Clustering Perturbation Resilient Instances

Apoorv Vikram Singh

Dec 1, 2018

International Institute of Information Technology, Bangalore

Outline

Definition(?)

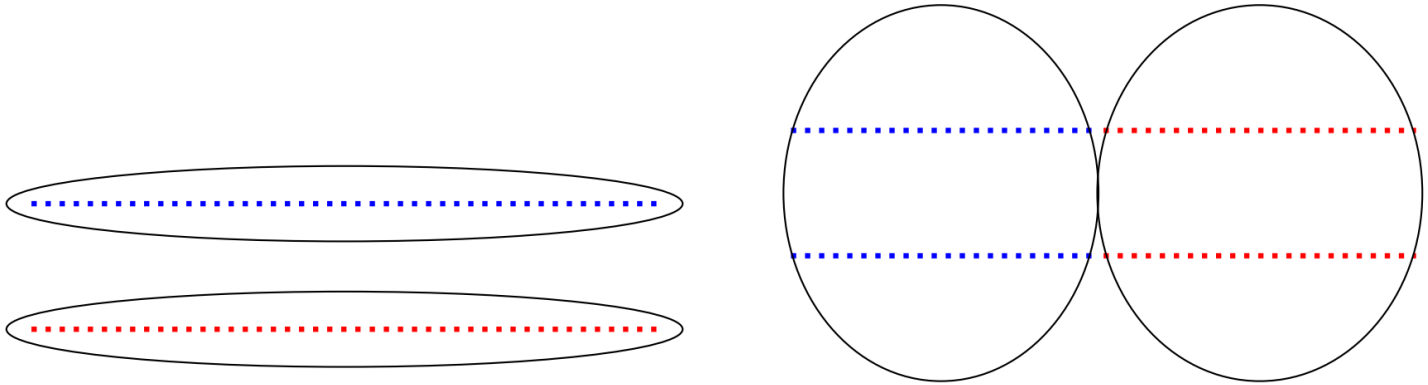
- Clustering is the task of grouping a set of objects such that similar objects end up in the same group and dissimilar objects end up in different groups.
- What do we mean by "similar", "dissimilar" objects?
- But before that, let's look at some applications.

Application

- Search engines try to group similar objects in one cluster and the dissimilar objects far from each other. It provides result for the searched data according to the nearest similar object which are clustered around the data to be searched.
- Exploratory data-analysis, almost across all disciplines.

Definition(?)

- Above description of similarity is vague, it is not at all clear how to define rigorously the notion of similarity.
- One of the problems as mentioned in is that the notion of similarity is not a transitive relation, while cluster sharing relation is an equivalence relation.



- Both the clusterings are equally justifiable. A given data-set may be clustered in various meaningful ways.

- One more problem is the lack of "ground-truth" for clustering. There is no absolute success evaluation procedure for clustering.
- One of the popular approaches is to define a cost function over a parameterized set of possible partition.
- the goal of the clustering algorithm is to find a partitioning that outputs a minimum/minimal cost clustering.
- Among the most popular ones, are the k-means, k-median, k-medoids, k-centers, etc.
- We will mostly focus on the k-means clustering algorithm.

Definition

- Group the data into k clusters $C = \{C_1, C_2, \ldots, C_k\}$ so as to minimize the squared sum of distances from elements to the centers of mass of their clusters $\mu = \{\mu_1, \mu_2, \ldots, \mu_k\}$.

$$\text{argmin}_{C_1,\ldots,C_k} \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 .$$

k-means is hard!!

- [Aloise et al., 09], [Dasgupta et al., 09], [Mahajan et al., 12] Optimization of the k-means objective is NP-hard in the worst case (even $k = 2$ or $d = 2$).
- [Awasthi et al., 15] There exists an $\epsilon > 0$ such that it is NP-hard to approximate the k-means objective to within a factor of $(1 + \epsilon)$.
- Doesn't really stop us from designing algorithms.

Lloyd's Algorithm

- Lloyd's Algorithm
  1. Start with k centers $\mu_1, \ldots, \mu_k$ arbitrarily.
  2. Assign every $x \in X$ to the cluster C_i whose cluster center c_i is closest to it, i.e., $\|x - \mu_i\| \leq \|x - \mu_j\|$ for all $j \neq i$.
  3. Set $\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$.
  4. If clusters or centers have changed go to step 2. Otherwise, terminate.
- Motivation: Parallel Axis Lemma

$$\sum_{i=1}^{n} \|a_i - x\|^2 = \sum_{i=1}^{n} \|a_i - \mu\|^2 + n \|\mu - x\|^2 .$$

Lloyd's can be bad!

- There is no provable bound Lloyd's algorithm achieves. The cost can be arbitrarily bad.
- There are also known worst-case instances where the popular Lloyd's algorithm takes exponentially many iterations to converge to a local optimum.
- However, Lloyd's work quite fast in real life!
- `[Arthur et al., 09]` Lloyds algorithm has a smoothed running time polynomial in n.

Lloyd's can be bad!

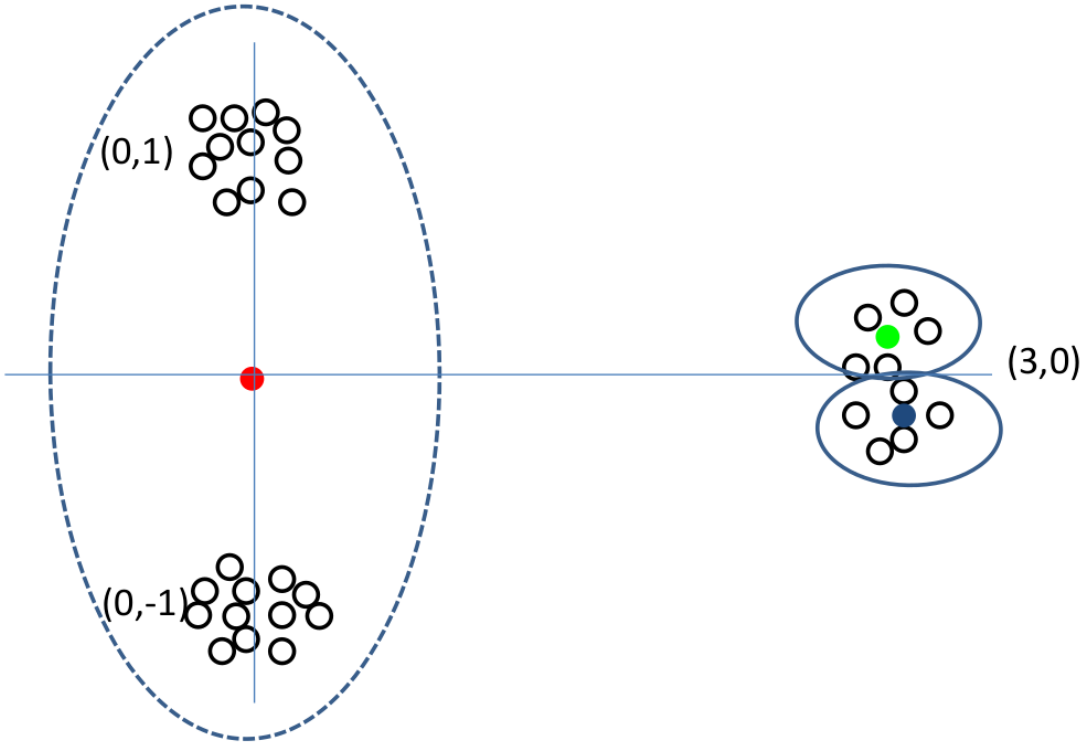


Figure: A locally-optimal but globally-suboptimal k-means clustering.

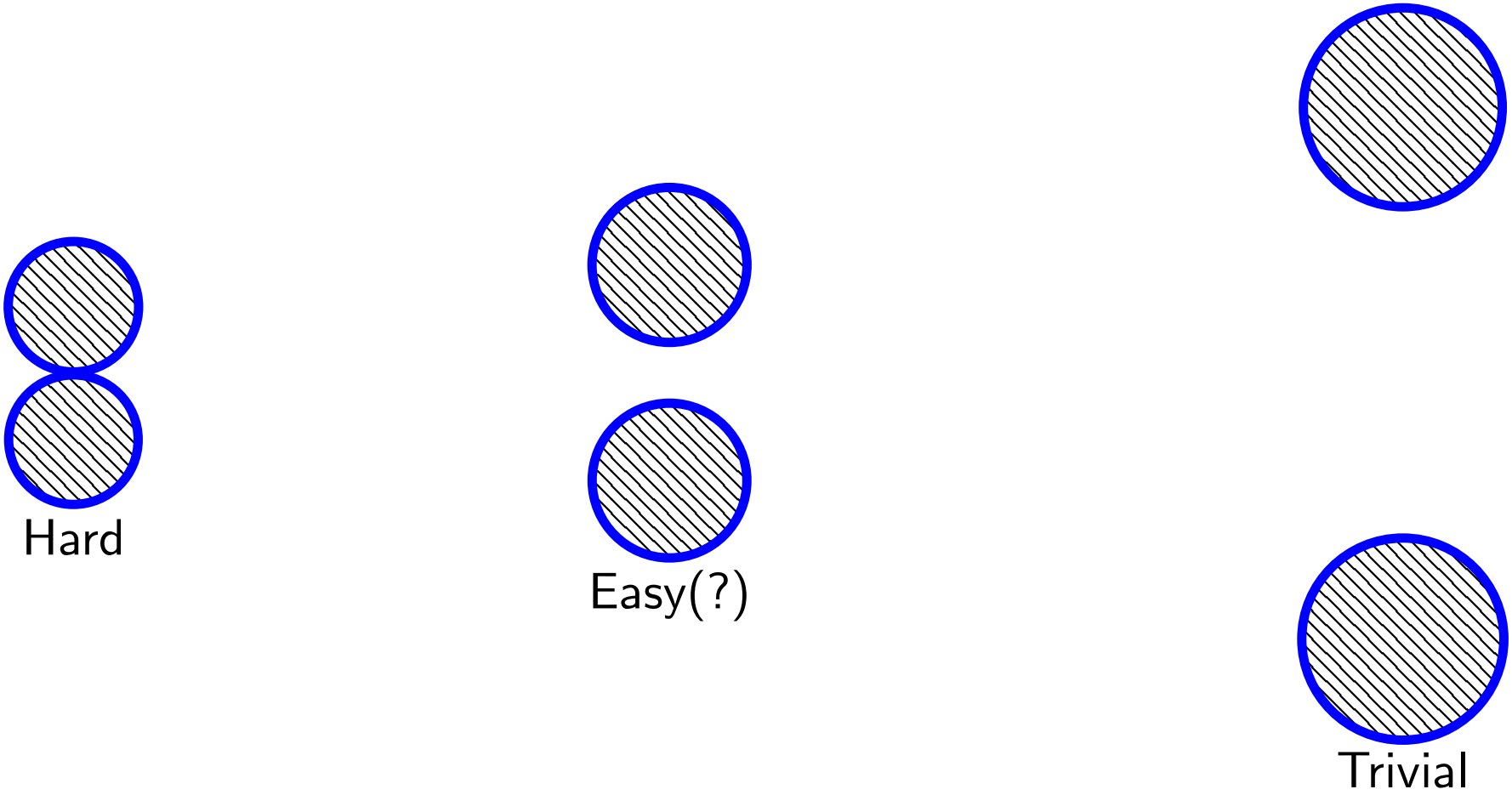
k-means known results

- [Inaba et al., 94] PTAS when both k, and d are constant.
- [Kumar et al., 04] PTAS only when k is assumed to be a constant.
- [Cohen-Addad et al., 16], [Friggstad et al., 16] PTAS only when d is assumed to be a constant.
- [Ahmadian et al., 17] $6.357 + \epsilon$-approximation algorithm for k-means.

- In practice, clustering algorithms like Lloyd's work very well on real world data sets.
- This dichotomy between the theoretical intractability and the empirical observations has lead to the CDNM thesis: Clustering is difficult only when it does not matter!

This has lead to a study of "Beyond Worst Case Analysis". The idea here is to assume something about the structure of the input itself, and then design provable algorithms accordingly.

Intuition



- How do we formalize the easiness?

Additive Perturbation Resilience

- Let the optimal clustering be $C = \{C_1, \ldots, C_k\}$.
- Move each point by a factor of ϵD in arbitrary direction, where D is the largest pairwise distance between optimal means.
- If even after *additive* perturbation, the clustering remains the same, then the instance is ϵ-additive perturbation resilient (ϵ-APS).
- Simple observation: Smallest distance between any two clusters is greater than $(2\ \epsilon D)$.... there's more...

Geometry of ϵ-APS

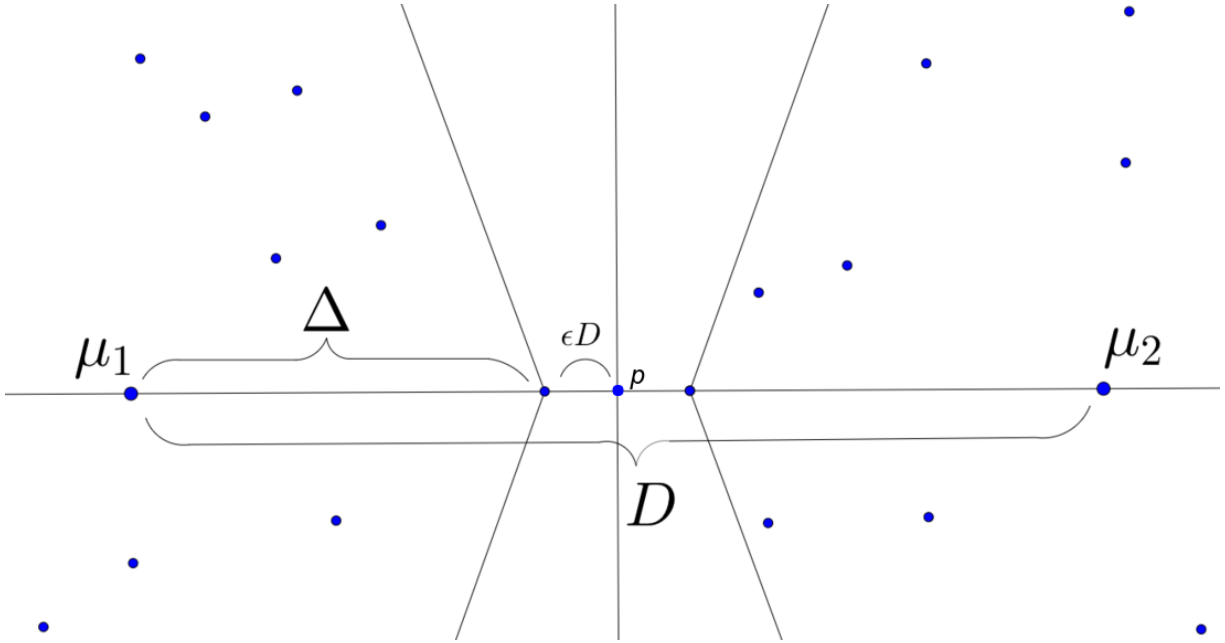


Figure: An ϵ-APS instance.

- The half angle of the cone is $\arctan(1/\epsilon)$.
- Can use "perceptron"-based algorithm, due to the geometry.

α-Perturbation Resilient

- Here instead of moving each point by an additive factor, we change the "edge weights" by a multiplicative factor of $\alpha \geq 1$.
- If even after this, the clustering remains the same, then we say that the instance is α-perturbation resilient.
- We will try to exploit the geometry of such instances...

α-Center Proximity

- A clustering is α-center proximal, if the ratio of the distance between a point and its cluster center, and the distance between that point to any other cluster center is less than or equal to $1/\alpha$.

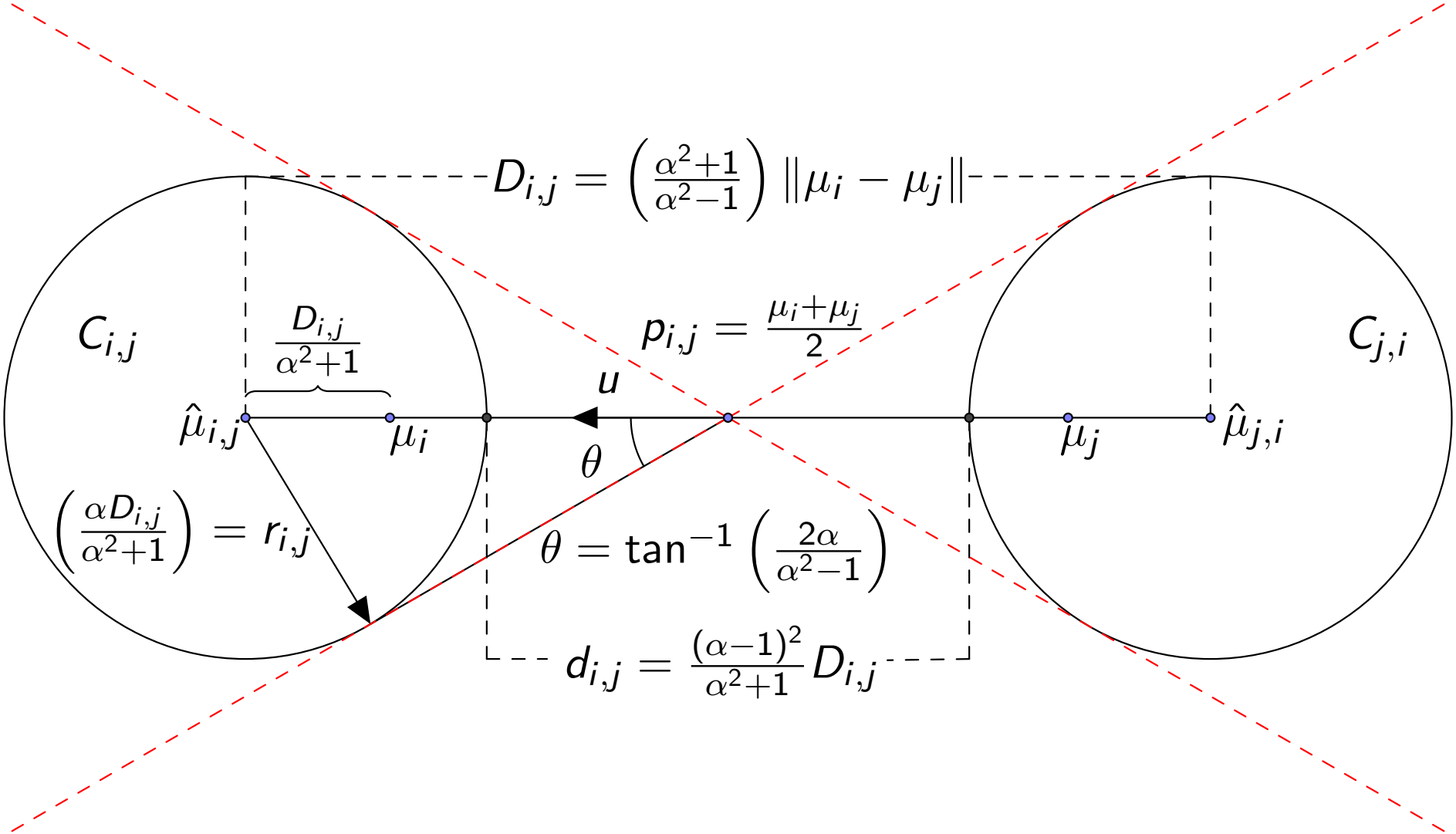
$$\forall i \neq j,\ x \in C_i,\ \ :\ \text{dist}(x, \mu_j) > \alpha\ \text{dist}(x, \mu_i)\,.$$

- Theorem: α-perturbation resilience implies α-center proximity.
- We will work with α-center proximal instances (a weaker condition than α-perturbation resilience).

Our Contribution

- [Angelidakis et al, 17] gave an algorithm which runs in time polynomial in n, d, and k for values of $\alpha \geq 2$ (best known).
- We give an algorithm (assuming k is a constant), linear in n, d, and exponential in k and $\frac{1}{\alpha-1}$. This works for **any** value of $\alpha > 1$.
- We return the exact clustering!! in $\mathcal{O}\left(nd2^{\text{poly}(\frac{k}{\alpha-1})}\right)$.
- Hard to approximate!
- Let us look at the geometry of α-center proximal instances.

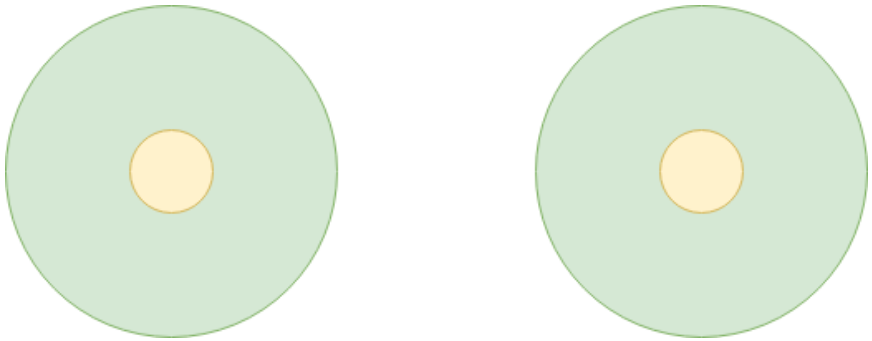
Geometry of center proximal instances



Main Theorem

- Desired Clustering: All the clusters are balanced, i.e., the size of the largest cluster to the smallest cluster is some constant.
- Our "similarity" assumption is the k-means similarity, and the "ground-truth" clustering is the α-center proximal instance.
- Theorem: Given a promise that the clustering desired (k-means similarity measure) is "balanced", and α-center proximal, then our algorithm will output the exact desired clustering with high probability in time $\mathcal{O}\left(nd2^{\text{poly}(\frac{k}{\alpha-1})}\right)$.
- One can iterate over various values of α and "balanced" ratio.

Main Idea



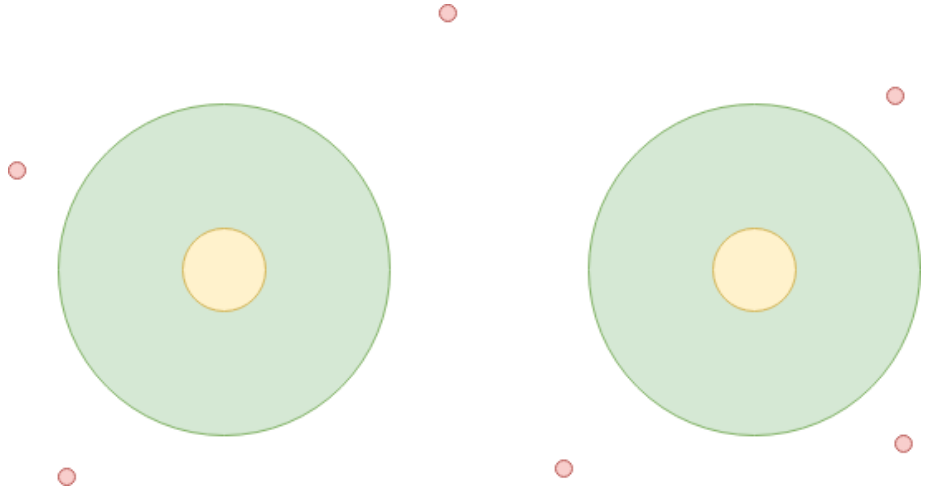
- One can get approximately close to the centers.
- Center proximity implies bounded radius of the clusters.
- Approximate Caratheodory Theorem: Sample points uniformly at random from a cluster (of bounded radius), and mean of those samples will get you close to the actual mean.

The Algorithm

1. From the given data, sample a constant (depends on α, k) number of points.
2. Find all k-partitions of the sample. Iterate over all the possible k-partitions:
   1. For each possible k-partition, find the mean (sampled means) of the k clusters.
   2. Assign points in the original data-set to the k-clusters on the basis of distance to the sampled means.
   3. Store this clustering.
3. For each of the stored clusterings obtained, verify if the clusters satisfy α-center proximity, and output the minimum cost one among them.

Clustering with Outliers

- For $x \in C_i$, $\text{dist}(x, \mu_j) > \alpha \text{ dist}(x, \mu_i)$.
- Moreover, $x \in C_i$ and $y \in Z$ we have $\text{dist}(y, \mu_j) > \alpha \text{ dist}(x, \mu_i)$.



- The same algorithm can be used because we are happy with the approximate means.

Lower Bounds

- The exponent in $\frac{1}{\alpha-1}$ is expected, since, when $\alpha = 1$, it corresponds to any general instance of k-means.
- For any $2 > \alpha' > 1$ there exists an $\alpha \leq \alpha'$, $(\alpha > 1)$, constants $\epsilon > 0$, and $\omega > 0$, such that it is NP-hard to approximate the optimal α-center proximal Euclidean k-means, where the size of each cluster is at least $\omega n/k$, to a factor better than $(1 + \epsilon)$.
  - Follows from hardness of approximation for Euclidean k-Means (Awasthi et al. '15)
- Number of clusterings can be huge: For any $2 > \alpha' > 1$, and any $k \in \mathbb{Z}_+$, there exists $\alpha \leq \alpha'$ (and $\alpha > 1$), n, d and a set of points $X \in \mathbb{R}^d$ such that such that the number of possible optimal α-center proximal clusterings is $2^{\tilde{\Omega}\left(k/\left(\alpha^2-1\right)\right)}$.

Comments on the Algorithm

The algorithm is closely related to the one given by [Kumar at al., 04].

- Can be extended to handle a certain class of outliers (points which are far away from optimal means).
- Can return a clustering of a desired cost as well (and not only minimum).

Speeding it up

- Under one more assumption of stability, i.e., the points which are "far away" from the optimal centers give a significantly worse cost compared to the original cost, we can speed up the algorithm.
- Heuristic: Use some state of the art clustering algorithm (like k-means-++), on the sampled data-set and use the clustering induced by them on the original data-set. (we loose all guarantees here :( )

Min-Max k-Means

- Given a clustering $C_1, \ldots C_k$, its min-max k-means cost is defined as

$$\max_{l \in [k]} \sum_{x_i \in C_l} \|x_i - \mu_l\|^2 .$$

- We want to minimize the cost of the heaviest cluster.
- Implies that each of the cluster has small cost.
- It is somewhere between k-means and k-center clustering.

- $\mathcal{O}(1)$ approximation for k-means implies $\mathcal{O}(k)$ approximation for min-max k-means.
- Related Work: Min-Max k-Median
  - [Ahmadian et al.]- approximation algorithms for line metrics and star metrics.
  - [Even et al.]- $(4, 4)$-bi-criteria approximation.
  - [Arkin et al.]- $(3, 3)$-bi-criteria approximation.
- Related to scheduling problems.

Estimating Means Suffices?

Let S be a set of points obtained by i.i.d. uniformly sampling M points from a point set $X \subset \mathbb{R}^d$. Then for any $\delta > 0$,

$$\mathbb{P}\left[\phi_{\mu(S)}(X) \leq \left(1 + \frac{1}{\delta M}\right) \cdot \Delta(X)\right] \geq (1 - \delta),$$

where Δ is the 1-means cost of X. For for $k = 1$ we are done. What about $k \geq 2$?.

Estimating Means Suffices?

Theorem: Given a set of centers, assigning points to those centers to minimize the min-max k-means cost is NP-hard to approximate to a factor better than $(3/2 - \epsilon)$.

- Reduce from minimum makespan scheduling to min-max k-means. Result follows from [Lenstra et al.]
  - Each machine mapped to $x = 0$.
  - Job j with processing time p_j is at a distance of $\sqrt{p_j}$ from the origin.

Estimating Means Suffices!

- Make use of the algorithms for minimum makespan scheduling.
  - k- centers corresponds to k- machines.
  - n- points corresponds to n jobs, with processing time $p_{ij} = \|x_i - \mu_j\|^2$.
  - Run the algorithm for minimum makespan scheduling on unrelated machines.
- Minimum makespan scheduling
  - [Lenstra et al.] 2-approximation algorithm in time $poly(n, k)$.
  - [Jansen & Mastrolilli] $(1+\epsilon)$-approximation in time $\tilde{\mathcal{O}}\left(2^{\tilde{\mathcal{O}}(k)} n\right)$.

Algorithm

- Sample $poly(k/\epsilon)$ points.
- Go over all k partitioning of these points and get a candidate set of centers.
- Run the minimum makespan scheduling algorithm by Jansen and Mastrolilli.
- Output the minimum cost clustering.

- Theorem: $(1+\epsilon)$-approximation in time $\mathcal{O}\left(2^{poly(k/\epsilon)}nd\right)$.
- Uniformly sampling might not work in case of small cluster. [Bhattacharya et al. '18] give an algorithm for this.

Bi-criteria Approximation

Motivated by the previous works, we know that estimating means suffices.

- What if we are allowed to sample a bit more than k points?
- Aim: to obtain a set of more than k centers such that all the points x_i are "close" to them.
- And then use the 2 approximation algorithm for scheduling problem.

How to achieve that aim?

D^2-Sampling: Sample a new point proportional to the squared distance from the current set of samples.

- Initialize $S = \emptyset$
- Pick x from X with probability proportional to $d(x, S)^2 = \min_{y \in S} \|x - y\|^2$.
- Add x to S and repeat.

◇ [Arthur et al.] k-means $++$ gives $\mathcal{O}(\log k)$ approximation to k-means.

- [Deshpande et al.] use k-means$++$ to get $(\mathcal{O}(1), \mathcal{O}(1))$ approximation to k-means.

Analysis

- Essentially use the same kind of analysis. Either the point in the current sample gives a good approximation to some centers or the new point comes from an uncovered cluster.
- Use scheduling algorithm.
- Sample $(1+\epsilon)k$ points and we get $\mathcal{O}\left(1/\epsilon^6\right)$ approximation to min-max k-means.

Open Problems

- Is there some other algorithm with running time polynomial in k as well for values of α a bit away from 1?
- Since the Lloyd's algorithm works so well in real life, can we use an algorithm similar in flavour to the Lloyd's algorithm.
- Other notions of stability, which are can be tested in sub-linear time.
- Suitable notions of perturbation resilience for other clustering objectives.
- Constant factor approximation algorithm for min-max k-means?

Thank You.
Questions?