A Beyond-Worst-Case Analysis of Greedy k-means++

Qingyun Chen Sungjin Im

UC Santa Cruz

Ryan Milstrey

UC Merced

Benjamin Moseley

CMU

Chenyang Xu

East China Normal University

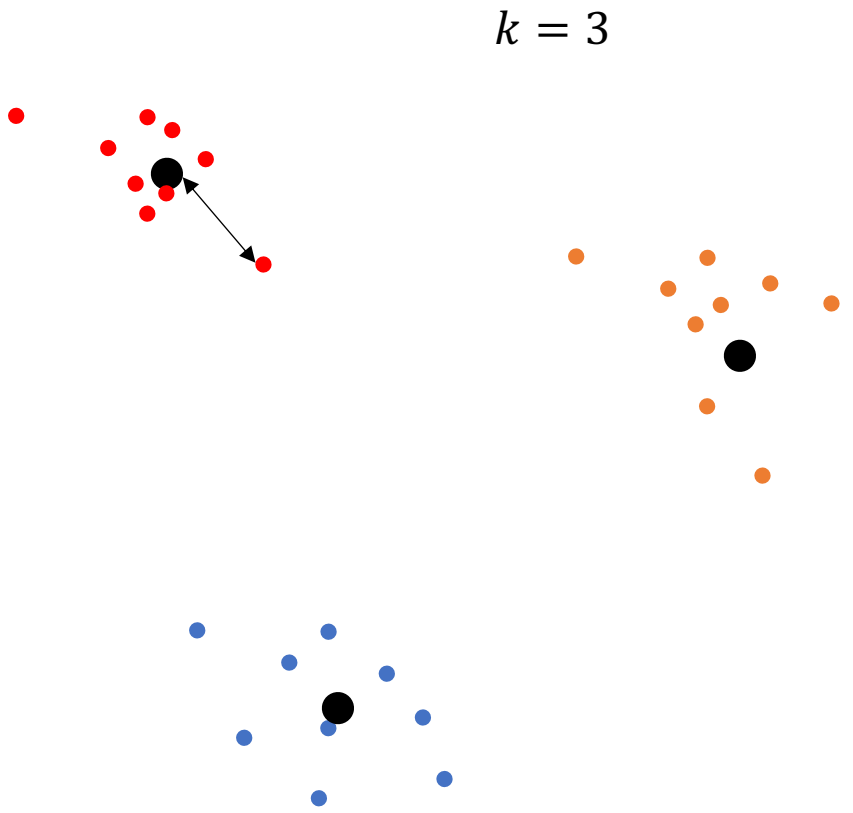
Ruilong Zhang

Technical University of Munich

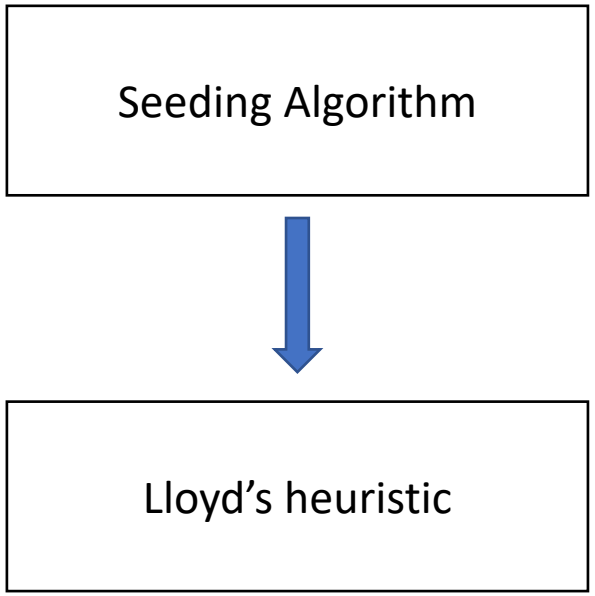
k-means Clustering

Input: n points $X \subseteq \mathbb{R}^d$, and a parameter k.

Goal: find a set $C \subseteq \mathbb{R}^d$ of k centers that minimize the sum of the squared distances between each point and its closest center, i.e., $\sum_{x \in X} \min_{c \in C} ||x - c||_2^2$.



Lloyd's heuristic



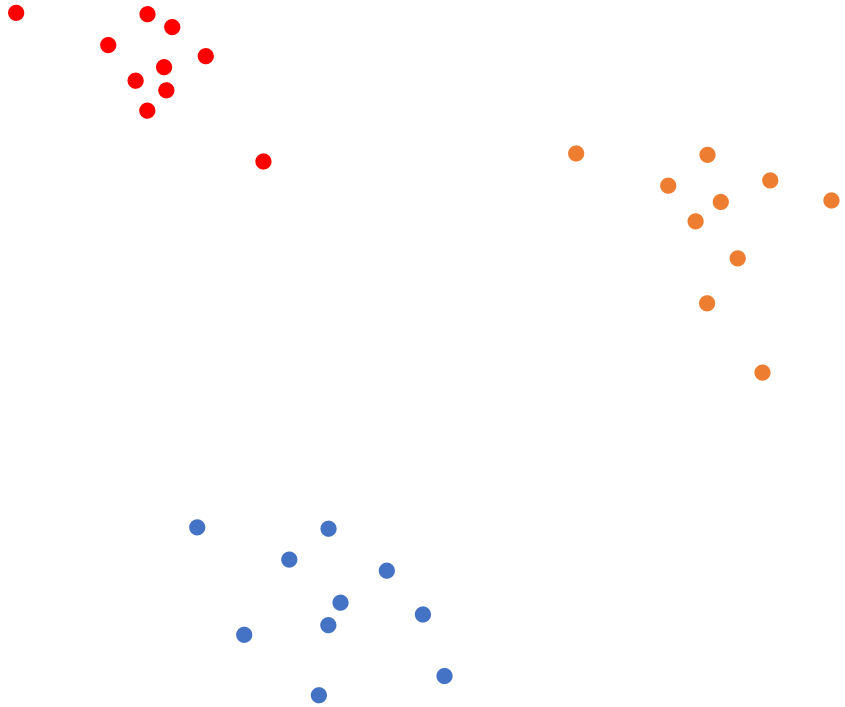
Provide k initial centers

Optimize the solution using k initial centers

The quality of Lloyd's heuristic depends on the initial centers.

k-means++ [Arthur-Vassilvitskii, SODA'07]

$k = 3$



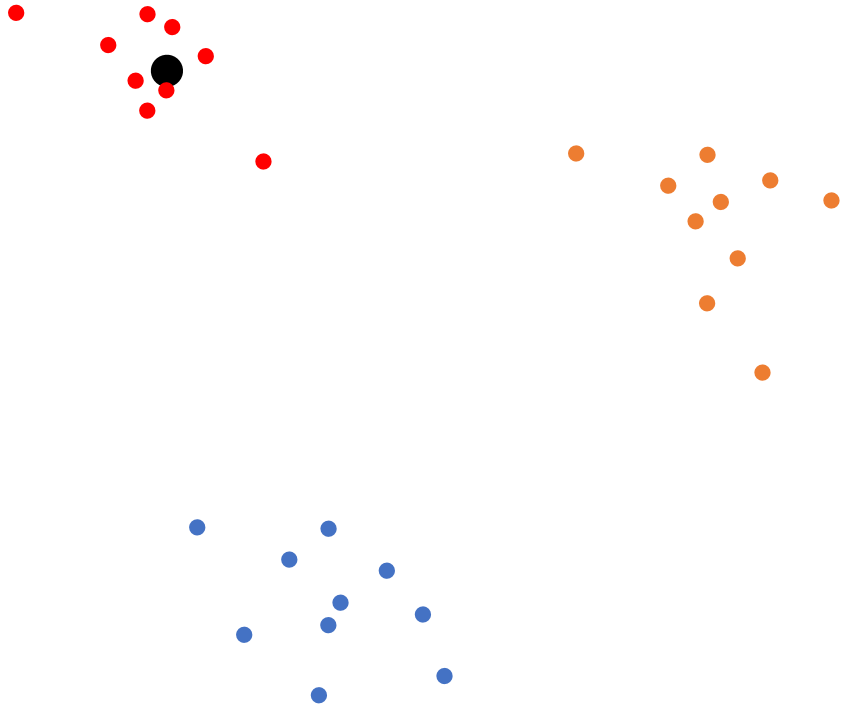
1. Choose an initial center c_1 uniformly at random from X.

2. Choose the next center c_i, with probability ***in proportional to*** the squared distance between the point to its closest chosen center.

3. Repeat Step 2 until we have k centers.

k-means++ [Arthur-Vassilvitskii, SODA'07]

$k = 3$



1. Choose an initial center c_1 uniformly at random from X.

2. Choose the next center c_i, with probability ***in proportional to*** the squared distance between the point to its closest chosen center.

3. Repeat Step 2 until we have k centers.

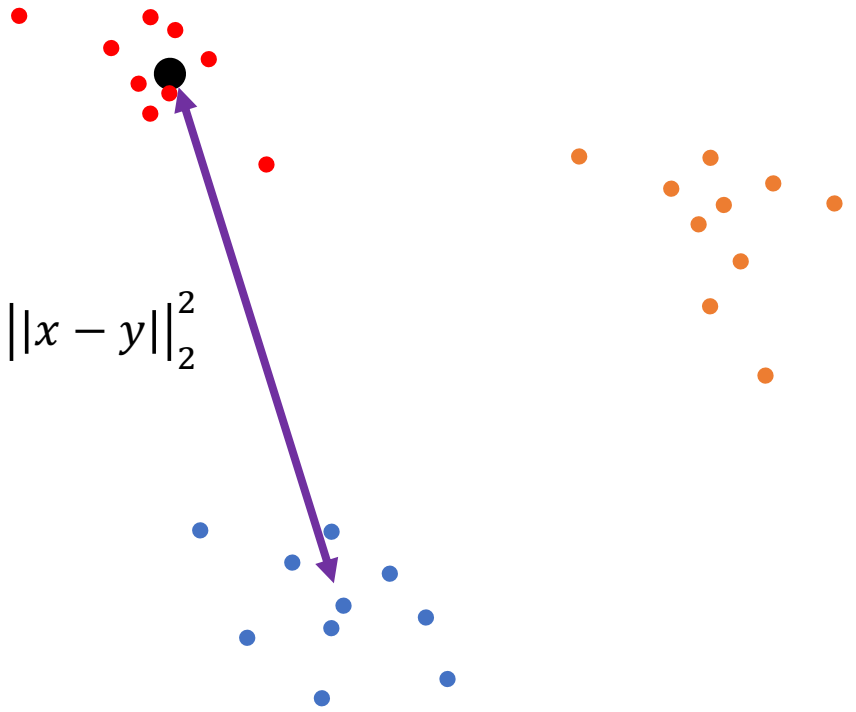
k-means++ [Arthur-Vassilvitskii, SODA'07]

$k = 3$

1. Choose an initial center c_1 uniformly at random from X.

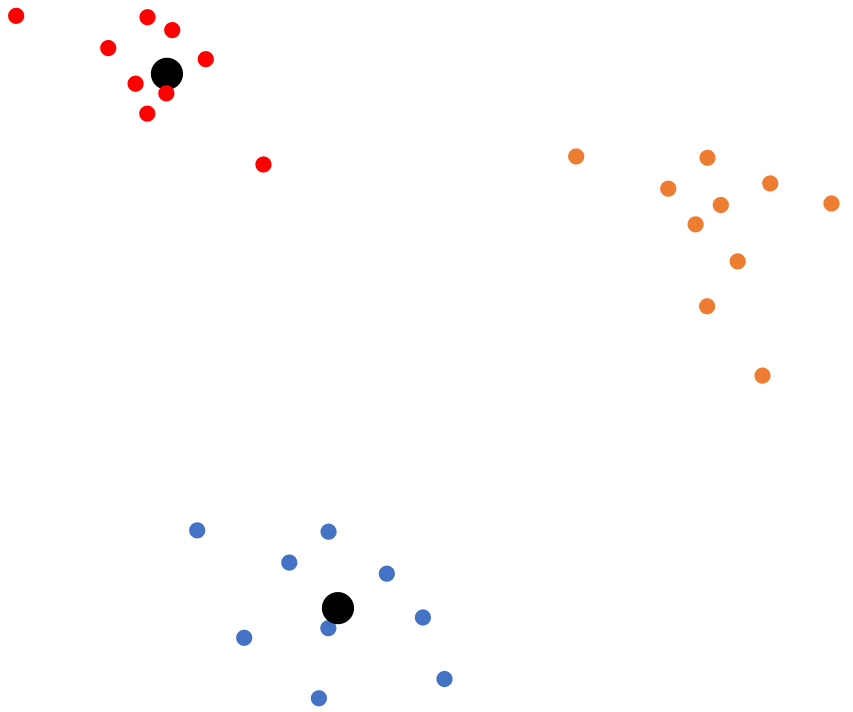
2. Choose the next center c_i, with probability ***in proportional to*** the squared distance between the point to its closest chosen center.

3. Repeat Step 2 until we have k centers.



k-means++ [Arthur-Vassilvitskii, SODA'07]

$k = 3$



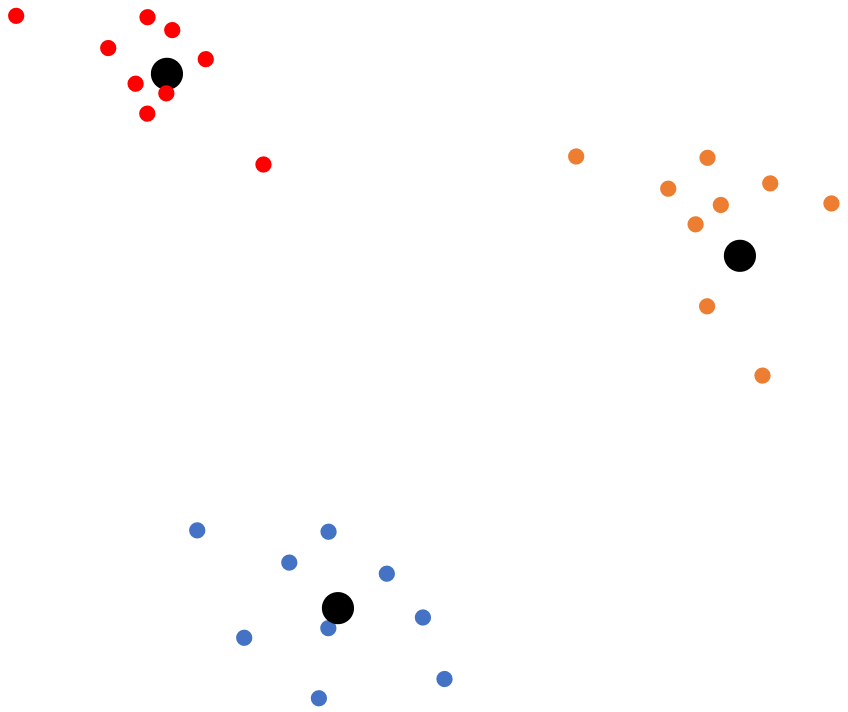
1. Choose an initial center c_1 uniformly at random from X.

2. Choose the next center c_i, with probability ***in proportional to*** the squared distance between the point to its closest chosen center.

3. Repeat Step 2 until we have k centers.

k-means++ [Arthur-Vassilvitskii, SODA'07]

$k = 3$



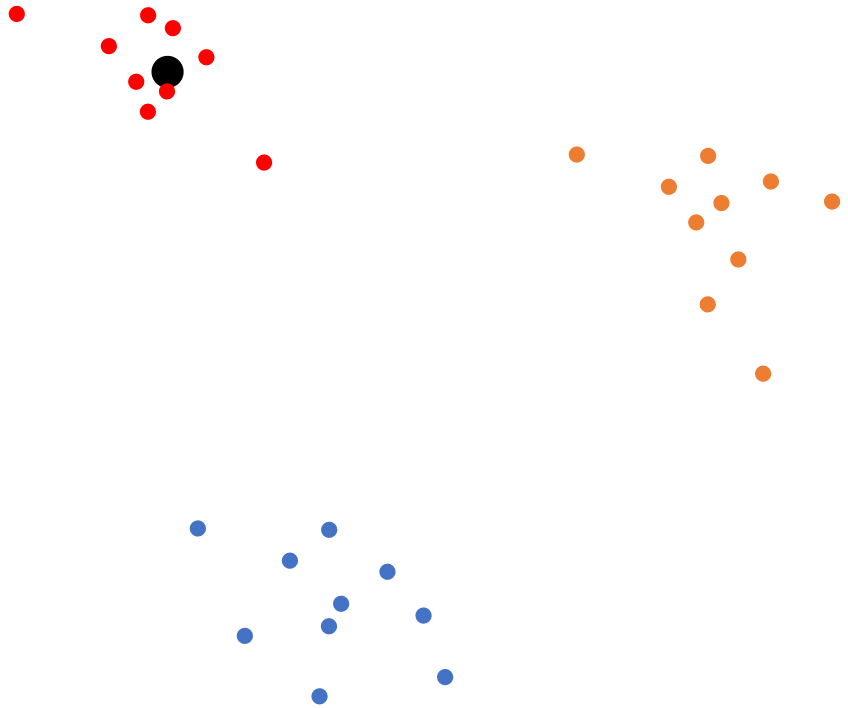
1. Choose an initial center c_1 uniformly at random from X.

2. Choose the next center c_i, with probability ***in proportional to*** the squared distance between the point to its closest chosen center.

3. Repeat Step 2 until we have k centers.

k-means++ [Arthur-Vassilvitskii, SODA'07]

$k = 3$

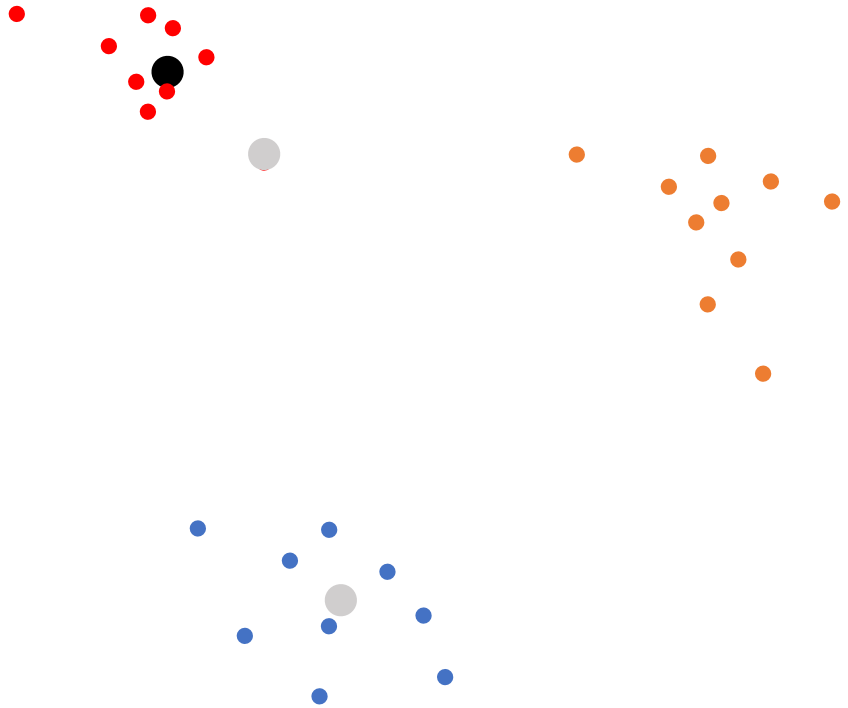


Greedy Variant

1. In each iteration, choose ℓ candidates, with probability ***in proportional to*** the squared distance between the point to its closest chosen center.
2. Select the one the minimizes the objective.
3. Repeat Step 2 until we have k centers.

k-means++ [Arthur-Vassilvitskii, SODA'07]

$k = 3$

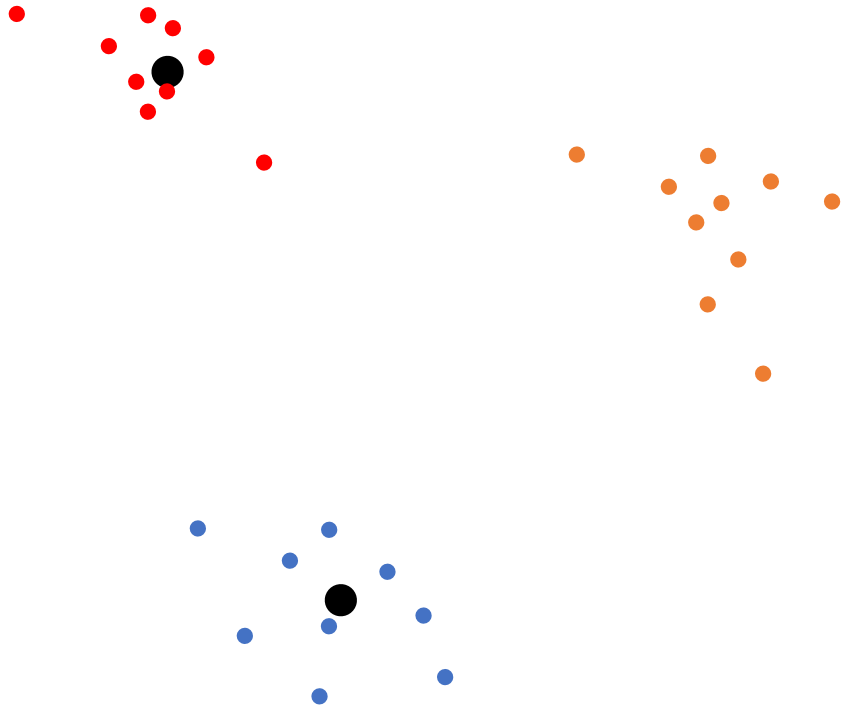


Greedy Variant

1. In each iteration, choose ℓ candidates, with probability ***in proportional to*** the squared distance between the point to its closest chosen center.
2. Select the one the minimizes the objective.
3. Repeat Step 2 until we have k centers.

k-means++ [Arthur-Vassilvitskii, SODA'07]

$k = 3$



Greedy Variant

1. In each iteration, choose ℓ candidates, with probability ***in proportional to*** the squared distance between the point to its closest chosen center.
2. Select the one the minimizes the objective.
3. Repeat Step 2 until we have k centers.

k-means++ [Arthur-Vassilvitskii, SODA'07]

1. Choose an initial center c_1 uniformly at random from X.

2. Choose the next center c_i, with probability ***in proportional to*** the squared distance between the point to its closest chosen center.

3. Repeat Step 2 until we have k centers.

Greedy Variant

1. In each iteration, choose ℓ candidates, with probability ***in proportional to*** the squared distance between the point to its closest chosen center.

2. Select the one the minimizes the objective.

3. Repeat Step 2 until we have k centers.

1. [Arthur-Vassilvitskii, SODA'07] suggests that the greedy performs better in practice.
2. Popular libraries such as Scikit-learn implement the greedy variant.

[Bhattacharya et al, ESA'20] shows that in worst scenarios,

the greedy k-means++ is $\Omega(\ell \log k)$-approximation,

while k-means++ is $O(\log k)$-approximation.

[Bhattacharya et al, ESA'20] shows that in worst scenarios,

the greedy k-means++ is $\Omega(\ell \log k)$-approximation,

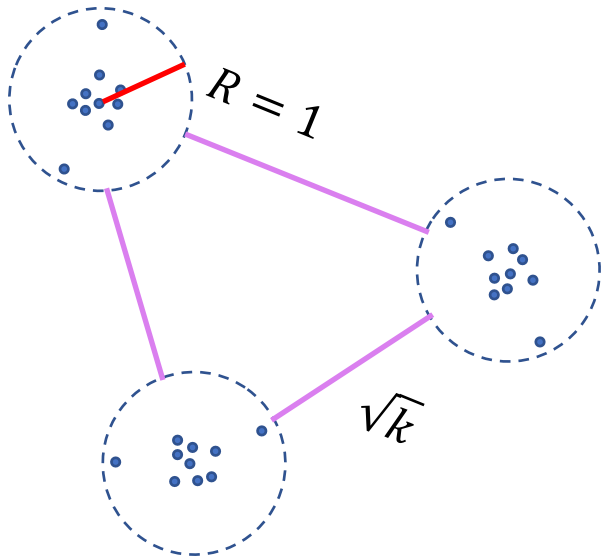
while k-means++ is $O(\log k)$-approximation.

Question: Why the greedy outperforms in practice?

Results

The greedy outperforms the k-means++ with the following instances.

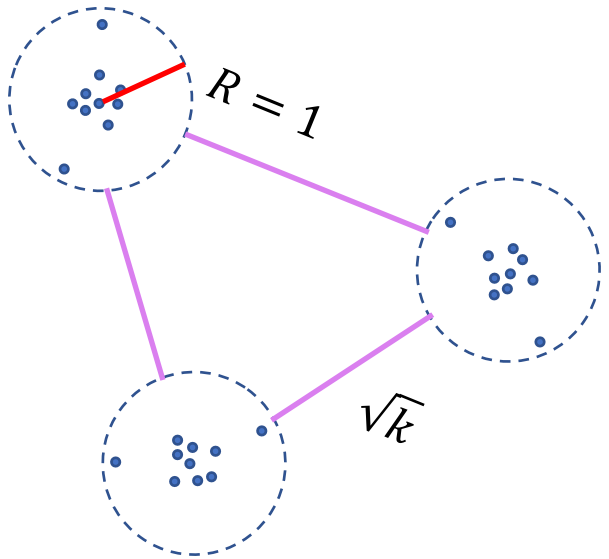
1. Points of each cluster follow an exponentially decaying distribution.
2. Clusters have a similar number of points
3. The distances between clusters are large k^{θ},where $\theta \in (0,1/2)$.



Results

The greedy outperforms the k-means++ with the following instances.

1. Points of each cluster follow an exponentially decaying distribution.
2. Clusters have a similar number of points
3. The distances between clusters are large k^{θ},where $\theta \in (0,1/2)$.



Theorem: Greedy k-means++ is $O((\ln \ln k)^2)$-approximation, while k-means++ is $\Omega(\ln k)$-approximation.

Conclusion

1. We give the first theoretical result towards closing the gap of the greedy outperforms the k-means++ in practice.

2. We present natural structural instances showing that the greedy has a better approximation ratio.