**Differentially Private** k**-Means via Exponential Mechanism and Max Cover**

**~**

**Huy L. Nguyên**\* **, Anamay Chaturvedi**\* **, Eric Z Xu**\*

Khoury College of Computer Sciences, Northeastern University

440 Huntington Ave

Boston, Massachusetts 02115

{hu.nguyen, chaturvedi.a, xu.er}@northeastern.edu

**Abstract**

We introduce a new (Ep, δp )-differentially private algorithm for the k-means clustering problem. Given a dataset in Eu- clidean space, the k-means clustering problem requires one to ﬁnd k points in that space such that the sum of squares of Euclidean distances between each data point and its clos- est respective point among the k returned is minimised. Al- though there exist privacy-preserving methods with good theoretical guarantees to solve this problem, in practice it is seen that it is the additive error which dictates the prac- tical performance of these methods. By reducing the prob- lem to a sequence of instances of maximum coverage on a grid, we are able to derive a new method that achieves lower additive error than previous works. For input datasets with cardinality n and diameter △, our algorithm has an O(△2 (k log2 n log(1/δp )/Ep + k√d log(1/δp )/Ep )) addi- tive error whilst maintaining constant multiplicative error. We conclude with some experiments and ﬁnd an improvement over previously implemented work for this problem.

**Introduction**

Clustering is a well-studied problem in theoretical computer science. A relatively general variant of this problem is when given a dataset D of size n to ﬁnd k centers that minimize the sum of distances of each point to its closest center. When the ambient space is Euclidean and the distance is the square of the Euclidean metric this is known as the k-means problem.

When algorithms handle sensitive information, an impor- tant requirement that they might be expected to fulﬁll is that of being *differentially private* (Dwork et al. 2016). Differen- tial privacy provides a framework for capturing the loss in privacy that occurs when sensitive data is processed. In this work we are interested in the centralized model of differential privacy, where the algorithm whose privacy loss we want to bound is executed by a trusted curator with access to many agents’ private information and who must reveal their answer publicly.

In the theoretical study of the k-means problem, reducing the worst-case multiplicative approximation factor has been the focus of a major line of work (Kanungo et al. 2004; Ah- madian et al. 2020). However, even Lloyd’s algorithm (Lloyd

\*Equal contribution, names in no particular order.

Copyright oc 2021, Association for the Advancement of Artiﬁcial Intelligence (www.aaai.org). All rights reserved.

1982), which has a tight sub-optimal multiplicative guarantee of O(log k), works well in practice. This behaviour can be understood by showing (Aggarwal,Deshpande, and Kannan 2009) that Lloyd’s ﬁnds a solution with O(1) multiplicative error with constant probability, or that for a general class of datasets satisfying a certain separability condition (Ostrovsky et al. 2012) the multiplicative error again has a strong O(1) bound.

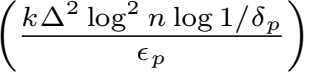
In contrast, when the algorithm is required to be (∈p , δp )- differentially private, no pure multiplicative approximation is attainable and additive error is necessary. This principle is formalised for the closely related discrete k-medians1 prob- lem in theorem 4.4 of Gupta et al. (2010) which shows that there is a family of instances whose optimal clustering cost is 0 but any differentially private algorithm must incur an Ω(△2 k(log n/k)/∈p ) expected cost. In practice, for many datasets it is seen that although the non-private clustering cost naturally decreases as the number of centers k increases, the costs incurred by differentially private algorithms quickly plateau (as in the experiments of Balcan et al. (2017)), sug- gesting that they have reached their limit in the additive error. Given this fundamental barrier, a major question is:

**Question:** Is it possible to obtain a ﬁnite approximation with additive error nearly linear ink?

**Contributions**

We introduce a differentially private k-means clustering al- gorithm for the global model of differential privacy. The additive error is nearly linear ink in contrast to a polynomial overhead in previous works, and the multiplicative error is constant, which is competitive with all previous works. The algorithm also exhibits an improvement experimentally over earlier work on synthetic and real-world datasets (Balcan et al. 2017). For a speciﬁc setting of parameters with constants ap- plicable for experiments, we have the following bound. More general bounds can be found in subsequent sections.

**Theorem 1.** *There is an* (∈p , δp ) *differentially private al- gorithm for the* k*-means problem that achieves a util-*

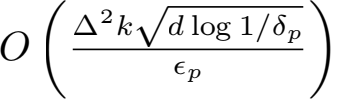
*ity bound of* O(1)fD ( OP TD ) + O  +

1The discrete k-medians problem is formulated similarly except the distance function is a metric, and the centers come from a public ﬁnite set and not the whole ambient space.

|  |  |
| --- | --- |
| **Reference** | |
| **Mult. Approx.** | **Add. Approx.** |

|  |  |
| --- | --- |
| Balcan et al. (2017) | |
| O (log3 n) | (k2 + d) |
| Stemmer and Kaplan (2018) | |
| O(1/√ ) | (k1.5 + d0.5+√ k1+√ ) |
| Jones, Nguyen, and Nguyen (2020) | |
| O(1/√ ) | (k + d0.5+√ k1+√ ) |
| This work | |
| O(1) | (k√d) |

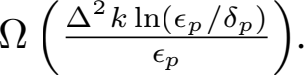
Table 1: Comparison with prior works where we omit all log terms and the common △2 factor in the additive error, the dependence on privacy parameters and set δp = 1/n1.5 .

*, where* D *is the input dataset,*

fD ( OP TD ) *is the optimal* k*-means cost for the input dataset* D*,* d *is the ambient dimension of* D*,* n *is the cardinality of* D*,* △ *is the diameter of* D*, and the failure probability of the algorithm is polynomially small in*n*.*

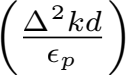
We extend the construction of Gupta et al. (2010) for the discrete k-medians problem to our setting and show that a linear dependence on kin the additive error is necessary for any ﬁnite multiplicative approximation.

**Theorem 2** (Informal)**.** *Any* (∈p , δp )*-differentially pri- vate algorithm must incur an expected additive error of*



The same construction also implies a lower bound for (∈p, 0)-differential privacy.

**Theorem 3** (Informal)**.** *Any* (∈p, 0)*-differentially private al-*

*gorithm must incur an expected additive cost of* Ω *.*

All full proofs maybe found in the supplementary material. We ﬁnish with an experimental evaluation of our algorithm, where we ﬁnd better performance than an implementation of previous work (Balcan et al. 2017).

**Related Work and Techniques**

In Gupta et al. (2010), the authors gave an algorithm for solv- ing the discrete k-medians problem and subsequent works focused on identifying good discretizations of the continuous domain to invoke their algorithm. A recent approach by Stem- mer and Kaplan (2018) uses locality sensitive hashing (LSH) to identify a small set of points that serve as potential cen- ters. Inherent in this approach is a trade-off between the multiplicative approximation and the size of this discrete set, which comes from the trade-off in LSH between the approx- imation and the number of hash functions. The number of candidate centers increases additive error and thereby causes

a~trade-off between the multiplicative (1/√ ) and additive

O(k + d0.5+√ k1+√ ) errors~. The work Jones, Nguyen, and Nguyen (2020) achieved O(k1+√ d0.5+√ ) additive error but the multiplicative error remained O(1/√ ).

In this work, we reduce the minimum additive error in this trade-off to nearly linear ink and also eliminate the resulting blow-up in the multiplicative factor using the most natural approach: discretizing the space using a grid and using all grid points as candidate centers. We reduce the data dimen- sions to O((log n)/∈2 ) and preserve all distances. However, there can be (n)log n many points in the grid that we construct since the grid size must start from 1/n for negligible additive error. It is not clear how to implement a selection algorithm (such as the exponential mechanism (McSherry and Talwar 2007)) on such a large number of choices. This hurdle,iden- tiﬁed in Balcan et al. (2017), prompted subsequent works to ﬁnd alternative approaches. Resolving it directly is our key contribution.

We observe that it is not inherently difﬁcult to sample uniformly among a large number of choices. Our task is non- trivial since the k-means cost objective is a complex function. To simplify the sampling weights, we exploit the connection between clustering and coverage and reduce the problem to ﬁnding maximum coverage: count the number of data points within a given radius of each candida~te center. The crucial ob- servation is that there are at most nO(1/∈2 ) grid points within the threshold radius of any data point, meaning that there are only a polynomial number of grid points with non-zero coverage. Thus, all but a polynomial number of choices have the same coverage of 0 making it possible to implement the exponential mechanism in polynomial time.

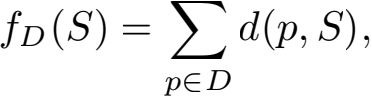
Given the implementation of the exponential mechanism for coverage, we follow the approach of Jones, Nguyen, and Nguyen (2020) to cover the points using clusters of increasing radii. Note that the approach goes back to the non-private coreset construction of Chen (2009). However, the use of coverage for dealing with each radius has another crucial advantage: as in Jones, Nguyen, and Nguyen (2020),by using the technique of Gupta et al. (2010), the privacy loss only increases by a log 1/δp factor even though the algorithm has Ω(k) adaptive rounds of exponential mechanism.

**Preliminaries**

We are given a dataset D of n points that lies in a ball B△/2(0) (the ball of radius △/2 centered at 0) in some high dimensional space Rd. The goal is to ﬁnd a set of k points S = {μ1, . . . , μk } such that Σp∈D d(p, S) is minimal. Here d(·, ·) : Rd × Rd → R is the square of the Euclidean dis-

tance, that is d(p, q) := Σ=1 (pi — qi )2 . Abusing notation,

d(p, S) := minμ∈S d(p, μ). Deﬁne



so when S is a set of size k, fD (S) is the k-means cost of the solution S for the dataset D.

Differential privacy is formalised as follows:

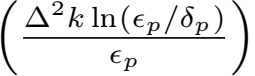
**Deﬁnition 4.** *Two datasets* D, D/ ∈ Xn *are called* neigh- bouring *if there is exactly one element in their symmetric difference. We say that an algorithm* A *is* (∈, δ)*-differentially private if for any two neighbouring datasets* D, D/ *and any*

*measurable output set* U *lying in the co-domain of* A*,* P(A(D) ∈ U) ≤ e∈ P(A(D/ ) ∈ U) + δ.

**Lower Bounds**

Following the construction in theorem 4.4 of Gupta et al. (2010), we derive lower bounds fork-means clustering in the (∈, δ) and (∈, 0)-differential privacy regimes.

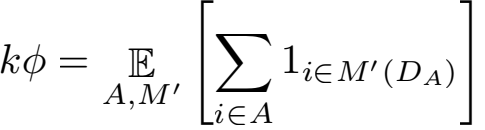
**Theorem 5.** *For any* 0 < ∈p , δp ≤ 1 *and integer* k*, there is a family of* k*-means instances over the cube* [0, △/√d]d *with* d = O(ln(k/(∈p δp ))) *dimensions such that the opti- mal clustering cost is* 0 *but any* (∈p , δp )*-differentially private*

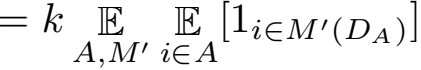
*algorithm would incur an expected cost of* Ω *.*

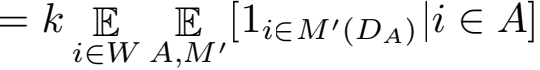
*Proof.* Let d = Θ(ln(k/((e∈p − 1)δp )) and W be the set of codewords of an error correcting code with constant rate and constant relative distance in {0, 1}d. The dimension d and codewords W are chosen so that |W | ≥ k/((e∈p − 1)δp ). Let L = ln((e∈p − 1)/(4δp ))/(2∈p ). Our input domain is [0, 1]d which has diameter △ = √d. Note that for other values of △, we can simply re-scale the construction.

Suppose M is an arbitrary (∈p , δp )-differentially private algorithm that on input D ⊂ [0, 1]d outputs a set of k lo- cations. Let M/ be the algorithm that ﬁrst runs M on the input and then snaps each output point to the nearest point in W; by post-processing, it has the same privacy guarantee. Furthermore, observe that if the input points are located at a subset of W then the cost of M/ is within a factor 4 of the cost of M. Let A be a size k subset of W chosen uniformly at random and the dataset DA be a multiset containing each point in A with multiplicity L. Note that the optimal cost for DA is 0.

We would like to analyze φ = EA,M, [|A ∩ M/ (DA )|]/k. We have:



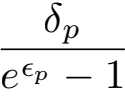




Let i/ be an random point in W not in A. Changing A to A/ = A \ {i} ∪ {i/ } requires changing 2L elements of DA . Notice that for random A \ {i} in W \ {i} and random i/ in W \ A, we have that A/ is still a uniformly random subset of W \ {i}. Thus,

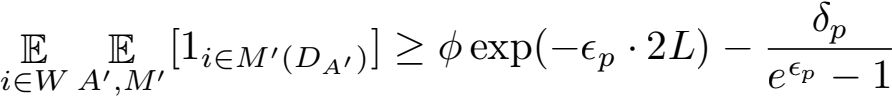
E E [1i∈M, (DA,) |i ∈/ A/]

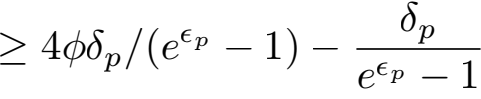
i∈W A,,M,

≥ (i  A, [1i∈M, (DA ) |i ∈ A]) e — ∈p ·2L − 

Here we use the fact that M/ is (∈p , δp )-differentially private, and that the δp losses in expectation decrease geometrically

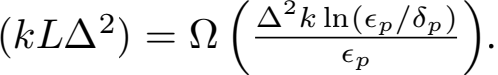
with factor exp(−∈p ) so the net leakage from the δ term can be lower bounded by the sum of an inﬁnite geometric progression. Continuing,





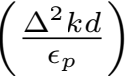
Since M/ (DA, ) has at most k points, the LHS is at most k/|W |. Thus, φ ≤ (k/|W | + δp /(e∈p − 1))/(4δp /(e∈p − 1)) ≤ 1/2.

For each point in A \ M/ (DA ), the algorithm incurs a cost of Θ(L△2 ) due to the multiplicity of L of points in DA and the fact that all points in W are at distance Θ(△) apart. The expected cost of M/ , and consequently the cost of M, is

hence Ω  ii

Using that |W | = 2Ω(d), and by setting L = ln(|W |/(2k))/(2∈p ), tracing the same proof one obtains a lower bound for (∈p, 0)-differential privacy.

**Theorem 6.** *For any* 0 < ∈p ≤ 1 *and integers* k *and* d = Ω(ln(k))*, there is a family of* k*-means instances over the cube* [0, △/√d]d *such that the optimal clustering cost is* 0 *but any* (∈p, 0)*-differentially private algorithm would incur*

*an expected cost of* Ω *.*

**Algorithm**

Our algorithm can be described in four steps.

**Step 1:** The dataset D ⊂ B(0, △/2) ⊂ Rd ispreprocessed via dimension reduction, scaling and projection to produce a dataset D/ ⊂ B1 (0) ⊂ Rd, where d/ = O((log n)/∈2 ). We let Gi be multi-dimensional grids of side lengthsti and observe that if μ is a center of a cluster with radius ≤ ri in the optimal solution, then by the triangle inequality a ball of radius ri + ti √d/ centered at lμ」(i) (the “ﬁoor" of μ the in grid) contains all the points of the same cluster.

**Step 2:** The threshold radii ri increase geometrically by a factor of (1 + ∈) from 1/n to 2; the unit length of grid Gi isti = ∈ri /√d. From Gi we choose candidate centers of clusters with radii in the interval [ri — 1 , ri ). This is done by counting the number of datapoints within ri + ti √d/ of every grid point Gi. We calculate a set of valid offsets Vi and, iterating over p ∈ D, increment counts for all grid points within an offset of lp」(i) . We use the exponential mechanism to greedily identify the set of k log「1/∈l best grid points Ci that attains close to optimal coverage. The candidate set C is the union of C1 , C2, . . . , Clog 1+∈ 2/(1/n) .

**Step 3:** We want to construct a proxy dataset D// by moving each datapoint in D/ to its closest center in C. To maintain privacy, we compute the count nc of datapoints whose closest center is c and add Laplace noise to get c. D// then contains c copies of c for all c ∈ C.

|  |  |
| --- | --- |
| **Algorithm 1:** Private k-means | |
| **Data:** D C Rd~dataset,~|D| = n.  **Result:** S = {μ1, . . . , μk } C Rd  T ~ JohnsonLindenstrauss(n, ∈); // Step 1  D/ ← T(D)  d/ ← dim(T) = O((log n)/∈2 )  Scale D/  B1 (0)  Let T/ be the composition of T with the scaling and projection so that T/ (D) = D/  r1 ← 1/n ; // Step 2  t1 ← ∈/(n√d/ )  **for** i = 1, . . . , m =「log1+∈ 2nl **do**  Ci ← algorithm 2(D/ , ti , ri )  ri+1 ← (1 + ∈)ri. ti+1 ← (1 + ∈)ti.  **end**  D/ ← T/ (D) ; // Step 3  C = u Ci  Assignallp ∈ D/ to the closest point C, denoted grid[p]  Let nc be the number of points in D/ assigned to c  For each c ∈ C set n = nc + Lap  Let D// be the dataset where every c ∈ C is repeated  n times  S// = {μ , . . . , μ} ← Lloyd(D// ) ; // Step 4  D ← {p ∈ D/ : arg minμ// ∈S// d(p, μ// ) = μ} for  i = 1, . . . , k  **for** i = 1, . . . , k **do** | |
|  | i = algorithm 3(D, 1D , ∈G , δG ) ; // 1D (p) indicates whether T/ (p) ∈ D for  p ∈ D |
| **end**  ~  **return** S = | |

**Step 4:** In the ﬁnal step we apply any non-private k-means clustering algorithm to D// to get some cluster centers S// . We cluster D/ using these cluster centers to get clusters C/ , and deﬁne ﬁnal clusters for D by identifying points with their images under the projection and re-scaling. To stay private we use NoisyAVG (Nissim, Stemmer, and Vadhan 2016) to derive centers by averaging over cluster sets.

The formal pseudocode requires some additional justiﬁca- tion; the construction of the offset set Vi , and the polynomial time implementation of the exponential mechanism.

**Lemma 7.** *A data point* p *is within distance* ri + ti √d/ *of*

*a grid point* tib *for* b ∈ Zd/ *only if* Σ1 min((lp」j(i) —

tibj )2 , (lp」j(i) — ti (bj + 1))2 ) ≤ (ri + ti √d/ )2*. Let* Vi =

{v : v ∈ Nd/ , Σ1 tv < (ri + ti √d/ )2 }*. If* d(p, tib) <

(ri + ti √d/ )2 *then for some* s ∈ {0, 1}d *and* v ∈ Vi*,* tib = lp」(i) + tis + (2s — ti , 1, . . . , 1)*.*

*Sketch of proof.* For any real number, either its ﬁoor or its

ceil is closer to a given integer than it is. Applying this prin- ciple coordinate-wise in the grid, we see that a point can lie within a distance ri + ti √d/ of a given grid point only if the vertex of the grid unit cube closest to that grid point were also to lie within a distance of ri + ti √d/ . The second half of the claim follows by noting that the expression tis + (2s — 1)tiv is exactly the closest vertex of the grid unit cube containing p to tib. 

**Lemma 8.** *After computing the cover of each grid point, algorithm 2 executes the exponential mechanism correctly and in polynomial time.*

*Proof.* We know that for any data point the only grid points

whose co˜ver must be updated lie in Vi. It sufﬁces to show

|Vi | < nO(1/∈2 ) . The number of ordered tuples v ∈ Nd/ for

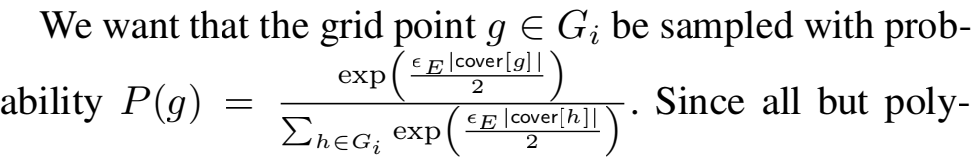
whichΣj tv < (ri + ti √d/ )2 ⇔ Σj v < d/ (  + 1)2 ,

equals the number of ways of partitioning d/ (  + 1)2 +

d/ + 1 balls into d/ + 1 distinguishable bins. It follows that

|V | = 2d/ (d/ (  d/ +1) < 2d/ ( ed/(+)21+d/+1 )d/ +1 =

2d/ O(1/∈2 )d/ +1 = nO((1/∈2 ) log 1/∈), using that d/ = O ((log n)/∈2 ).



nomially many grid points {g : cover[g] = 0} are being sampled with the smallest probability any point is sampled with, we can use the law of total probability to write this sampling distribution as a uniform distribution on the entire grid with some probability 1 — Psamp , and a second distri- bution with P/ supported only on the polynomially many grid points with non-zero cover with probability Psamp , i.e.

P(g) = PsampP/ (g) + (1 — Psamp ) 

point with 0 cover for which P/ (g) = 0, one derives the necessary expression for Psamp.

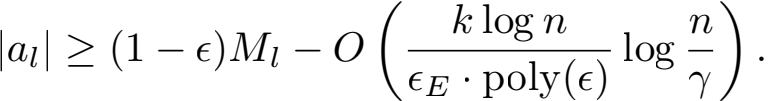
**Utility**

To derive a bound for the utility attained by algorithm 1, we have three steps; ﬁrst we show that the set C constructed by choosing points from the grid contains a discretized version of any optimal k-means solution with high probability. Sec- ond, we show that a k-means solution for the proxy dataset constructed using C works well for the dimension reduced dataset D/ . Third, we derive cluster centers for the original dataset D by taking the average of all datapoints in each cluster.

The analysis of the ﬁrst step proceeds as in Jones, Nguyen, and Nguyen (2020). We letoi = {p ∈ D/ : d(p, OPTD/ ) ∈ [ri-1, ri )} and ai = D/ ∩ Bri +ti √d/ (Ci ), where Ci is the set of grid points selected from Gi in the ith call to algo- rithm 2. The ith call to algorithm 2 would be successful if the grid points Ci C Gi returned cover close to the maximum possible.

**Lemma 9.** *If* Ml *is the maximum number of points that can be covered within distance* rl + tl √d/ *of* k *grid points in* Gl*, then with probability* 1 − √

|  |
| --- |
| **Algorithm 2:** Private grid set cover |
| **Data:** D/ dataset (passed by reference), ti grid unit length, ri threshold radius  **Result:** set Ci ⊂ Gi  Ci ← ∅  **repeat** k/ **times**  cover ← empty linked list  Vi ← {v : v ∈ Nd, , Σ=, 1 (ti vj )2 <  (ri + ti √d/ )2 }  **for** *all* p ∈ D/ **do**  **for** *all* v ∈ Vi **do**  **for** *all* s ∈ {0, 1}d, **do**  tib = lp」(i) + tis + (2s − tiv ;  // where is the  all-ones vector  **if** d(tib, p) < (ri + ti √d/ )2 **then**  cover[tib] += {p}  **end**  **end**  **end**  **end**  totalCover ← 0 **for** g ∈ cover **do**    **end**  totalCover += |Gi | − len[cover] Let Psamp = 1 −  **if** *Ber* (Psamp ) = 1 **then**  .p. ∼ P  **else**  g ← uniformly at random from Gi  **end**  Ci ← Ci ∪ {g}  D/ ← D/ \cover[g]  **end**  **return** Ci |

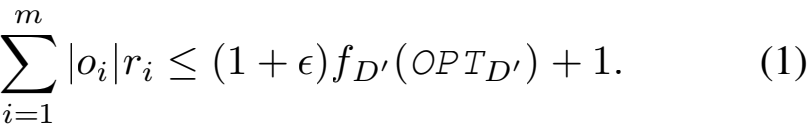


*where* ∈E *is the privacy parameter used in the exponential mechanism.*

*Sketch of proof.* The set cover function that counts the num- ber of datapoints that lie within ri + ti √d/ within any set of grid points Ci is submodular. It follows that greedily picking points by maximising the marginal increase in cover leads to covering (1 − ∈) as many points as the maximum, provided we pick O(log「1/∈l) as many grid points than there are in the optimal solution. Calls to the exponential mechanism lead to covers within logarithmic loss of the maximum and after accounting for these losses we get the stated bound.

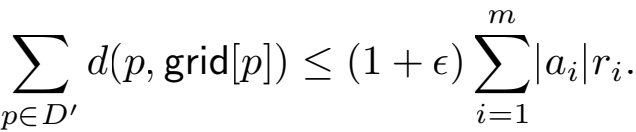
To use the fact that the number of datapoints covered in the ith call to algorithm 2 |ai | is close to |oi |, we bound the optimal total movement of points when mapping each datapoint p ∈ D/ to its closest candidate grid point g ∈ C in terms of the optimal clustering cost.

**Lemma 10.** *The thresholded cost obeys the bound*



Similarly, we can bound the actual increase in cost incurred from total distance moved by datapoints when constructing the proxy dataset D/ in terms of ai.

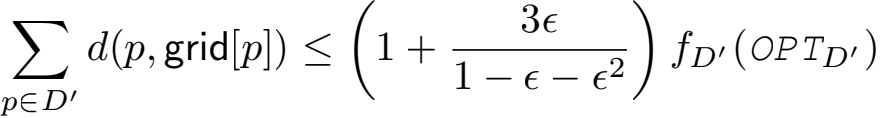
**Lemma 11.** *The total movement of points* p ∈ D/ *to the closest point* grid[p] ∈ C *is can be bounded in terms of the* ai *as follows:*

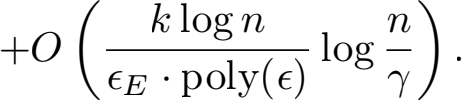


From the previous two lemmata and by deriving a rela- |ai |ri |oi |ri , we can complete the ﬁrst step of the proof.

**Lemma 12.** *The total movement of points* p ∈ D/ *to the closest point* grid[p] ∈ C *can be bounded in terms of the optimal cost as follows:*

|  |
| --- |
| **Algorithm 3:** NoisyAVG(Nissim, Stemmer, and Vad- han 2016, Algorithm 5) |
| **Data:** Multiset V of vectors in Rd , predicate g, parameters ∈, δ  Set  = |{v ∈ V : g(v) = 1}| + Lap(5/∈) −  ln(2/δ).  If < 0, output a uniformly random point in the  domain B△/2(0).  Denote σ = 2 ln(3.5/δ), and let η ∈ Rd be a  random noise vector with each coordinate sampled  independently from N(0, σ2 ).  **return** g(V) + η |



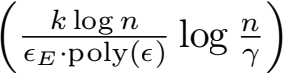


*Proof.* Let Oi = Σi |oj | and Ai = Σi |aj |. Then Σ|ai |ri = Σ Ai (ri − ri — 1 ). Centers in OPTD, cover

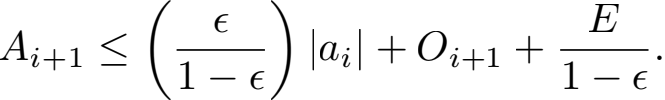
n − Oi+1 points at a maximum distance of ri. We also know that algorithm 1 has already covered n − Ai points at a dis- tance of ri — 1 + ti — 1 √d/ . It then follows that there are some k grid points in Gi (snapping the centers in OPTD, to grid)

that cover at least Ai - Oi+1 points in oi. From the lemma 9 guarantee, we know

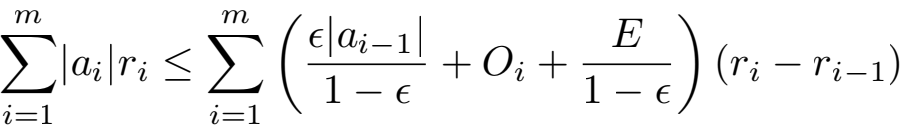
jai j ≥ (1 - ∈)(Ai - Oi+1) - E,

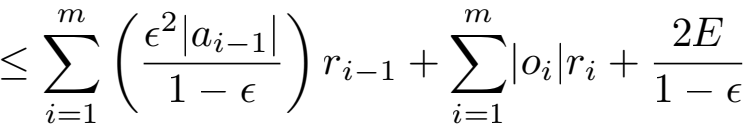
where E = O  . Since Ai = jai j + Ai+1 ,

we have that

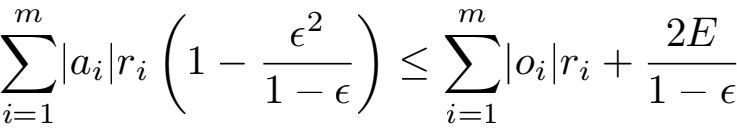


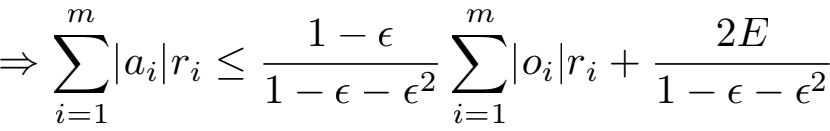
Substituting this in the telescoping Σ Ai (ri - ri — 1 ),





where to get from the second to the third line we use that ri - ri — 1 = ∈ri — 1 , and that rm - r0 = 2. Rearranging, we get



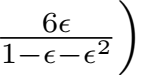
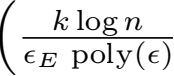
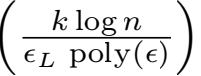


Substituting the order term E, using that ∈ is bounded away from 1 and applying the previous two lemmata we get the

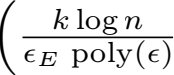
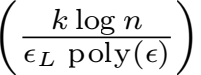
desired inequality.

In lemma 13 we bound the k-means cost of the proxy dataset D// in terms of the k-means cost of D/ . Doing so requires us to account for the noisy counts used to construct the proxy dataset; this leads to the additional O( ) error term. The proof of this step essentially follows from two applications of the triangle inequality, where since d is not a true metric we must gain a factor of 2 in the multiplicative loss for every application.

**Lemma 13.** *With probability* 1 - √*,* fD// ( OP TD/ ) *is at*

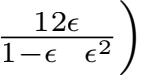
*most* (4 +  fD/ ( OP TD/ ) + O  log  + O *.*

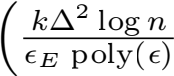
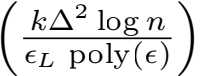
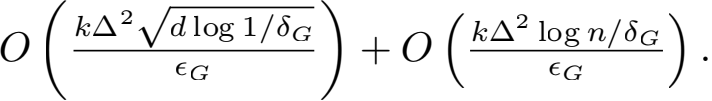
In lemma 14, by essentially applying the triangle inequal- ity, we bound the cost incurred by using the non-private k-means clustering solution for the proxy dataset D// for the dataset D/ completing the second step of the analysis.

**Lemma 14.** *Let* A *be the clustering algorithm used in algo- rithm 1 of algorithm 1. If* A *has amultiplicative loss of* EM *, then* fD/ (A(D// )) ≤ (8EM +2+(8EM +4)∈)fD/ ( OP TD/ )+ O  log  + O *.*

To complete the utility analysis, we need to account for the projection and scaling as well as the Gaussian noise added to maintain privacy. In theorem 15 we account for the scaling and projection and then add the cost incurred due to the privacy preserving NoisyAVG of (Nissim, Stemmer, and Vadhan 2016). This gives us an upper bound for the net cost incurred by algorithm 1.

**Theorem 15.** *Algorithm 1 returns a set of points*

*such that* E [fD ()] ≤ (1 + ∈) (8 + — (EM +

1)fD ( OP TD ) + O  log  + O  + 

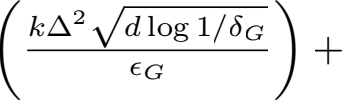
*Sketch of proof.* The scaling factor was picked according to the Johnson-Lindenstrauss (JL) transform to ensure that

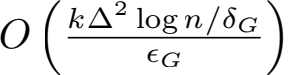
with high probability nothing needs to be projected, so we only need account for the re-scaling. We recall that the k- means clustering cost can be expressed only using the clus- ter sets D1, . . . , Dk via the expressionΣp∈D/ d(p, S) =

Σ=1 Σpq∈jjd(p,q) . As the JL transform preserves the

square of the Euclidean distance within a factor of (1 + ∈), the k-means cost can increase by a factor of at most 1 + ∈ when using the same clusters for D as were found in D/ .

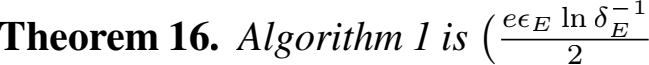
The ﬁnal centers are computed using the NoisyAVG sub- routine of Nissim, Stemmer, and Vadhan (2016). NoisyAVG modiﬁes the well-known Gaussian mechanism by using a noisy version of the cluster size since in this application the cluster size is also private. For large clusters the noisy count does not increase the variance too much and for small clusters the worst-case cost is dominated by other error

terms. This noisy averaging adds the O 

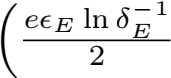
 term to the clustering cost. ii

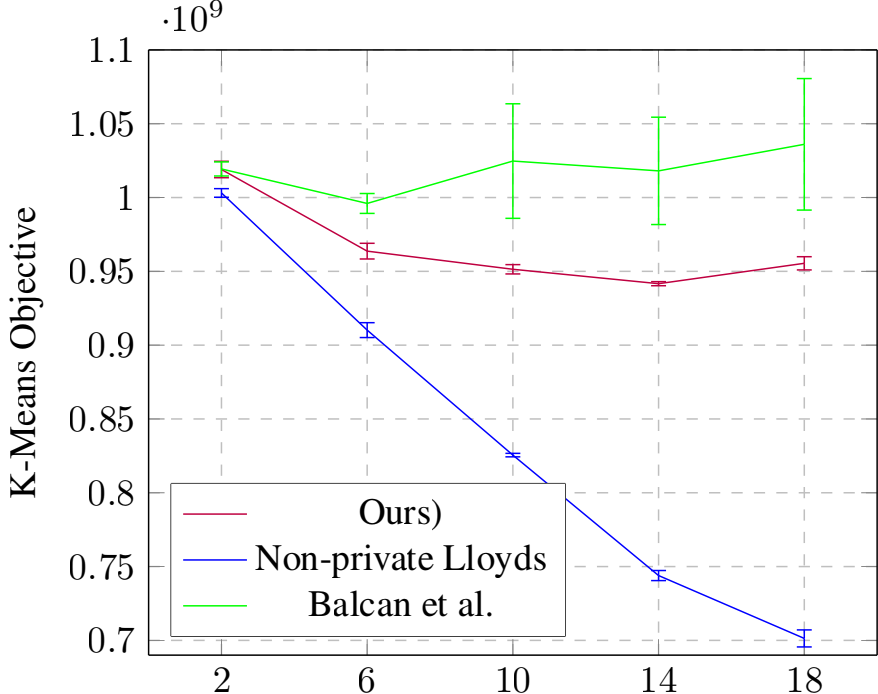
**Privacy**

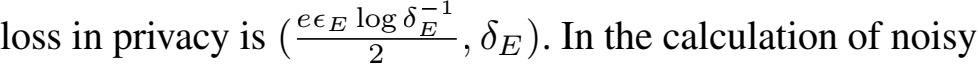
The main result of this section is the following:

 +∈L +∈G , δE +δG )*- differentially private.*

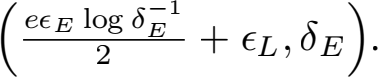
From the basic and parallel composition laws of differen- tial privacy and the privacy guarantees of the Laplace mecha- nism and algorithm 3 most of the expression for the bound on privacy loss claimed in this result follows relatively straight- forwardly. To bound the privacy loss incurred in the calls to algorithm 2, we adapt a technique from Gupta et al. (2010). We use this technique in the following lemma to show that the privacy loss when using the exponential mechanism many times successively can be bounded as an expression of the sum of expected gains in the cover. For the set cover function this sum of expected gains can be shown to decay exponen- tially, which leads to a strong bound on the privacy loss.

**Lemma 17.** *The* m *calls to algorithm 2 from algo- rithm 1 that construct the set of centers* C *are collectively*   , δE )*-differentially private.*

Synthetic Dataset ∈ = 1

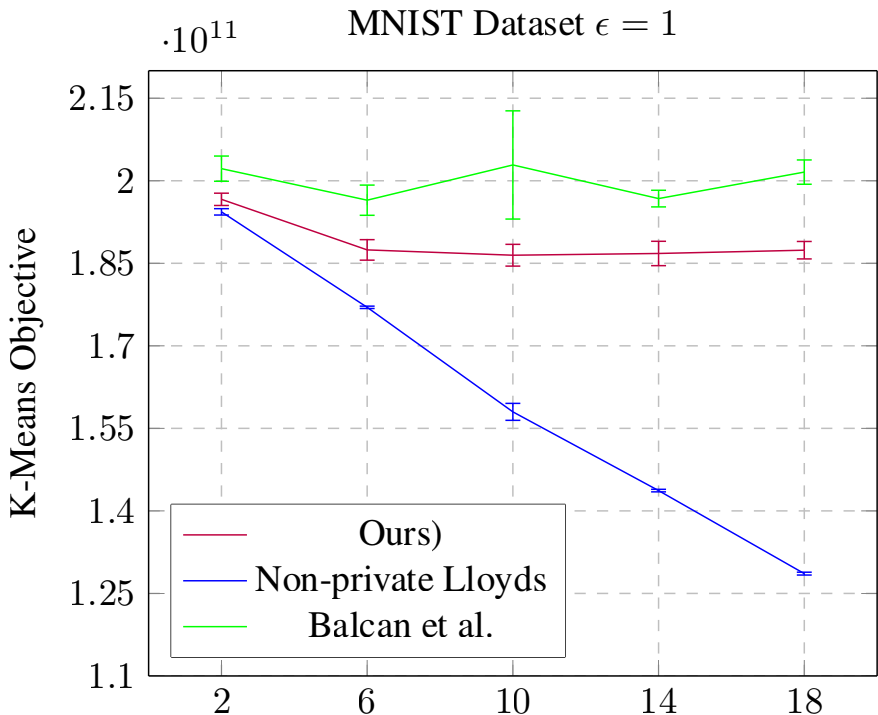
*Proof of theorem 16.* First, we bound the loss in privacy that occurs when constructing the proxy dataset D,, . From lemma 17 we know that in them calls to algorithm 2 the net 

counts we see that two neighbouring datasets can only differ in their true counts by 1 unit at one center of C. It follows that the l1 sensitivity of the tuple of all counts is 1 unit; this justiﬁes the choice of parameter in the Laplace mechanism. Using basic composition along with the privacy loss bound for the Laplace mechanism we see that the net loss in privacy

on releasing the proxy dataset D,, is 

Now D,, is publicly known and the low-dimensional do- main can be partitioned by identifying each point in the domain with the closest point in the set returned by the non- private clustering algorithm used (a Voronoi diagram). To bound any further loss in privacy we use the parallel composi- tion theorem of McSherry (2010) along with the privacy guar- antee of algorithm 3. Since each application of algorithm 3 on the separate clusters is (∈G , δG )-differentially private, by parallel composition the net privacy loss over all k applica- tions is still (∈G , δG ). By basic composition the stated result

Centers



follows.

**Experiments**

We present an experimental comparison between algo- rithm 12 , the private k-means clustering algorithm from Bal- can et al. (2017), and the non-private Lloyd’s algorithm. We are not aware of any other private clustering methods which have been implemented. Two datasets are used; a synthetic dataset reproducing the construction in Balcan et al. (2017) and the MNIST training dataset (Lecun et al. 1998).

Centers

The empirical results shown here for Balcan et al.’s algo- rithm (Balcan et al. 2017) come largely from their MATLAB implementation available on Github. Some corrections were made to the implementation of Balcan et al. (2017); although the pseudocode uses a noisy count of the cluster sizes when computing the noisy average of the clusters found their im- plementation used the non-private exact count. We replaced this subroutine with algorithm 3 to use the best method we know for privately computing the average.

**Implementation details:** We set ∈ = 1 and δ = n-1.5 for both algorithms. Similar to Balcan et al. (2017), we project to a smaller subspace of dimension log(n)/2 rather than O(log(n)/∈2 ) - this does not affect privacy. At the conclu- sion of both algorithms, we run one round of differentially private Lloyd’s algorithm; adding this call to the differen- tially private Lloyd’s yielded better empirical results for both algorithms. The addition of these rounds of Lloyd’s requires adjusting privacy parameters by a constant factor but other- wise does not affect the privacy guarantees of the original algorithms. Being (∈, 0)-private, Balcan et al. (2017) use the Laplace mechanism for their noisy average which we re- placed with the noisyAVG routine of Nissim, Stemmer, and

Figure 1: Empirical comparison of algorithm 1 and the pri- vate k-means clustering algorithm from (Balcan et al. 2017). Averages and standard deviations computed over 5 runs.

Vadhan (2016) for a comparison in the (∈, δ)-regime. Lloyd’s algorithm was executed with 10 iterations.

**Datasets:** The synthetic dataset is comprised of 50,000 points randomly sampled from a mixture of 64 Gaussians in R100 . The MNIST training dataset uses the raw pixels; it is comprised of 60,000 points with 784 features each.

**Results:** As can be seen in ﬁg. 1, our algorithm achieves a lower k-means objective score for both datasets. Similar to the experimental results in Balcan et al. (2017), increasing the number of centers results in a decrease in the cost in the non-private algorithm but did not result in a concomitant decrease in the cost of the private algorithms. This behavior suggests that these algorithms are limited by their additive error and that perhaps further decreasing even the constants in the additive error would improve the gap between them and their non-private counterparts.

2The code used for our experiments is available at https://github. com/Anamay-Chaturvedi/Differentially-private-k-means

**Acknowledgements**

This material is based upon work supported by the National Science Foundation under NSF grants NSF AF 1909314 and NSF CAREER 1750716. Any opinions, ﬁndings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reﬁect the views of the National Science Foundation.

**References**

Aggarwal, A.; Deshpande, A.; and Kannan, R. 2009. Adap- tive Sampling fork-Means Clustering. In Dinur, I.; Jansen, K.; Naor, J.; and Rolim, J. D. P., eds., *Approximation, Ran- domization, and Combinatorial Optimization. Algorithms and Techniques, 12th International Workshop, APPROX 2009, and 13th International Workshop, RANDOM 2009, Berkeley, CA, USA, August 21-23, 2009. Proceedings*, vol- ume 5687 of *Lecture Notes in Computer Science*, 15—28. Springer. doi:10.1007/978-3-642-03685-9\\_2. URL https: //doi.org/10.1007/978-3-642-03685-9\\_2.

Ahmadian, S.; Norouzi-Fard, A.; Svensson, O.; and Ward, J. 2020. Better Guarantees for k-Means and Euclidean k-Median by Primal-Dual Algorithms. *SIAM J. Comput.* 49(4). doi:10.1137/18M1171321. URL <https://doi.org/10>. 1137/18M1171321.

Balcan, M.; Dick, T.; Liang, Y.; Mou, W.; and Zhang, H. 2017. Differentially Private Clustering in High-Dimensional Euclidean Spaces. In Precup, D.; and Teh, Y. W., eds., *Pro- ceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Re- search*, 322—331. PMLR. URL <http://proceedings.mlr.press/> v70/balcan17a.html.

Chen, K. 2009. On Coresets fork-Median and k-Means Clus- tering in Metric and Euclidean Spaces and Their Applications. *SIAM J. Comput.* 39(3): 923—947. doi:10.1137/070699007. URL <https://doi.org/10.1137/070699007>.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. D. 2016. Calibrating Noise to Sensitivity in Private Data Analysis. *J. Priv. Conﬁdentiality* 7(3): 17—51. doi:10.29012/jpc.v7i3.405. URL <https://doi.org/10.29012/jpc.v7i3.405>.

Gupta, A.; Ligett, K.; McSherry, F.; Roth, A.; and Talwar, K. 2010. Differentially Private Combinatorial Optimization. In Charikar, M., ed., *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, 1106—1125. SIAM. doi:10.1137/1.9781611973075.90. URL <https://doi.org/10>. 1137/1.9781611973075.90.

Jones, M.; Nguyen, H. L.; and Nguyen, T. 2020. Differ- entially Private Clustering via Maximum Coverage. *CoRR* abs/2008.12388. URL <https://arxiv.org/abs/2008.12388>.

Kanungo, T.; Mount, D. M.; Netanyahu, N. S.; Piatko, C. D.; Silverman, R.; and Wu, A. Y. 2004. A local search approx- imation algorithm fork-means clustering. *Comput. Geom.* 28(2-3): 89—112. doi:10.1016/j.comgeo.2004.03.003. URL <https://doi.org/10.1016/j.comgeo.2004.03.003>.

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11): 2278—2324.

Lloyd, S. P. 1982. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* 28(2): 129—136. doi:10.1109/TIT.1982.

1056489. URL <https://doi.org/10.1109/TIT.1982.1056489>. McSherry, F. 2010. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. *Commun. ACM* 53(9): 89—97. doi:10.1145/1810891.1810916. URL https: //doi.org/10.1145/1810891.1810916.

McSherry, F.; and Talwar, K. 2007. Mechanism Design via Differential Privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20- 23, 2007, Providence, RI, USA, Proceedings*, 94—103. IEEE Computer Society. doi:10.1109/FOCS.2007.41. URL https: //doi.org/10.1109/FOCS.2007.41.

Nissim, K.; Stemmer, U.; and Vadhan, S. P. 2016. Lo- cating a Small Cluster Privately. In Milo, T.; and Tan, W., eds., *Proceedings of the 35th ACM SIGMOD-SIGACT- SIGAI Symposium on Principles of Database Systems, PODS 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, 413—427. ACM. doi:10.1145/2902251.2902296. URL <https://doi.org/10.1145/2902251.2902296>.

Ostrovsky, R.; Rabani, Y.; Schulman, L. J.; and Swamy, C. 2012. The effectiveness of lloyd-type methods for the k- means problem. *J. ACM* 59(6): 28:1—28:22. doi:10.1145/ 2395116.2395117. URL <https://doi.org/10.1145/2395116>. 2395117.

Stemmer, U.; and Kaplan, H. 2018. Differentially Private k-Means with Constant Multiplicative Error. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa- Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 5436—5446. URL <http://papers.nips.cc/paper/7788-differentially-private>- k-means-with-constant-multiplicative-error.