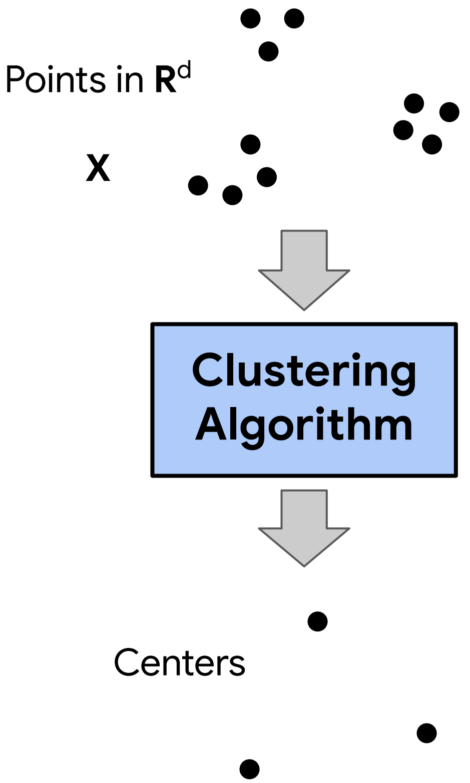
Locally Private k-Means in One Round

Alisa Chang, Badih Ghazi, Ravi Kumar, Pasin Manurangsi

Google

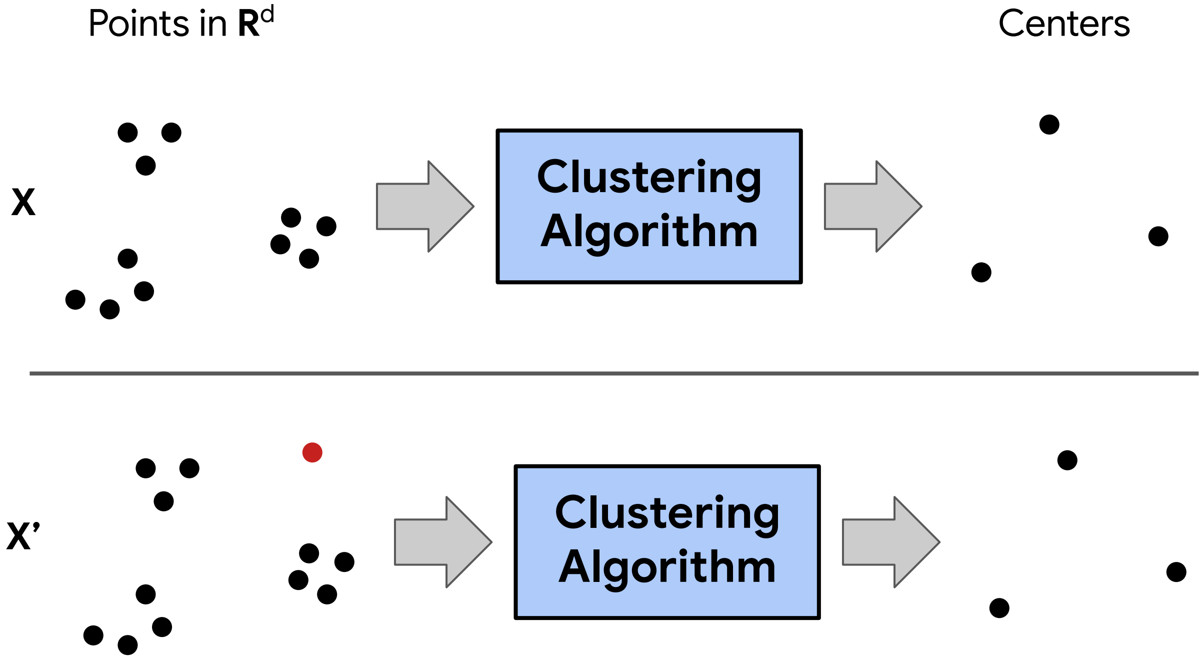
Private Clustering



k-Means

Given n points $x_1, \ldots, x_n$ in B_d and k, find $c_1, \ldots, c_k$ that minimize $\Sigma_{i\in[n]} \min_{j\in[k]} ||x_i - c_j||^2$

Clustering in Central DP

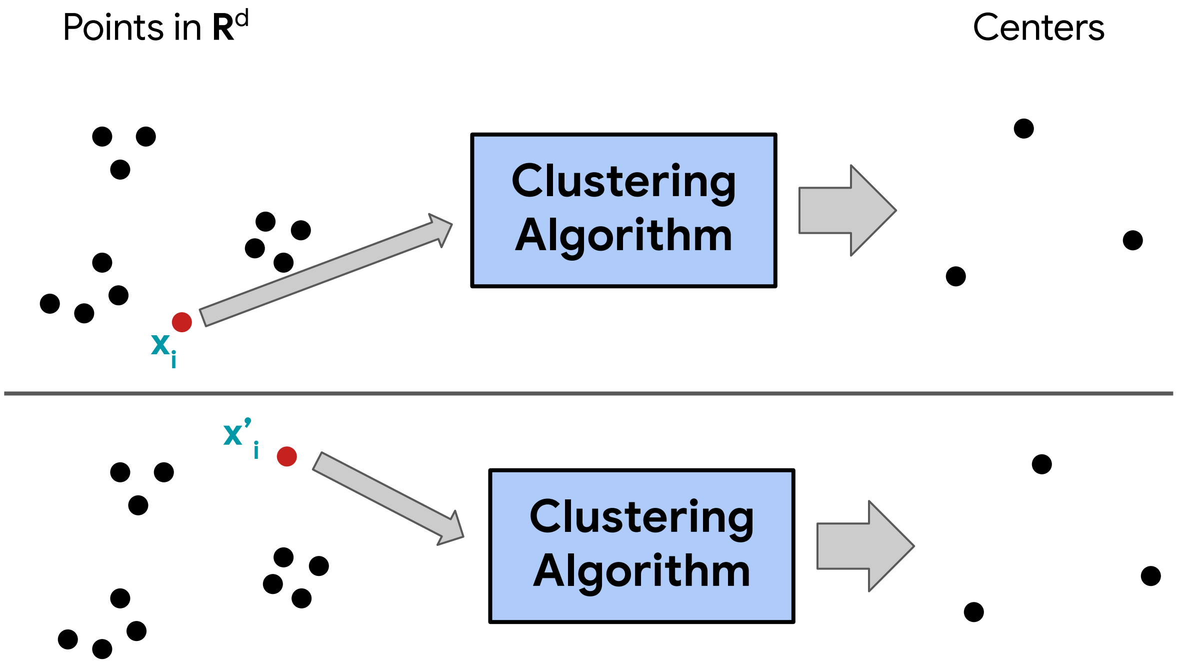


(ε, δ)-Differential Privacy

[Dwork et al.'06]

For every set of outputs S,
$\Pr[A(X) \in S] \le e^{\varepsilon} \cdot \Pr[A(X') \in S] + \delta$

Clustering in Local DP



ε-Local DP

[Kasiviswanathan et al.'08]

For every set of outputs S,

$\Pr[A(x_i) \in S] \leq e^{\varepsilon} \cdot \Pr[A(x'_i) \in S]$

Prior Works & Our Results

Approximation Ratio

	Non-Private	Central DP	Central DP	Local DP	One-Round Local DP
k-Means	6.359 *[Ahmadian et al.'17]*	O(1) *[Kaplan-Stemmer'18]*	6.359 *[GKM'20]*	O(1) *[Stemmer'20]*	6.359 *[our work]*

[GKM20]
"Private approx ratio ≈ non-private approx ratio"

$\underbrace{\text{Central DP} \quad \text{Central DP}}_{\text{Additive errors} \approx \text{poly}(dk \log n)}$

$\underbrace{\text{Local DP} \quad \text{One-Round Local DP}}_{\text{Additive errors} \approx \sqrt{n} \cdot \text{poly}(dk \log n)}$

$\boxed{\varepsilon\text{-local DP poly-time algo: } w^*(1 + \gamma)\text{-approx, add. error } \sqrt{n} \cdot \text{poly}(dk \log n)}$

w^* = best non-private approximation ratio

Framework of *[GKM'20]*

Random Projection

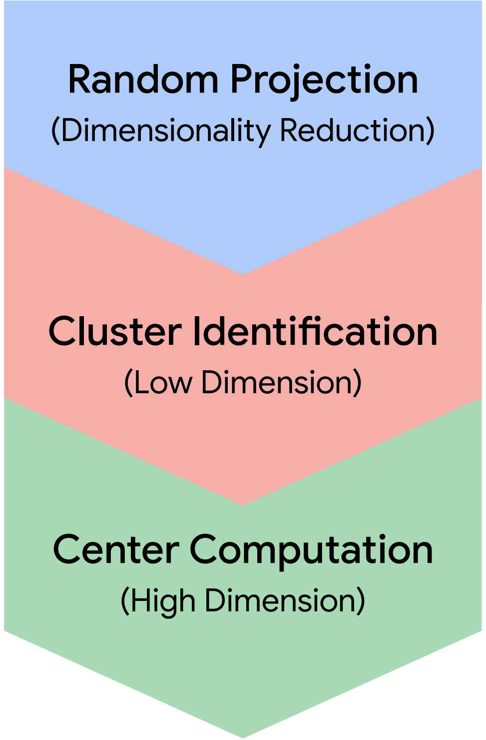
(Dimensionality Reduction)

Framework of *[GKM’20]*

Random Projection
(Dimensionality Reduction)

Cluster Identification
(Low Dimension)

Framework of *[GKM'20]*



Our Algorithm

Random Projection
(Dimensionality Reduction)

Cluster Identification
(Low Dimension)

Center Computation
(High Dimension)

Project to d' ≈ log(k) dimensions

[Makarychev–Makarychev -Razenshteyn'19]
Suffices to find $(1 + 0.5\gamma)w^*$-approx in lower dimension

k-Means

ε-local DP poly-time algo: $w^*(1 + \gamma)$-approx, add. error $\sqrt{n} \cdot \mathrm{poly}(dk \log n)$

w^* = best non-private approximation ratio

Our Algorithm

$d' \approx \log(k)$

Random Projection
(Dimensionality Reduction)

Cluster Identification
(Low Dimension)

Center Computation
(High Dimension)

Private Coreset *[Our work]*
ε-DP algo that produces a set of points Y such that, for any set Z of k centers,
$\mathrm{cost}_Z(Y) = (1 \pm 0.1\gamma) \cdot \mathrm{cost}_Z(X) \pm \sqrt{n}\, \mathrm{poly}(k) \cdot 2^{O(d')}$

Run best non-private approx algo on Y

Clusters $C_1, \ldots, C_k \subseteq X$

k-Means

ε-local DP poly-time algo: $w^*(1 + \gamma)$-approx, add. error $\sqrt{n} \cdot \mathrm{poly}(dk \log n)$

w^* = best non-private approximation ratio

Our Algorithm

Random Projection
(Dimensionality Reduction)

Cluster Identification
(Low Dimension)

Center Computation
(High Dimension)

Coreset Construction

$d' \approx \log(k)$

Net Tree *[Har-Peled&Mendel'06]*

- Hierarchical partitioning of the space
- Each level is a finer and finer net
  - # of children = $2^{O(d')}$ = poly(k)

k-Means

ε-local DP poly-time algo: $w^*(1 + \gamma)$-approx, add. error $\sqrt{n} \cdot \mathrm{poly}(dk \log n)$

w^* = best non-private approximation ratio

Our Algorithm

Random Projection
(Dimensionality Reduction)

Cluster Identification
(Low Dimension)

Center Computation
(High Dimension)

Coreset Construction

d’ ≈ log(k)

Net Tree *[Har-Peled&Mendel’06]*

- Hierarchical partitioning of the space
- Each level is a finer and finer net
    - # of children = $2^{O(d')}$ = poly(k)

Sketching ⇒ estimate # points in each node

Algorithm Outline:

- Start at the root
- At each level, explore τ = poly(k)·$2^{O(d')}$ nodes with largest # of points
- Each node has one “representative” in coreset

k-Means

ε-local DP poly-time algo: w*(1 + γ)-approx, add. error √n·poly(dk log n)

w* = best non-private approximation ratio

Our Algorithm

Random Projection
(Dimensionality Reduction)

Cluster Identification
(Low Dimension)

Center Computation
(High Dimension)

k-means center = average of all points in nodes mapped to that partition

↓

Sketching technique +
[Duchi-Jordan-Wainwright'13]
Estimate average of points assigned to each node explored

↓

Estimated cluster center = weighted average

k-Means

ε-local DP poly-time algo: $w^*(1 + \gamma)$-approx, add. error $\sqrt{n} \cdot \mathrm{poly}(dk \log n)$

w^* = best non-private approximation ratio

Conclusions

- One-Round Local Model: DP Clustering Algo for k-Means
  - Approximation ratio arbitrarily close to non-private
  - Additive error: $\sqrt{nd} \cdot \mathrm{poly}(k \log n)$
- Shuffle Model: DP Clustering Algo for k-Means
  - Approximation ratio arbitrarily close to non-private
  - Additive error: $\mathrm{poly}(dk \log n)$

Open Questions

- *Local Model*
  - k-median: O(1)-approx with add. error $n^{0.51}$ in O(1) rounds *[Stemmer'20]*
  - Densest ball: O(1)-approx with add. error $n^{2/3+o(1)}$ in O(1) rounds *[Kaplan-Stemmer'18]*
  - One Round? Tight Approximation Ratio? Tight Additive Error for Densest Ball?