum_{j=s/\eta+1}^{d} \frac{c_{\tau(j)}(n - c_{\tau(j)}) v_{\tau(j)}^2}{n} \\
\leq &\frac{\eta}{1-\eta} \sum_{j=1}^{d} \frac{c_{\tau(j)}(n - c_{\tau(j)}) v_{\tau(j)}^2}{n} \leq \frac{\eta}{1-\eta} \mathsf{OPT}.
\end{aligned}$$

Putting them together, we have

$$\begin{aligned}
\sum_{i=1}^{n} \|x_i - v\|^2 = &\sum_{i=1}^{n} \|x_i - \mu\|^2 + n\|\mu - v\|^2 \\
\leq &\mathsf{OPT} + n\|\mu - \tilde{\mu}\|^2 + \mathsf{O}\left(\frac{\Lambda^2 s^2 \ln \frac{ds}{\eta\delta}}{\eta^2 \epsilon} \right) \\
\leq &\frac{1}{1-\eta} \mathsf{OPT} + \mathsf{O}\left(\frac{\Lambda^2 s^2 \ln \frac{ds}{\eta\delta}}{\eta^2 \epsilon} \right).
\end{aligned}$$

□

Theorem 12 (Theorem 12 in the Main Body)**.** *For k-median clustering problem, there is an ϵ-differential private algorithms that runs in $poly(k, d, n)$ time, which releases a set of centers $\tilde{z}_1, \tilde{z}_2, \cdots, \tilde{z}_k$, that satisfies the following with probability $1 - \delta$:*

$$\mathcal{L}(\tilde{z}_1, \tilde{z}_2, \cdots, \tilde{z}_k) \leq \mathsf{O}(\log^{3/2} n) \cdot \mathsf{OPT} + \mathsf{O}\left(\frac{(k^2 + d)\Lambda}{\epsilon} \log^3 \frac{n}{\delta} \right).$$

Proof. By plugging in the error bounds and simple calculation, the centers $u_1, u_2, \cdots, u_k$ in projected space satisfy the following with probability $1 - \delta$:

$$\sum_{i=1}^{n} \min_j \|y_i - u_j\|^2 \leq \mathsf{O}(\log^{3/2} n) \cdot \mathsf{OPT}_{\mathsf{projected}} + \mathsf{O}\left(\frac{(k^2 + d)\Lambda}{\epsilon} \log^3 \frac{n}{\delta} \right).$$

The JL transform preserves distances with up to $1 \pm \frac{1}{2}$ multiplicative error with probability $\frac{2}{3}$. To make use of this fact, we replace sum-of-square decomposition with triangle inequality, using data points as intermediate step.

$$\begin{aligned}\mathsf{OPT}_{\mathsf{projected}} \leq& \sum_{i\in C_j^*} \|u_i - \nu_j^*\| \leq \sum_{j=1}^{k} \min_l \sum_{i\in C_j} \|u_i - u_l\| \\ \leq& \frac{3}{2}\sum_{j=1}^{k} \min_l \sum_{i\in C_j} \|x_i - x_l\| = \frac{3}{2}\mathsf{OPT}.\end{aligned}$$

On the other hand, we have

$$\begin{aligned}\sum_{i\in S_j} \|x_i - \mu_j\| \leq& \sum_{j=1}^{k} \min_l \sum_{l\in S_j} \|x_i - x_l\| \\ \leq& 2\sum_{j=1}^{k} \min_l \sum_{l\in S_j} \|u_i - u_l\| \leq 4 \sum_{i\in S_j} \|u_i - \nu_j\|,\end{aligned}$$

where μ_j (μ_j^*, resp.) are clustering center for S_j (C_j^*, resp.) in $\mathbb{R}^d$, and ν_j (ν_j^*, resp.) are clustering center for S_j (C_j^*, resp.) in $\mathbb{R}^p$.

The excess losses incurred by log-concave sampling and discrete exponential mechanisms in the algorithm are dominated by the previous term. Putting them altogether, we have the bound. □

6. Additional Experimental Details

In this section we provide additional details about experiments in our paper. We will first give a detailed description on the real-world and synthetic datasets we are using, and then provide additional details on the comparison to existing works including (Nock et al., 2016) and (Su et al., 2016). We will also present results about effect of number of clusters k and dimension d on the clustering loss.

6.1. Description of Datasets

We compare our algorithm against the non-private k-means++ algorithm, SuLQ, k-variates++ (Nock et al., 2016), low-dimensional algorithm (Su et al., 2016) and Sample and Aggregate on the following datasets.

MNIST: We used the raw pixels of the MNIST (LeCun et al., 1998) dataset. It has 70000 examples and 784 features.

CIFAR-10: We used the CIFAR10 dataset (Krizhevsky, 2009). Rather than the raw pixels, we obtained our feature representations from layer `in3c` (160 features) of a Google Inception (Szegedy et al., 2015) network trained for the classification task. We also created a second version of this dataset using the feature representations from layer `in4d` (144 features). Our dataset contains 100000 randomly sampled examples from this dataset.

Synthetic: For experiments in the main body, we used a synthetic datasets of 100000 samples drawn from an equal-weight mixture of 64 Gaussians in $\mathbb{R}^{100}$. Each Gaussian's covariance matrix is the identity and the mean is randomly sampled from $[0, 100]^d$. For further experiments in Appendix, we change the value of k and d to illustrate the effect of these parameters. In these experiments we always set k equal to the number of Gaussians in the mixture, with $k \in \{8, 16, 32, 64\}$, and we also select $d \in \{5, 50, 500, 5000\}$ to illustrate effect of dimension.

6.2. Additional Comparisons to Existing Works

We can also compare our approach with recent existing works such as (Nock et al., 2016) and (Nock et al., 2016) through experiments. Based on experimental comparisons, we find that our algorithm is the only one that works reasonably well simultaneously for large d and large k, while keeping good performance with small k and d.

It has been noticed that, the griding algorithm in (Su et al., 2016) only works for spaces with constant dimensions (see e.g. (Park et al., 2016)). So we are unable to evaluate Su *et al.*'s griding algorithm on the above datasets. Actually their algorithm has time and space complexity that is exponential in the data dimension (because it constructs a regular grid in d dimensions). In experiments, their grids caused memory allocation failure for $d > 6$, with 16GB RAM memory.

On the other hand, k-variates++ algorithm (Nock et al., 2016) is not designed for clustering problem with k more than constant. In their experimental part for privacy, (Nock et al., 2016) only did experiments with $k \leq 5$. Actually, its noise scale $\tilde{\epsilon}$ relies upon empirical estimates of data-dependent parameter δ_w and δ_s. We estimated these parameters on MNIST dataset with varying k and plug them into the formula. As k becomes larger than 5, the formula for setting parameter $\tilde{\epsilon}$ in Theorem 12 of (Nock et al., 2016) becomes negative, which makes the algorithm invalid.

Therefore, we carried on three sets of experiments to compare our methods with recent baselines.

- To compare with (Nock et al., 2016) in high dimensions, we did experiments on MNIST dataset with $k \in \{2, 3, 4, 5\}$.
- To compare with (Su et al., 2016) with many clusters, we did experiments on Gaussian mixture dataset with $k = 32$ and $d = 5$, which will be discussed in the results about varying d.
- To compare all approaches together, we also carried out a small scale experiment on a mixture of Gaussian dataset with 100000 samples in $d = 3$ dimensions with $k = 4$.

In Figure 2, we do the comparison in the first setting, where we set the privacy parameter $\epsilon = 0.5$. The plot is in logarithmic scale, and we can easily seen that 2 ends up adding too much noise for slightly larger k, while our algorithm performs reasonably well.

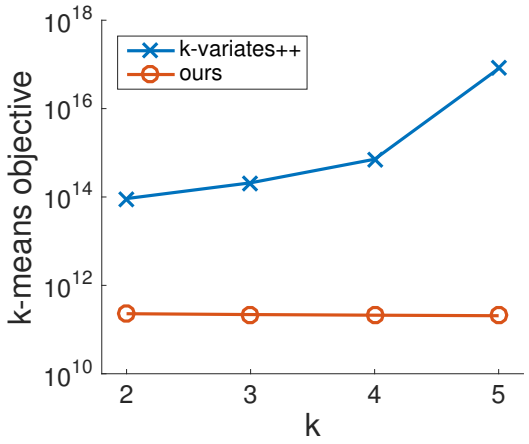

Figure 2. Comparison to (Nock et al., 2016) on MNIST dataset with $k = 2, 3, 4, 5$

We show the results of the third setting in Table 1, and compare all existing approaches together. Among all private methods, the griding algorithm in (Su et al., 2016) achieves the lowest objective value in this setting, while our algorithm and SuLQ both achieved reasonably good performance. As our focus is on modern big data setting with high-dimensional datasets, in this cases the griding algorithm cannot be applied.

Table 1. Objective values for all baseline algorithms on small-scale synthetic dataset.

Algorithm	Objective Value
k-means++	2.636e5
(Su et al., 2016)	2.638e5
SuLQ k-means	2.927e5
Ours	2.985e5
(Nock et al., 2016)	1.390e6
S&A	2.831e6

It is also worth noticing that running (Su et al., 2016) after randomized dimensionality reduction is not a polynomial-time algorithm. Actually the regular grids need $n^{\Omega(d)}$ cells to preserve good performance, and for the case of $d \sim \log n$, the running time and storage is still prohibitive. Even for constant-dimensional case like $d = 5$, the griding algorithm (Su et al., 2016) runs more than 5 hours while our algorithm runs within 15 minutes on the same machine. As the time and space costs for (Su et al., 2016) grows rapidly with $d \sim \log n > 5$, the costs will become prohibitive.

6.3. Additional CIFAR-10 Dataset

In this section we present the empirical comparison of algorithms on our second CIFAR-10 dataset, which is identical to the first except features are instead taken from a later layer of the inception network (layer `in4d`). Figure 3 shows the k-means objective of each algorithm when run for values of k from 2 to 64. Results are averaged over 5 runs.

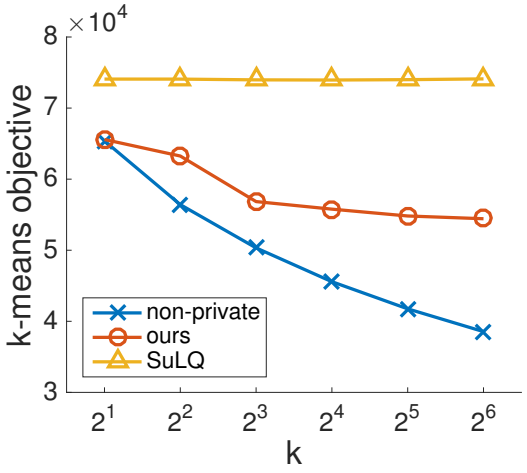

Figure 3. Objective values for various algorithms on the CIFAR-10 dataset with features extracted from layer `in4d` of a Google inception network.

6.4. Effect of dimension

In this section we directly evaluate the effect of the dimension on the objective value of the various algorithms. For this evaluation, we generate samples of size $n = 100000$ sampled from a mixture of 32 Gaussians with dimensions $d \in \{5, 50, 500, 5000\}$. All algorithms are run with the privacy parameter $\epsilon = 0.5$ and $k = 32$. Figure 4 shows the results of this experiment. This provides further justification for our claim that our algorithm scales to larger dimensions better than the existing algorithms. Again, the sample and aggregate algorithm is omitted from the plot because its objective value is several orders of magnitude worse and makes the plot difficult to read, even in log scale.

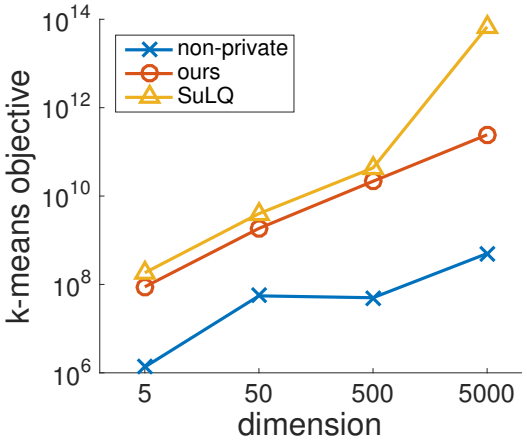

Figure 4. Objective values for various algorithms on synthetic datasets of increasing dimension.

To make a comprehensive comparison, we also run the griding algorithm in (Su et al., 2016) for the case of $d = 5$. (As we have discussed before, their algorithm is prohibitive for larger d). Their algorithm got clustering loss 1.84×10^7, which is much smaller compared to our clustering loss 6.43×10^7. However, the running time of their algorithm is much longer than us in this setting. And there's no obvious approach for their algorithm to scale up with k, as we have discussed.

6.5. Effect of number of intrinsic clusters

Finally, we directly evaluate the effect of the number of intrinsic clusters in the dataset. To do this, we generate datasets of $n = 100000$ sampled from mixtures of $G \in \{8, 16, 32, 64\}$ Gaussians in 100 dimensional space. All algorithms are run with k set to the true number of intrinsic clusters for each dataset. Figure 5 shows the results of this comparison.

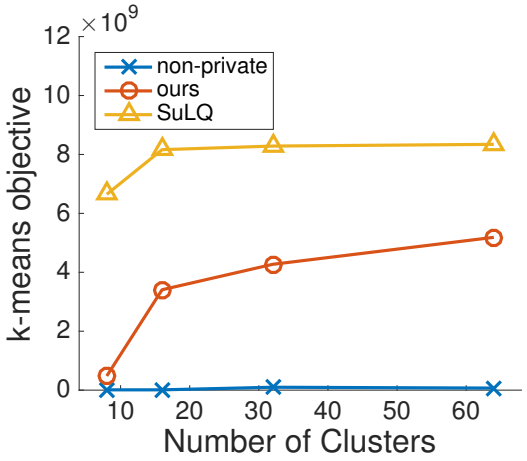

Figure 5. Objective values for various algorithms on synthetic datasets of growing intrinsic numbers of clusters.

References

Arthur, David and Vassilvitskii, Sergei. k-means++: The advantages of careful seeding. In *ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035, 2007.

Arya, Vijay, Garg, Naveen, Khandekar, Rohit, Meyerson, Adam, Munagala, Kamesh, and Pandit, Vinayaka. Local search heuristics for k-median and facility location problems. *SIAM Journal on computing*, 33(3):544–562, 2004.

Awasthi, Pranjal and Balcan, Maria-Florina. Center based clustering: A foundational perspective. 2014.

Balcan, Maria-Florina, Blum, Avrim, and Gupta, Anupam. Approximate clustering without the approximation. In *ACM-SIAM Symposium on Discrete Algorithms*, pp. 1068–1077, 2009.

Dasgupta, Sanjoy. *The hardness of k-means clustering*. Department of Computer Science and Engineering, University of California, San Diego, 2008.

Gupta, Anupam, Ligett, Katrina, McSherry, Frank, Roth, Aaron, and Talwar, Kunal. Differentially private combinatorial optimization. In *ACM-SIAM symposium on Discrete Algorithms*, pp. 1106–1125, 2010.

Kanungo, Tapas, Mount, David M, Netanyahu, Nathan S, Piatko, Christine D, Silverman, Ruth, and Wu, Angela Y. A local search approximation algorithm for k-means clustering. In *Annual Symposium on Computational Geometry*, pp. 10–18, 2002.

Krizhevsky, Alex. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Kumar, Amit, Sabharwal, Yogish, and Sen, Sandeep. Linear-time approximation schemes for clustering problems in any dimensions. *Journal of the ACM*, 57(2):5, 2010.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998.

Makarychev, Konstantin, Makarychev, Yury, Sviridenko, Maxim, and Ward, Justin. A bi-criteria approximation algorithm for k means. *arXiv preprint arXiv:1507.04227*, 2015.

Matoušek, Jirı. On approximate geometric k-clustering. *Discrete & Computational Geometry*, 24(1):61–84, 2000.

Nock, Richard, Canyasse, Raphaël, Boreli, Roksana, and Nielsen, Frank. k-variates++: more pluses in the k-means++. *arXiv preprint arXiv:1602.01198*, 2016.

Park, Mijung, Foulds, Jimmy, Chaudhuri, Kamalika, and Welling, Max. Practical privacy for expectation maximization. *arXiv preprint arXiv:1605.06995*, 2016.

Su, Dong, Cao, Jianneng, Li, Ninghui, Bertino, Elisa, and Jin, Hongxia. Differentially private k-means clustering. In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, pp. 26–37. ACM, 2016.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.