Coresets for Clustering Problems

Anup Bhattacharya
Indian Statistical Institute, Kolkata

Basics of Coresets

- Small, weighted summary of the input.
- Given an unweighted (possibly weighted) dataset and some computational problem on this dataset, compute a small summary such that the summary approximates the dataset well for that task.

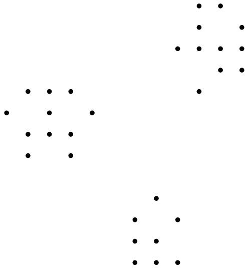
k-means Clustering Problem

- Input: Dataset $X \subseteq \mathbb{R}^d$, and integer k.
- Cost function: For $C \subseteq \mathbb{R}^d, |C| = k$, $\Phi(X, C) = \sum_{x \in X} \min_{c \in C} ||x - c||^2$.

- Objective: Find set $C \subseteq \mathbb{R}^d$ of k centers that minimizes $\Phi(X, C)$.

Coresets for *k*-means

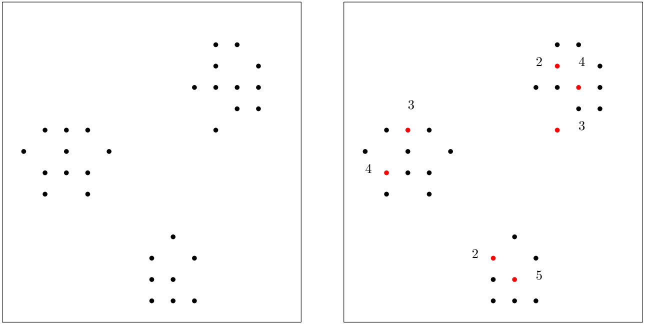
- Coresets to *approximate the dataset well* for k-means.



Coresets for *k*-means

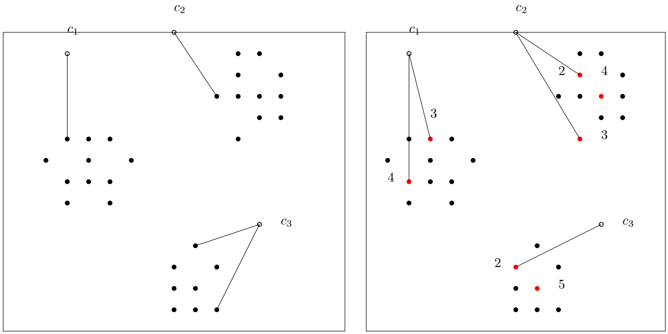
- Coresets to *approximate the dataset well* for k-means.

- How to guarantee coresets approximate dataset well.
- Coresets approximate dataset with respect to k-means objective function.



Coresets for k-means

- Approximates the objective function for input dataset simultaneously for all queries.
- Query for k-means: Cost of k-means objective function with respect to set of k centers.



Basics of Coreset

- Let X be a dataset with non-negative weights $\mu_X(x)$.
- Let $\mathcal{Q}$ be set of possible queries or solutions.
- Weighted set S is an ε-coreset of X if for all $Q \in \mathcal{Q}$,

$$(1-\varepsilon)\text{cost}(X, Q) \leq \text{cost}(S, Q) \leq (1+\varepsilon)\text{cost}(X, Q)$$

Coresets for k-means

(k, ε)-Coreset for k-means

- [HPM2004] Given a point set $X \subseteq \mathbb{R}^d$, a weighted subset $S \subseteq X$ is a (k, ε)-coreset of X for k-means if for all $C \subseteq \mathbb{R}^d$ such that $|C| = k$,

$$(1-\varepsilon)\text{cost}(X, C) \leq \text{cost}(S, C, w) \leq (1+\varepsilon)\text{cost}(X, C)$$

$\text{cost}(X, C) = \sum_{x \in X} d(x, C)^2$, $\text{cost}(S, C, w) = \sum_{x \in S} w(x) d(x, C)^2$.

Basics of Coresets

- Strong coreset: If above inequality is true for all queries $Q \in \mathcal{Q}$.
- Weak coreset: If above inequality is true for optimal solution $Q^* \in \mathcal{Q}$.

Basics of Coresets

Obtain Approximate Solutions using Coresets

- Construct coreset and solve problem on the coreset.
- Exact or approximate solution on the coreset gives approximate solution for dataset.
- We show: $\text{cost}(X, Q_S^*) \leq (1+2\varepsilon)\text{cost}(X, Q_X^*)$.
- $\text{cost}(X, Q_S^*) \leq \frac{1}{1-\varepsilon}\text{cost}(S, Q_S^*) \leq \frac{1}{1-\varepsilon}\text{cost}(S, Q_X^*) \leq \frac{1+\varepsilon}{1-\varepsilon}\text{cost}(X, Q_X^*) \leq (1+2\varepsilon)\text{cost}(X, Q_X^*)$.

Applications of Coresets

Properties of Coresets

Union of Coresets is a Coreset

- Let S_1, S_2 be (k, ε)-coresets for disjoint sets X_1 and X_2, then $S_1 \cup S_2$ is a (k, ε)-coreset for $X_1 \cup X_2$.

Composable Coresets

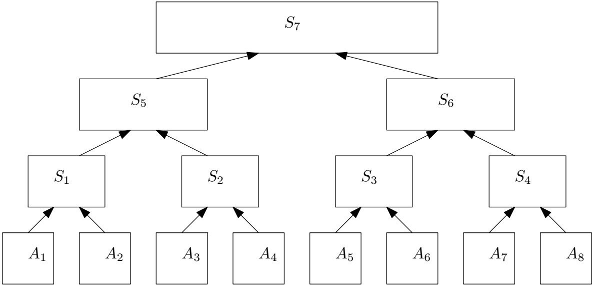
- If S_1 is a (k, ε)-coreset for S_2, and S_2 is a (k, δ)-coreset for S_3, then S_1 is $(k, \varepsilon + \delta + \varepsilon\delta)$-coreset for S_3.
- $\forall C$, $(1 - \varepsilon)\text{cost}(S_2, C, w_2) \leq \text{cost}(S_1, C, w_1) \leq (1 + \varepsilon)\text{cost}(S_2, C, w_2)$.
- $\forall C$, $(1 - \delta)\text{cost}(S_3, C, w_3) \leq \text{cost}(S_2, C, w_2) \leq (1 + \delta)\text{cost}(S_3, C, w_3)$.

Informally

- If S_1 is coreset of S_2 with $(1 + \varepsilon)$-guarantee, and S_2 is coreset of S_3 with $(1 + \delta)$-guarantee, then S_1 gives $(1 + \varepsilon)(1 + \delta)$-guarantee for S_3.

Merge and Reduce

- Design streaming algorithm on insertion only data streams [BS1980, HPM2004].



Merge and Reduce

Storage

- $\log n$ levels in the tree, each level has at most one coreset: $|S| \log n$.

Error of Approximation

- We compute coresets of coresets, the error of approximation goes up.
- Composing (k, ε) and (k, δ)-coresets gives guanrantee $(1 + \varepsilon)(1 + \delta)$.
- Guarantee using $\log n$ levels becomes $(1 + \varepsilon)^{\log n}$.
- We set $\varepsilon' = \frac{\varepsilon}{\log n}$.

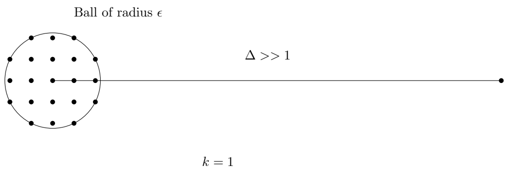
Distributed Algorithms using Coresets

- Data partitioned across machines, they compute coreset on local data.
- Machines send coresets to the central server.
- Server computes union of coresets, coresets of coresets.
- Complexity: Communication from machine to server: Coreset size.

Techniques for Coreset Constructions

Coresets using Uniform Sampling

- Idea: Subset of points sampled uniformly gives a coreset.
- Question: How many samples do we need? Size of coreset using uniform sampling?
- $\Omega(n)$ uniform samples.



Har-Peled and Majumder (HPM2004)

- Coresets for k-means and k-median in low dimensions.
- Computes coresets of size $O(k\varepsilon^{-d}\log n)$.

- Let C be constant factor approximation for k-means or k-median.
- Build exponential grid of $O(\log n)$ levels around each center.
- Snap input points to the closest point in the grid.
- Price of snapping smaller than εOPT.
- The weighted set S is a coreset.

Har-Peled and Kushal (2005)

- Computes coreset of size independent of n of size $O(\frac{k^2}{\varepsilon^d})$ for k-median and $O(\frac{k^3}{\varepsilon^{d+1}})$ for k-means.

- Let C be a constant factor approximation.
- Draw $O(\frac{1}{\varepsilon^{d-1}})$ lines from each center.
- Project each input point to the closest line.
- Coreset size of $O(\frac{k}{\varepsilon})$ and $O(\frac{k^2}{\varepsilon^2})$ for points on 1-D for k-median and k-means respectively.

Chen's Construction (2009)

- Coreset size for k-median and k-means $O(dk^2 \log n\varepsilon^{-2})$.

- Key idea: Partition dataset into disjoint subsets and take random samples from each subset.
- Start with an (α, β)-bicriteria approximation for k-means.
- Partition space using concentric rings around these centers.
- Take random samples from each ring.
- Coreset size for k-median and k-means $O(dk^2 \log n\varepsilon^{-2})$.

Feldman-Langberg (2011)

- Coreset size for k-means $\tilde{O}(k^3\varepsilon^{-4})$.

- Samples points based on how important the points are with respect to the objective function.
- First computes sensitivity of points, and then samples points with probability proportional to sensivity.

Coreset Constructions

Coresets for k-means

Reference	Coreset Size
Har-Peled & Majumdar	$O(k\varepsilon^{-d} \log n)$
Har-Peled & Kushal	$O(k^3\varepsilon^{-(d+1)})$
Chen	$\tilde{O}(dk^2\varepsilon^{-2} \log n)$
Feldman & Langberg	$\tilde{O}(dk\varepsilon^{-4})$
Feldman-Schmidt-Sohler	$\tilde{O}(k^3\varepsilon^{-4})$

Coreset Constructions using Dimensionality Reduction

Coresets for k-means/k-median

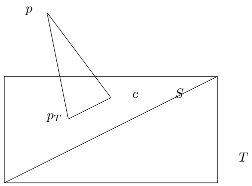
- Can you design coresets whose size is indepedent of d and n?
- Coreset size is polynomial in k and $\frac{1}{\varepsilon}$.

Coresets for k-means (FSS2013)

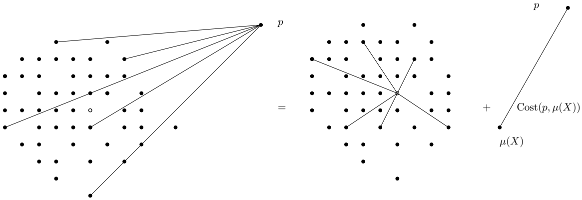
- Assume that the data is very high dimensional.
- They give a dimensionality reduction scheme to show that most of data lies in a much smaller dimensional subspace.
- Apply known coreset constructions on data in smaller dimensional subspace.

Coresets for *k*-means (FSS2013)

- Key idea: Cost of clustering of high dimensional points has a pseudo-random part and a structured part.
- Pseudo-random part of cost is same for all queries (with k centers).
- Structured part of the cost comes from clustering projected points.



Coreset for 1-means



- Identity for k-means: $\text{cost}(X, p) = \text{cost}(X, \mu(X)) + |X|||p - \mu(X)||^2$.
- Coreset centroid $\mu(X)$ with weight $|X|$ and constant $\text{cost}(X, \mu(X))$.

Coresets for k-means (FSS 2013)

Coreset Definition

- Let A be a set of n points in $\mathbb{R}^d$. A weighted set $S \in \mathbb{R}^{m \times d}$ and a constant $\Delta > 0$ is an ε-coreset for k-means if for all C

$$(1 - \varepsilon) cost(A, C) \leq cost(S, C) + \Delta \leq (1 + \varepsilon) cost(A, C)$$

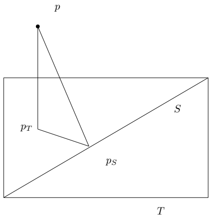
Coreset Construction for k-means (FSS2013)

Dimensionality Reduction Algorithm

- Let OPT is known for k-means.
- Compute k-dim subspace S that minimizes the sum of squared distances from points to the subspace.
- While there exists k dimensions such that adding those to S reduces the subspace approximation cost by at least ε^2OPT, add them to subspace S.
- Dimension of S is at most $\frac{k}{\varepsilon^2}$.
- Coreset for k-means: Projected points on S (Structured part) and cost of projection onto S (Pseudo-random part).

Analysis

- Let T be the subspace containing S and C (query with k centers).
- $\text{cost}(X, C) = \text{cost}(X, T) + \text{cost}(X_T, C) \approx \text{cost}(X, S) + \text{cost}(X_S, C)$.
- We have $\text{cost}(X, S) - \text{cost}(X, T) \leq \varepsilon^2 \text{OPT}$.
- On avarage projected points on T and S are close. Because, $\text{cost}(X_T, X_S) = \text{cost}(X, S) - \text{cost}(X, T) \leq \varepsilon^2 \text{OPT}$.
- Show that $|\text{cost}(X_S, C) - \text{cost}(X_T, C)| \leq \varepsilon \text{OPT}$.



Coresets for k-means

FSS13

- Let A be a set of n points in $\mathbb{R}^d$, equivalently, $A \in \mathbb{R}^{n \times d}$. Let A_m be its rank m-approximation for $m = O(\frac{k}{\varepsilon^2})$. Then, there exists a constant $\Delta = ||A - A_m||_F^2$ such that for all sets of k centers C,

$$(1-\varepsilon)\text{cost}(A, C) \leq \text{cost}(A_m, C) + \Delta \leq (1+\varepsilon)\text{cost}(A, C)$$

Coreset

- We have n points on $O(\frac{k}{\varepsilon^2})$-dimensional subspace S, and a constant equals the projection cost on subspace S.
- We apply Feldman-Langberg coreset construction on S to obtain a coreset of size $\tilde{O}(\frac{k^2}{\varepsilon^6})$.

Coresets for k-median Problem

Euclidean k-median Problem

- Given a set X of n points in $\mathbb{R}^d$, and an integer k, the objective is to find a set $C \subseteq \mathbb{R}^d$ of k centers such that the objective function

$$\sum_{x \in X} \min_{c \in C} ||x - c||_2$$

is minimized.

- k-median is NP-hard, and constant factor approximation algorithms are known for k-median.

Coresets for k-median

- Many results on designing strong coresets for k-median.
- Feldman-Langberg framework for k-median has coreset of size $\frac{kd}{\varepsilon^2}$.

Focus for this talk

- Woodruff-Sohler designs a coreset for k-median of size $\text{poly}(k, \frac{1}{\varepsilon})$, independent of d.

Coreset for k-median (Woodruff-Sohler'18)

- Can we get a coreset for k-median similar to k-means?
- Let X_S be the set of projected points on subspace S and a constant Δ. Do we have for all queries C,

$$(1-\varepsilon)\text{cost}(X, C) \leq \text{cost}(X_S, C) + \Delta \leq (1+\varepsilon)\text{cost}(X, C)$$

- Gave a counterexample to any such guarantee for k-median.

Coreset for k-median (Woodruff-Sohler'18)

Counterexample for $k = 1$

- Let there be n points on a unit ball in $\mathbb{R}^d$ for very high d.
- We project these points on a $l = poly(k, \frac{1}{\varepsilon})$-dimensional subspace.
- With high probability, norms of the projected points are very small.
- For query with center at origin, we require $\Delta = n$.
- For query with center at $\{1, 0, \cdots, 0\}$, we get cost of original points as $\sqrt{2}n$ and total cost of coreset and constant is $2n$.

Coreset for k-median (Woodruff-Sohler'18)

- Unlike for k-means, we cannot apply Pythagorean theorem here to split the cost among the cost of projection and cost of clustering of projected points.

Coreset for k-median (Woodruff-Sohler'18)

- Show that a variant of dimensionality reduction scheme works for k-median.
- Dimensionality reduction gives a set n points in $\mathbb{R}^{d+1}$ such that most of the points live in a much smaller dimensional subspace.

Coreset for k-median (Woodruff-Sohler'18)

- Key idea: Add a special dimension to any point with value equal to the distance to subspace S.

Dimensionality Reduction

Dimensionality Reduction Algorithm

- Let Opt be the cost of the optimal k-median clustering.
- Compute optimal k-dimensional subspace S for minimizing sum of distances from points to subspace S.
- While we can add k dimensions to S to reduce the cost of the subspace approximation problem by ε^2OPT, do that.
- Let S be the best such subspace.
- For each point p in X,
  1. Compute distance $d(p, p_S)$ where p_S is the projection on subspace S.
  2. Return $(p_S, d(p, p_S)) \in \mathbb{R}^{d+1}$

Analysis

- Let T denote the subspace containing both S and C.
- For any center $c_p \in C$, we have $d(p, c_p) = (d(p, p_T)^2 + d(p_T, c_p)^2)^{1/2}$.
- Cost with respect to the coreset is $d((p_S, d(p, p_S)), (c_p, 0)) = (d(p_S, c_p)^2 + d(p, p_S)^2)^{1/2}$.
- (Distance to Subspace Lemma) $\text{cost}(X, S) - \text{cost}(X, T) = \sum_p (d(p, p_S) - d(p, p_T)) \leq \varepsilon^2 \text{OPT}$.
- (Distance inside Subspace Lemma) $\sum_{p \in X} |d(p_T, c_p) - d(p_S, c_p)| \leq \varepsilon \text{OPT}$.

Distance inside Subspace Lemma

- To show: $\sum_{p \in P} |d(p_T, c_p) - d(p_S, c_p)| \leq \varepsilon \text{OPT}$.

- Using triangle inequality, this is at most $\text{cost}(X_S, X_T)$.
- For all $p \in Q$ such that $d(p_T, p_S) \leq \varepsilon d(p, p_S)$, we have $\sum_{p \in Q} d(p_T, p_S) \leq \varepsilon OPT$.
- Else, $d(p_T, p_S) = (d(p, p_S)^2 - d(p, p_T)^2)^{1/2}$.
- Since $d(p_T, p_S) > \varepsilon d(p, p_S)$, using triangle inequality, we have above expression is at most $\frac{d(p,p_S) - d(p,p_T)}{\varepsilon}$.
- Since $\sum_p d(p, p_S) - d(p, p_T) \leq \varepsilon^2 \text{OPT}$, we are done.

Analysis contd.

- $|\text{cost}(S, C) - \text{cost}(X, C)| \leq \varepsilon \text{cost}(X, C)$.
- We show: $\sum_p |d(p, c_p) - d((p_S, d(p, p_S)), (c_p, 0))| \leq 2\varepsilon \text{OPT}$.

$$
\begin{aligned}
&|d(p, c_p) - d((p_S, d(p, p_S)), (c_p, 0))| \\
&= |(d(p, p_T)^2 + d(p_T, c_p)^2)^{1/2} - (d(p, p_S)^2 + d(p_S, c_p)^2)^{1/2}| \\
&= |d(p, p_T), d(p_T, c_p)|_2 - |d(p, p_S), d(p_S, c_p)|_2 \\
&\leq |d(p, p_T) - d(p, p_S), d(p_T, c_p) - d(p_S, c_p)|_2 \\
&\leq |d(p, p_T) - d(p, p_S), d(p_T, c_p) - d(p_S, c_p)|_1 \\
&= |d(p, p_T) - d(p, p_S)| + |d(p_T, c_p) - d(p_S, c_p)| \\
&\leq 2\varepsilon \text{OPT}
\end{aligned}
$$

using Distance to Subspace Lemma and Distance inside Subspace Lemma

Thanks & Questions

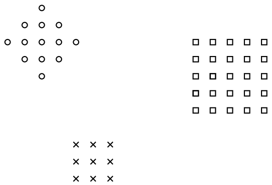
Sampling-based Algorithms for Clustering Problems

Anup Bhattacharya
Indian Statistical Institute, Kolkata

k-means Clustering Problem

k-means Clustering Problem

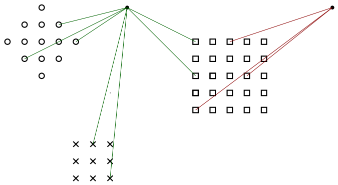
- Input: Dataset $X \subseteq \mathbb{R}^d$, and integer k.
- Cost function: For $C \subseteq \mathbb{R}^d, |C| = k$, $\Phi(X, C) = \sum_{x \in X} \min_{c \in C} ||x - c||^2$.



k-means Clustering Problem

- Input: Dataset $X \subseteq \mathbb{R}^d$, and integer k.
- Cost function: For $C \subseteq \mathbb{R}^d, |C| = k$, $\Phi(X, C) = \sum_{x \in X} \min_{c \in C} ||x - c||^2$.

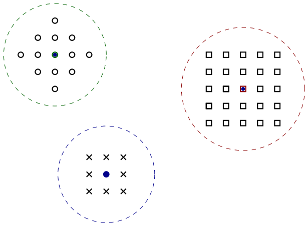
- Objective: Find set $C \subseteq \mathbb{R}^d$ of k centers that minimizes $\Phi(X, C)$.



k-means Clustering Problem

- Input: Dataset $X \subseteq \mathbb{R}^d$, and integer k.
- Cost function: For $C \subseteq \mathbb{R}^d, |C| = k$, $\Phi(X, C) = \sum_{x \in X} \min_{c \in C} ||x - c||^2$.

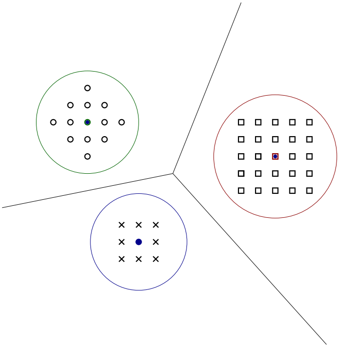
- Objective: Find set $C \subseteq \mathbb{R}^d$ of k centers that minimizes $\Phi(X, C)$.



k-means Clustering Problem

- Input: Dataset $X \subseteq \mathbb{R}^d$, and integer k.
- Cost function: For $C \subseteq \mathbb{R}^d, |C| = k$, $\Phi(X, C) = \sum_{x \in X} \min_{c \in C} ||x - c||^2$.
- Objective: Find set $C \subseteq \mathbb{R}^d$ of k centers that minimizes $\Phi(X, C)$.

- Voronoi partitioning gives k clusters.



Known Results: k-means Clustering

- α-approximation ALG: for any instance I, $\text{ALG}(I) \leq \alpha \cdot \text{OPT}(I)$.

Hardness Results	Approximation Algorithms
NP-hard for $k \geq 2$ [D2008]	6.357 by Ahmadian *et al.* (2016)
NP-hard for $d \geq 2$ [V2009,MNV2012]	$(1+\varepsilon)$ in $O(nd2^{\tilde{O}(\frac{k}{\varepsilon})})$ [JKS2014]
APX-hard [Awasthi *et al.* (2015)]	

Approximation Algorithm for k-means

1-means Problem

- Objective function: $\min_{c \in \mathbb{R}^d} \Phi(X, \{c\}) = \min_{c \in \mathbb{R}^d} \sum_{x \in X} ||x - c||^2$.

Exact Solution

- Centroid of points is the optimal center for 1-means.

Approximate Solution

- A uniformly sampled point gives 2-approximation in expectation.
- Fact: $\Phi(X, p) = \Phi(X, \mu(X)) + |X| \cdot ||p - \mu(X)||^2$
- Centroid of $O(\frac{1}{\varepsilon})$ points sampled uniformly at random gives $(1+\varepsilon)$-approximation for 1-means with constant probability [IKI1994].

2-means Problem

- 2-means is NP-hard.

Approximate Solution

- Require a sample of size $O(\frac{1}{\varepsilon})$ chosen uniformly at random from each of the optimal clusters.

2-means Problem

Approximate Solution

- Require a sample of size $O(\frac{1}{\varepsilon})$ chosen uniformly at random from each of the optimal clusters.

Approximate Larger Optimal Cluster

- Uniformly sample $\frac{2}{\varepsilon}$ points. Sample contains at least $\frac{1}{\varepsilon}$ points from the larger optimal cluster.
- Consider all subsets of size $\frac{1}{\varepsilon}$ of the sample. Running time $\binom{\frac{2}{\varepsilon}}{\frac{1}{\varepsilon}}$.
- Centroid of these subsets are candidate centers for the optimal center of the larger cluster.

Approximate Smaller Optimal Cluster

- How do you approximate the center for the smaller optimal cluster?

2-means Problem

Approximate Smaller Optimal Cluster

- How do you approximate the center for the smaller optimal cluster?

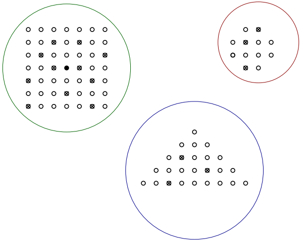
Prune and Sample

- For each of the candidate centers of the larger optimal cluster, consider the set Q of farthest $\frac{n}{2^{i-1}}$ points from the candidate center for $1 \leq i \leq \log n$.
- Randomly sample $O(\frac{1}{\varepsilon^2})$ points from Q. Consider all possible subsets of size $O(\frac{1}{\varepsilon})$ from the sample.
- Centroid of at least one subset gives $(1+\varepsilon)$-approximation for the smaller optimal cluster.
- Same idea works for any $k \geq 2$ [KSS2010].

Sampling based $(1+\varepsilon)$-approximations for k-means

Approximate Largest Optimal Cluster

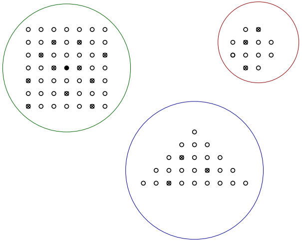
- Step 1: Uniformly sample $O(\frac{k}{\varepsilon})$ points.
- Whp, sample contains $O(\frac{1}{\varepsilon})$ points from largest optimal cluster.
- Step 2: Consider means of subsets of size $O(\frac{1}{\varepsilon})$ of sample.
- Approximates cluster in time $O(\frac{k}{\varepsilon})^{O(\frac{1}{\varepsilon})}$.



Sampling based $(1+\varepsilon)$-approximations for k-means

Approximate Smaller Optimal Clusters

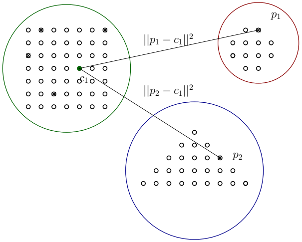
- Number of points in some optimal clusters may be very small.
- Uniform sampling does not help to approximate smaller clusters.



D^2-Sampling

D^2-Sampling

- Let C be set of already chosen centers.
- D^2-sampling chooses point p as next center wp prop. to $\min_{c \in C} ||p - c||^2$.



D^2-Sampling based Algorithms

- k centers using D^2-sampling gives $O(\log k)$-approximation [AV2007].
- $O(k)$ such centers give constant pseudo-approximation [ADK2009].

D^2-Sampling based Algorithms

- k centers using D^2-sampling gives $O(\log k)$-approximation [AV2007].
- $O(k)$ such centers give constant pseudo-approximation [ADK2009].

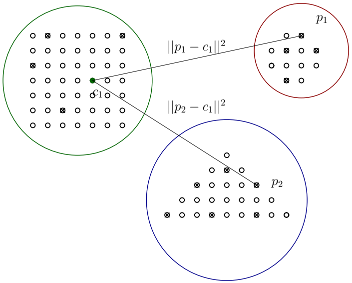
k-means++

- A point is sampled from an *uncovered* optimal cluster, that cluster is well-approximated.
- Overall $(\log k)$-approximation because may miss some clusters.
- Lower bound of $\Omega(\log k)$ for k-means++.

Sampling based $(1+\varepsilon)$-approximations for k-means

D^2-Sampling based Algorithm

- Iterative algorithm, C_i be chosen centers till ith iteration.
- Step 1: S is D^2-sample with respect to C_i of $O(\frac{k}{\varepsilon^3})$ points.
- Step 2: Consider mean of subsets of size $O(\frac{1}{\varepsilon})$ of sample.



- $(1+\varepsilon)$-approx for k-means in time $O(nd \cdot 2^{\tilde{O}(\frac{k}{\varepsilon})})$ [JKS2014].

Constrained Clustering: Examples

- Given n points in $\mathbb{R}^d$, and integer k.
- Objective function: $\sum_{x \in X} \min_{c \in C} ||x - c||^2$
- Minimize objective while obeying additional constraints.
- Examples of constraints:
  - r-gather clustering: Each cluster has size at least r.
  - Capacitated clustering: Cluster sizes have upper bounds.
  - Chromatic clustering: No two points in cluster with same color.

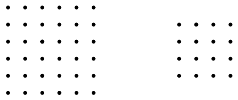


Figure : r-gather clustering: Input points in $\mathbb{R}^2$, $k = 2$, $r = 20$

Constrained Clustering: Examples

- *r*-gather clustering: Each cluster has size at least *r*.
- Unconstrained *k*-means clustering on the input instance.

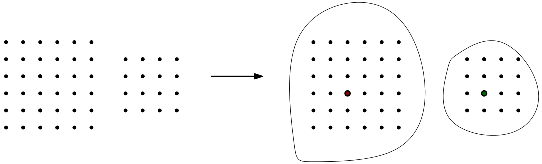


Figure : Solution for Unconstrained clustering

Constrained Clustering: Examples

- r-gather clustering: Each cluster has size at least r.

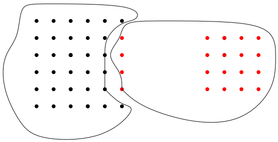


Figure : r-gather clustering: Input points in $\mathbb{R}^2$, $k = 2$, $r = 20$

Constrained *k*-means Problem

- Constrained k-means [Ding & Xu 2015]: Given n points in $\mathbb{R}^d$, integer k, and set of constraints, find k clusters which minimize objective function.
- $(1+\epsilon)$-approximation for constrained k-means [Ding & Xu 2015].

Constrained *k*-means Problem

- Locality property: Points in the same cluster are closer to each other.
- True for unconstrained clustering.
- Locality not valid for constrained clustering.

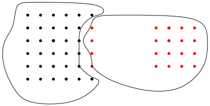


Figure : r-gather Clustering: Input points in $\mathbb{R}^2$, $k = 2$, $r = 20$

Cluster Assignment: Find Clusters from Centers

- Find clusters given k centers.
- Voronoi partitioning works for unconstrained clustering.
- Constrained clustering: [Ding & Xu 2015] Designed polynomial time assignment algorithms for various constrained k-means problems.

Cluster Assignment Algorithm

- Find clusters minimizing objective while satisfying constraints.
- Assignment algorithm for r-gather clustering [Ding & Xu 2015]
- Reduces to min-cost circulation problem.

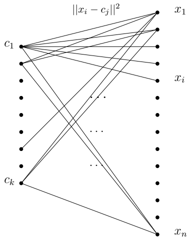


Figure : Assignment algorithm for r-gather Clustering

Constrained k-means: Known Results

- Number of candidate centers $L \leq O((\log n)^k 2^{\text{poly}(\frac{k}{\epsilon})})$.
- Assignment takes $P(X)$ time to find clustering cost.
- Ding & Xu give $(1+\epsilon)$-approximation in time $O(nd \cdot L + P(X) \cdot L)$.

List k-means Problem

- Given $X \subseteq \mathbb{R}^d$, integer k, $\epsilon > 0$, implicit OPT partition $X_1, \ldots, X_k$.
- List k-means finds a set $C = \{C_1, \ldots, C_L\}$.
- Each C_i is set of k centers.
- Such that $\exists j \in [1, L]$, C_j gives $(1 + \epsilon)$-approximation wrt $X_1, \ldots, X_k$.

List k-means to Constrained k-means

- List k-means outputs a list of candidate k-centers.
- For each k-center, compute clustering using assignment algorithm.
- The clustering with minimum cost would be the solution for constrained k-means.

List k-means

- List size in [Ding & Xu] is $L \leq O((\log n)^k 2^{\text{poly}(\frac{k}{\epsilon})})$
- [BJK2018] has list size $L \leq 2^{\tilde{O}(\frac{k}{\epsilon})}$
- Notice that list size is independent of n.
- Almost matching lower bound: $L \geq 2^{\tilde{\Omega}(\frac{k}{\sqrt{\epsilon}})}$
- Running time: $O(nd \cdot L + P(X) \cdot L)$

List k-means

- List size in [Ding & Xu] is $L \leq O((\log n)^k 2^{\text{poly}(\frac{k}{\epsilon})})$
- [BJK2018] has list size $L \leq 2^{\tilde{O}(\frac{k}{\epsilon})}$
- Notice that list size is independent of n.
- Almost matching lower bound: $L \geq 2^{\tilde{\Omega}(\frac{k}{\sqrt{\epsilon}})}$
- Running time: $O(nd \cdot L + P(X) \cdot L)$

List k-means

- List size in [Ding & Xu] is $L \leq O((\log n)^k 2^{\text{poly}(\frac{k}{\epsilon})})$
- [BJK2018] has list size $L \leq 2^{\tilde{O}(\frac{k}{\epsilon})}$
- Notice that list size is independent of n.
- Almost matching lower bound: $L \geq 2^{\tilde{\Omega}(\frac{k}{\sqrt{\epsilon}})}$
- Running time: $O(nd \cdot L + P(X) \cdot L)$

List k-means

- List size in [Ding & Xu] is $L \leq O((\log n)^k 2^{\text{poly}(\frac{k}{\epsilon})})$
- [BJK2018] has list size $L \leq 2^{\tilde{O}(\frac{k}{\epsilon})}$
- Notice that list size is independent of n.
- Almost matching lower bound: $L \geq 2^{\tilde{\Omega}(\frac{k}{\sqrt{\epsilon}})}$
- Running time: $O(nd \cdot L + P(X) \cdot L)$

List *k*-means

- List size in [Ding & Xu] is $L \leq O((\log n)^k 2^{\text{poly}(\frac{k}{\epsilon})})$
- [BJK2018] has list size $L \leq 2^{\tilde{O}(\frac{k}{\epsilon})}$
- Notice that list size is independent of n.
- Almost matching lower bound: $L \geq 2^{\tilde{\Omega}(\frac{k}{\sqrt{\epsilon}})}$
- Running time: $O(nd \cdot L + P(X) \cdot L)$

List k-means

- List size in [Ding & Xu] is $L \leq O((\log n)^k 2^{\text{poly}(\frac{k}{\epsilon})})$
- [BJK2018] has list size $L \leq 2^{\tilde{O}(\frac{k}{\epsilon})}$
- Notice that list size is independent of n.
- Almost matching lower bound: $L \geq 2^{\tilde{\Omega}(\frac{k}{\sqrt{\epsilon}})}$
- Running time: $O(nd \cdot L + P(X) \cdot L)$
- Can be extended for List k-median problem.

Constrained Clustering

- For the largest OPT cluster things are fine.
- D^2-sampling based scheme does not work for constrained clustering.

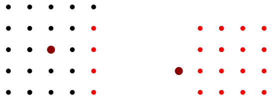


Figure : D^2-sampling points, $k = 2$

Constrained Clustering

- Centroid of none of the subsets may be good.

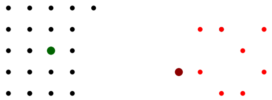


Figure : D^2-sampling points, $k = 2$

Idea: Constrained Clustering

- Cluster misses representation if portions of it close to covered clusters.
- Idea: Add $O(\frac{1}{\epsilon})$ copies of centers in C to the set of sampled points.
- Trying all subsets of this new set works.
- We obtain $(1+\epsilon)$-approximation for List k-means with $L = 2^{\tilde{O}(\frac{k}{\epsilon})}$.

Thanks & Questions