Fast Noise Removal for k-Means Clustering

AISTATS 2020

Sungjin Im
(University of California – Merced)

Mashid Montazer Qaem
(University of California – Merced)

Benjamin Moseley
(Carnegie Mellon)

Xiaorui Sun
(University of Illinois- Chicago)

Rudy Zhou
(Carnegie Mellon)

k-Means Clustering

- Clustering – group together data points that are similar into clusters
- k-Means Clustering:
  - Data points in d-dimensional Euclidean space

k-Means Clustering

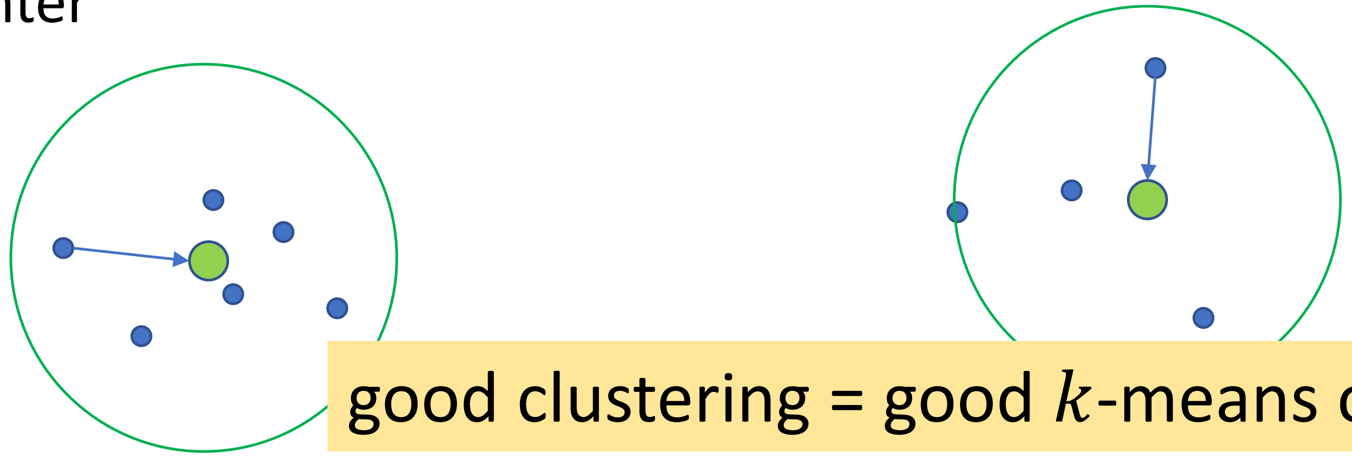
- Clustering – group together data points that are similar into clusters
- k-Means Clustering:
  - Data points in d-dimensional Euclidean space
  - Clusters defined by k **centers**

k-Means Clustering

- Clustering – group together data points that are similar into clusters
- k-Means Clustering:
  - Data points in d-dimensional Euclidean space
  - Clusters defined by k **centers**
  - Each data point is assigned to its closest center

k-Means Clustering

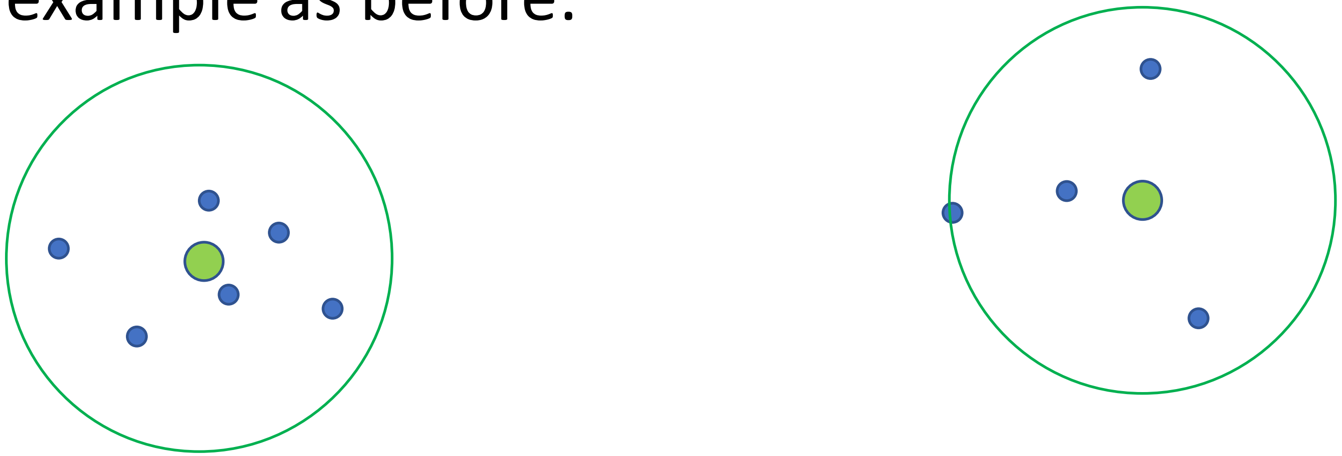
- Clustering – group together data points that are similar into clusters
- k-Means Clustering:
  - Data points in d-dimensional Euclidean space
  - Clusters defined by k **centers**
  - Each data point is assigned to its closest center
  - Goal is to minimize sum of squared distances of each data point to its closest center



good clustering = good k-means objective value

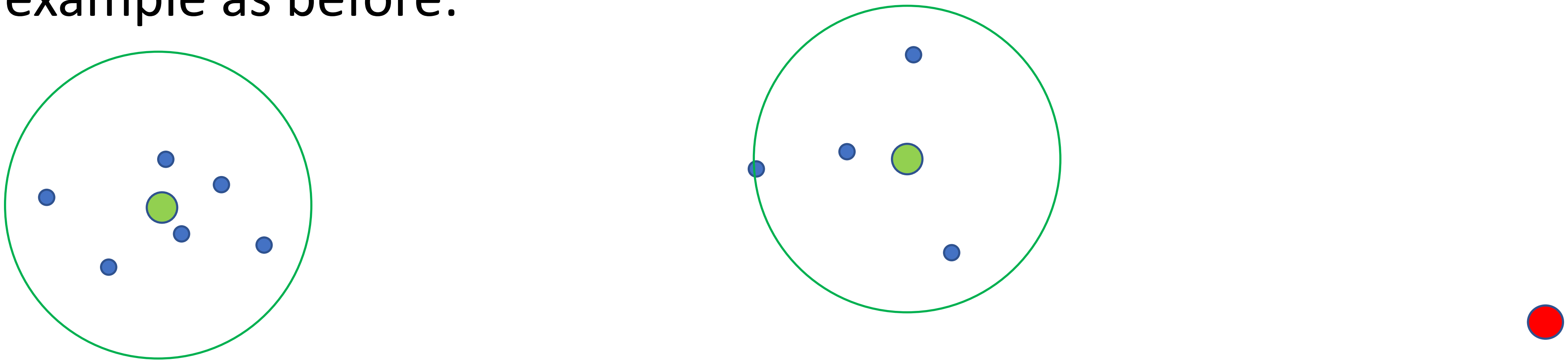
Problem – Noisy Data

- Same example as before:



Problem – Noisy Data

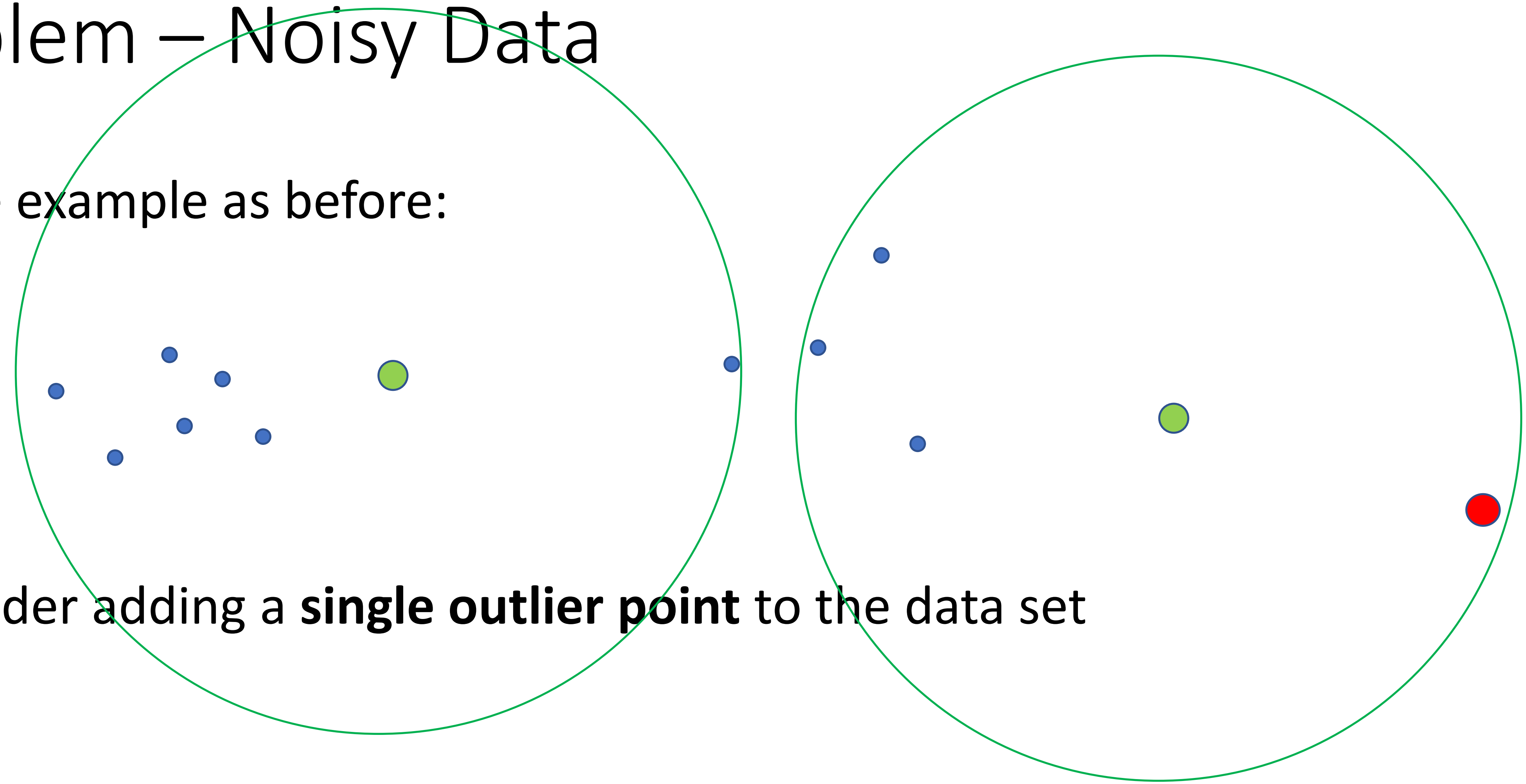
- Same example as before:



- Consider adding a **single outlier point** to the data set

Problem – Noisy Data

- Same example as before:



- Consider adding a **single outlier point** to the data set

Could significantly change objective value/optimal clustering

k-Means Clustering with Outliers

- Input:
  - n **datapoints** in d-dimensional Euclidean space
  - parameter k (number of **centers**)

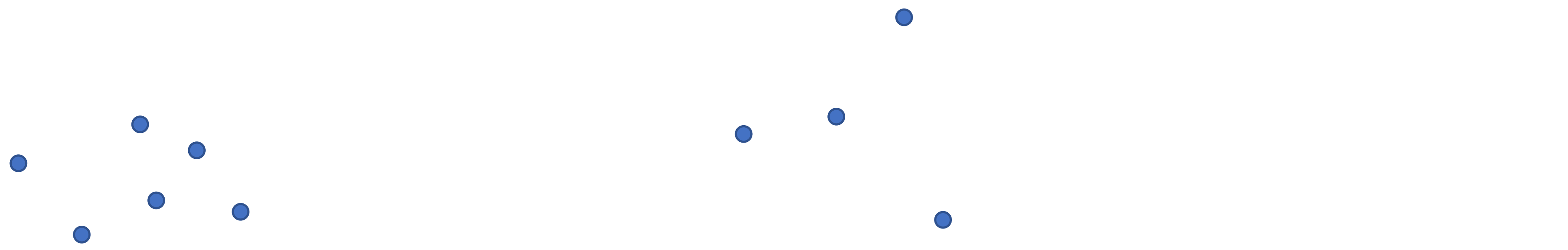
- Goal:
  - **($\boldsymbol{k}$-Means Objective)** Choose a set of $\boldsymbol{k}$ **centers** to minimize the sum of squared Euclidean distances from each datapoint to its closest center…

k-Means Clustering with Outliers

- Input:
  - n **datapoints** in d-dimensional Euclidean space
  - parameter k (number of **centers**)
  - parameter z (number of **outliers**)
- Goal:
  - **($\boldsymbol{k}$-Means Objective)** Choose a set of $\boldsymbol{k}$ **centers** to minimize the sum of squared Euclidean distances from each datapoint to its closest center…
  - **(Outlier Removal)** while ignoring $\boldsymbol{z}$ **outliers**

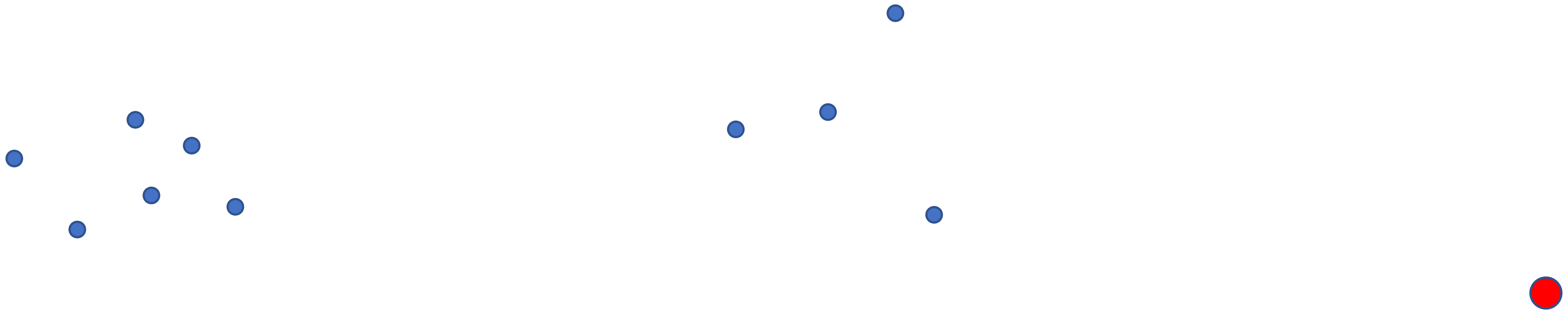
Solution to Previous Example

- Same example as before with one outlier point:
- $k = 2$ and $z = 1$



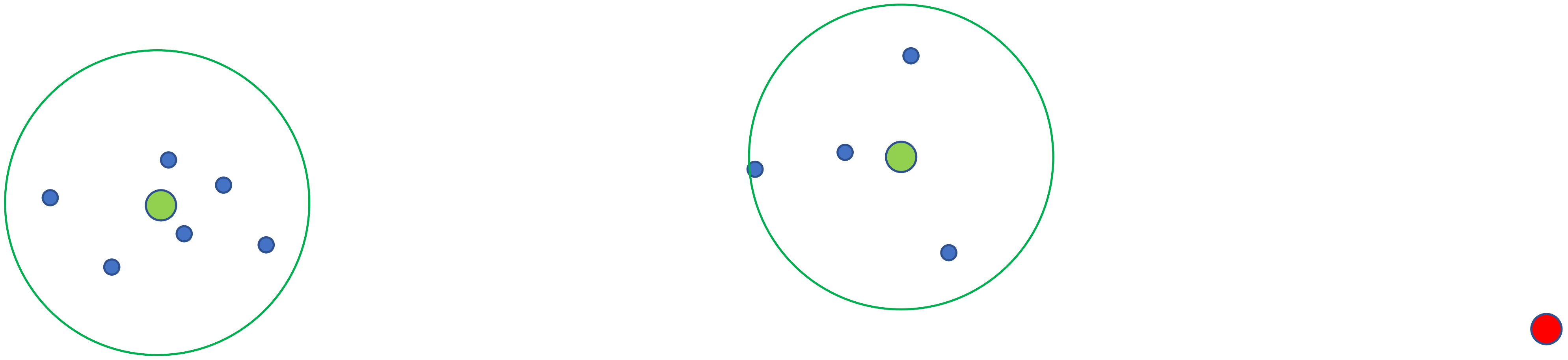
Solution to Previous Example

- Same example as before with one outlier point:
- $k = 2$ and $z = 1$
- Pick one outlier



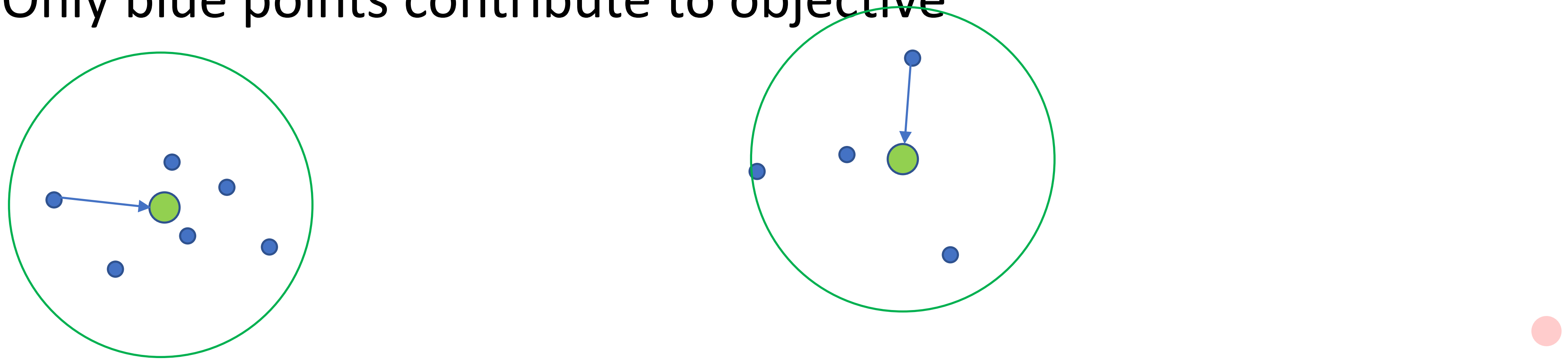
Solution to Previous Example

- Same example as before with one outlier point:
- $k = 2$ and $z = 1$
- Pick one outlier
- Pick two centers for blue points



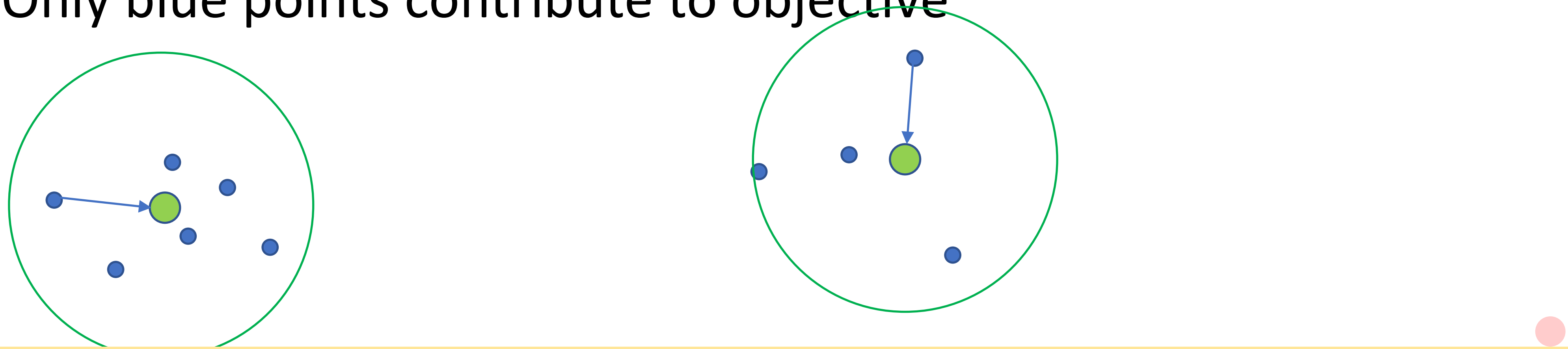
Solution to Previous Example

- Same example as before with one outlier point:
- $k = 2$ and $z = 1$
- Pick one outlier
- Pick two centers for blue points
- Only blue points contribute to objective



Solution to Previous Example

- Same example as before with one outlier point:
- $k = 2$ and $z = 1$
- Pick one outlier
- Pick two centers for blue points
- Only blue points contribute to objective



Noisy Data: good clustering = good k-means <u>with outliers</u> objective value

Optimizing the Outliers Objective

- Pick k centers and z outliers that are competitive with the optimal centers and outliers

Optimizing the Outliers Objective

- Pick k centers and z outliers that are competitive with the optimal centers and outliers
- NP-Hard to solve optimally $\Rightarrow$ try to solve <u>approximately</u>

Optimizing the Outliers Objective

- Pick k centers and z outliers that are competitive with the optimal centers and outliers
- NP-Hard to solve optimally $\Rightarrow$ try to solve <u>approximately</u>
- **Approximation Ratio** (minimization problem)
  - Opt = optimal objective value
  - Alg = algorithm's objective value
  - $\boldsymbol{Approximation\ Ratio = \frac{Alg}{Opt} \geq 1}$

Approximating the Outliers Objective

- **Approximation Algorithms:**
  - local search [5,6]
  - linear program-based algorithms [2]
  - k-means++ with thresholding [1]
- **Heuristics:**
  - DBSCAN [4]
  - k-means-- [3]

Approximating the Outliers Objective

- **Approximation Algorithms:**
  - local search [5,6]
  - linear program-based algorithms [2]
  - k-means++ with thresholding [1]
- **Heuristics:**
  - DBSCAN [4]
  - k-means-- [3]

Simultaneously solve outlier removal and clustering tasks

Approximating the Outliers Objective

- **Approximation Algorithms:**
  - local search [5,6]
  - linear program-based algorithms [2]
  - k-means++ with thresholding [1]
- **Heuristics:**
  - DBSCAN [4]
  - k-means-- [3]

Simultaneously solve outlier removal and clustering tasks

Our Approach: First solve outlier removal and then clustering

Main Result – Outlier Removal Algorithm

- Runs in near linear time: $\boldsymbol{O(kdn\log^2 n)}$
- Removes $\boldsymbol{O(kz)}$ points from data set as outliers…

Main Result – Outlier Removal Algorithm

- Runs in near linear time: $\boldsymbol{O(kdn \log^2 n)}$
- Removes $\boldsymbol{O(kz)}$ points from data set as outliers…
- such that picking k good centers for remaining points gives a good solution

Main Result – Outlier Removal Algorithm

- Runs in near linear time: $\boldsymbol{O(kdn\log^2 n)}$
- Removes $\boldsymbol{O(kz)}$ points from data set as outliers…
- such that picking k good centers for remaining points gives a good solution

Formally: Running any α-approximate k-means algorithm on remaining points gives $O(\alpha)$-approximation for k-means with outliers.

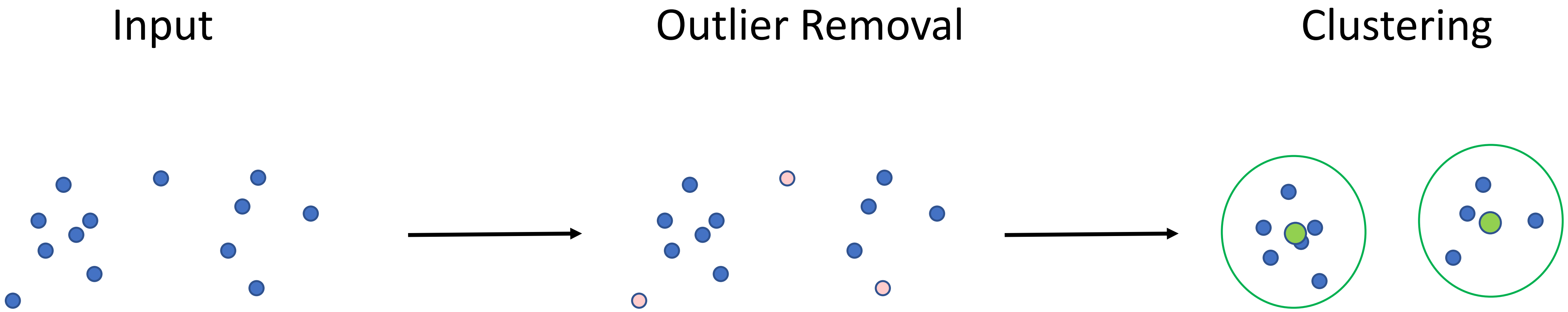
Main Result – Outlier Removal Algorithm

- Runs in near linear time: $\boldsymbol{O(kdn\log^2 n)}$
- Removes $\boldsymbol{O(kz)}$ points from data set as outliers…
- such that picking k good centers for remaining points gives a good solution

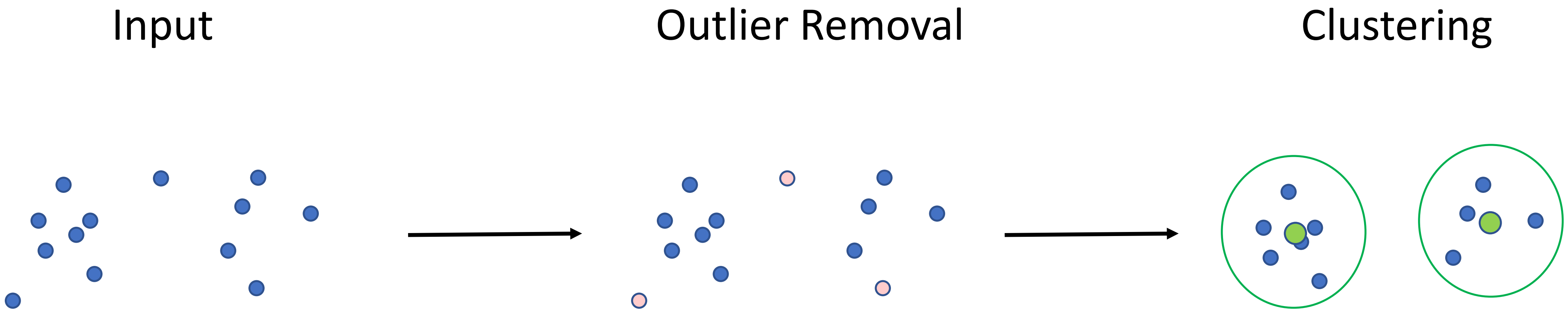
Formally: Running any α-approximate k-means algorithm on remaining points gives $O(\alpha)$-approximation for k-means with outliers.

If every optimal cluster has size $\geq 3z$, then only removes $\boldsymbol{O(z)}$ points

Separating Outlier Removal and Clustering



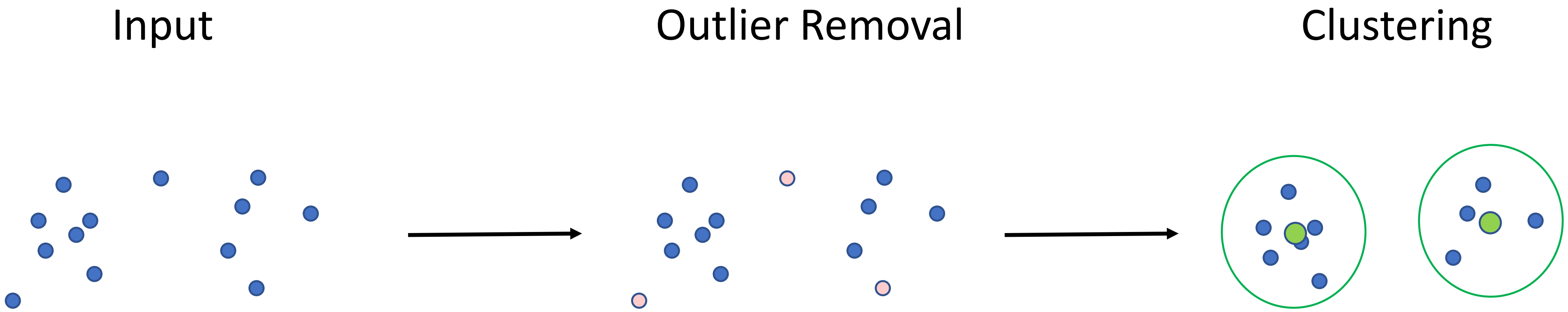
Separating Outlier Removal and Clustering



Our algorithm

Any k-means algorithm

Separating Outlier Removal and Clustering



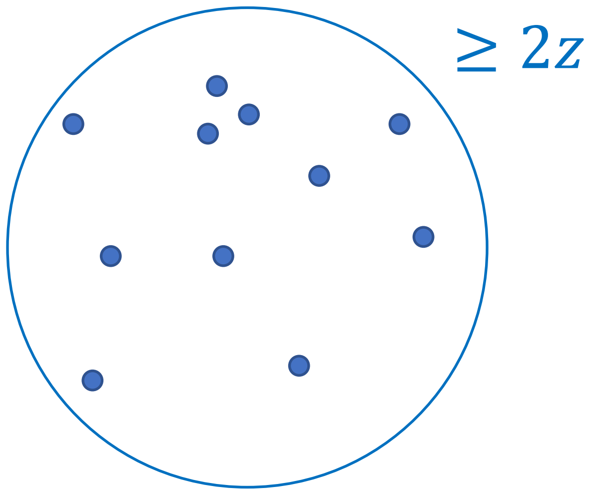
Our algorithm

Any k-means algorithm

Algorithms for basic k-means are fast and have good performance with no noise $\Rightarrow$ Remove outliers to **reduce** problem to basic k-means

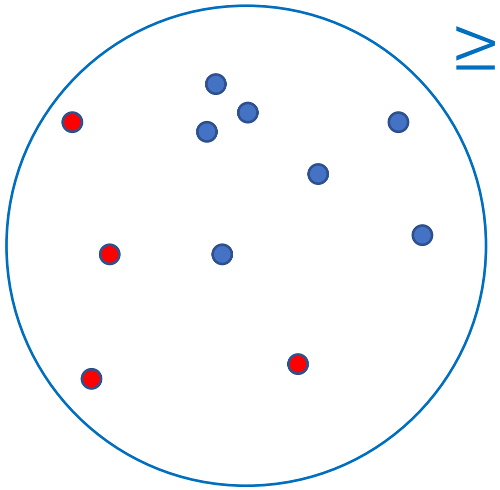
Outlier Removal Algorithm- Key Idea

- Pick outliers to remove = pick points to keep
- Consider any ball of $\geq 2z$ data points



Outlier Removal Algorithm- Key Idea

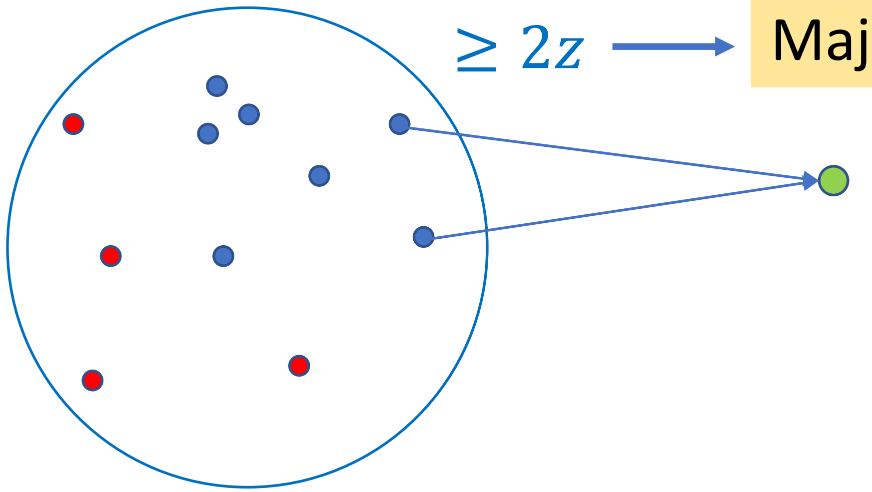
- Pick outliers to remove = pick points to keep
- Consider any ball of $\geq 2z$ data points



$\geq 2z$ → Majority of ball is real points

Outlier Removal Algorithm- Key Idea

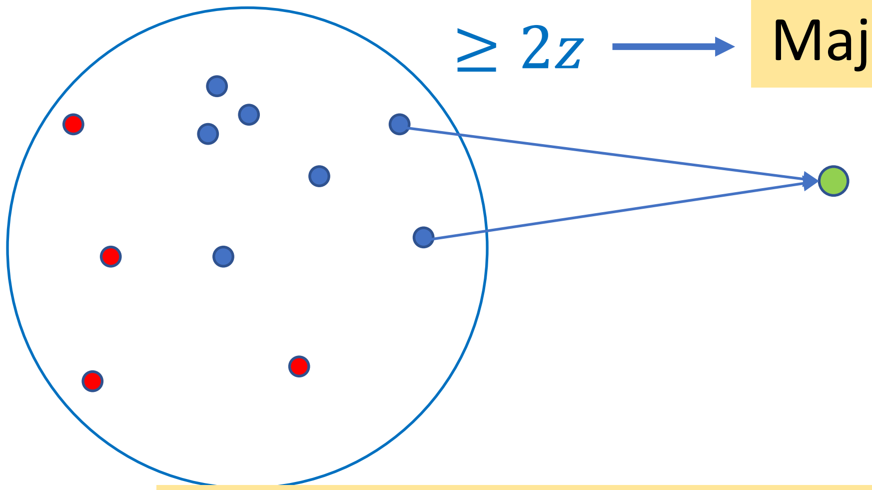
- Pick outliers to remove = pick points to keep
- Consider any ball of $\geq 2z$ data points



$\geq 2z$ → Majority of ball is real points → Optimal center nearby

Outlier Removal Algorithm- Key Idea

- Pick outliers to remove = pick points to keep
- Consider any ball of $\geq 2z$ data points



$\geq 2z$ → Majority of ball is real points → Optimal center nearby

Algorithm Sketch: Keep balls containing many points

Summary

- Known k-means algorithms are fast and have good performance with no noise $\Rightarrow$ remove good outliers and let a k-means algorithm handle the rest

Summary

- Known k-means algorithms are fast and have good performance with no noise $\Rightarrow$ remove good outliers and let a k-means algorithm handle the rest

Our Contribution: Near linear time algorithm that removes $O(kz)$ good outliers, and only $O(z)$ when optimal clusters are large

Summary

- Known k-means algorithms are fast and have good performance with no noise $\Rightarrow$ remove good outliers and let a k-means algorithm handle the rest

Our Contribution: Near linear time algorithm that removes $O(kz)$ good outliers, and only $O(z)$ when optimal clusters are large

Open Question: Can we find a fast algorithm that only removes $O(z)$ good outliers?

Thank You!

References

1. *Aditya Bhaskara, Sharvaree Vadgama, Hong Xu: Greedy Sampling for Approximate Clustering in the Presence of Outliers. NeurIPS 2019: 11146-11155*
2. *Moses Charikar, Samir Khuller, David M. Mount, Giri Narasimhan: Algorithms for facility location problems with outliers. SODA 2001: 642-651*
3. *Sanjay Chawla, Aristides Gionis: k-means-: A Unified Approach to Clustering and Outlier Detection. SDM 2013: 189-197*
4. *Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. KDD 1996: 226-231*
5. *Zachary Friggstad, Kamyar Khodamoradi, Mohsen Rezapour, Mohammad R. Salavatipour: Approximation Schemes for Clustering with Outliers. ACM Trans. Algorithms 15(2): 26:1-26:26 (2019)*
6. *Shalmoli Gupta, Ravi Kumar, Kefu Lu, Benjamin Moseley, Sergei Vassilvitskii: Local Search Methods for k-Means with Outliers. Proc. VLDB Endow. 10(7): 757-768 (2017)*