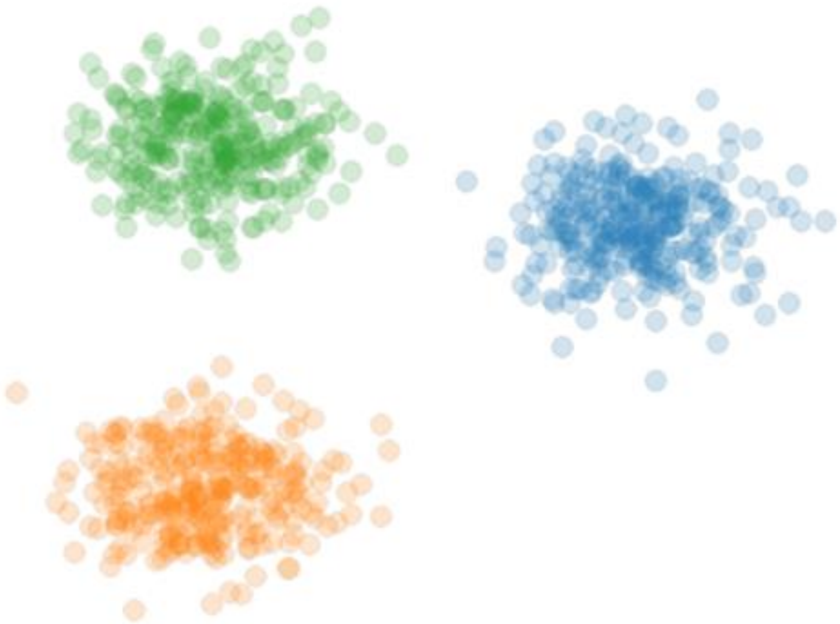
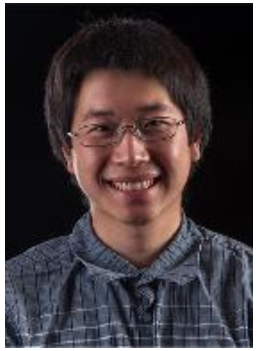
Learning Augmented Algorithms for k-Means Clustering

Samson Zhou





Learning Augmented Algorithms for k-Means Clustering

Jon Ergun
Zhili Feng
Sandeep Silwal
David P. Woodruff
Samson Zhou

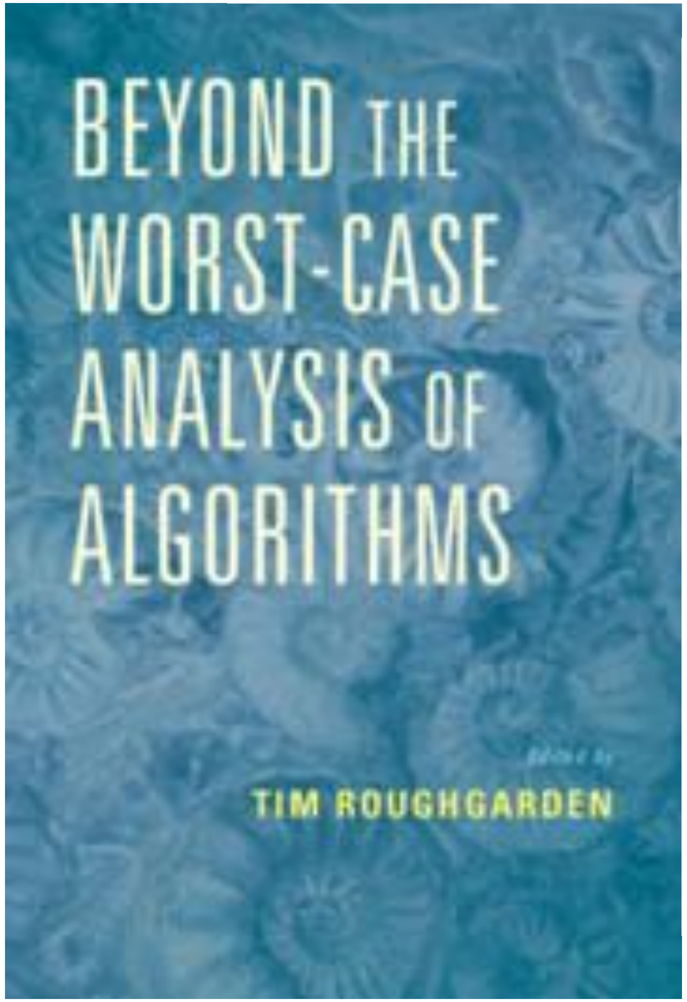
[EFSWZ22]

Thy Nguyen
Anamay Chatuvedi
Huy Lê Nguyễn

[NCN23]

Stellar™ 6.890 Learning-Augmented Algorithms

LOGIN



Learning-Augmented Algorithms

- For a certain task and input, algorithm is given advice

- Advice could be “good”, advice could be “bad”

- Goal: “Good” performance if the advice is good, “normal” performance if the advice is bad


7:08
LTE
Mahabalipuram
73°
Partly Cloudy
H:86° L:72°
Sunny conditions expected around 9AM.
Now 8AM 9AM 10AM 11AM 12P
73° 76° 79° 82° 84° 85
10-DAY FORECAST
Today 72° 86°
Fri 72° 86°
Sat 71° 85°
7:08
LTE
Mahabalipuram
77°
Partly Cloudy
H:84° L:76°
Sunny conditions expected around 9AM.
Now 8AM 9AM 10AM 11AM
77° 78° 80° 82° 83°
10-DAY FORECAST
Today 76° 84°
Fri 75° 84°
Sat 74° 85°
Sun 74° 85°
7:08
LTE
Mahabalipuram
74°
Partly Cloudy
H:85° L:73°
Sunny conditions expected around 9AM.
Now 8AM 9AM 10AM 11AM 12PM
74° 76° 79° 81° 83° 84°
10-DAY FORECAST
Today 73° 85°
Fri 73° 85°
Sat 72° 84°
Sun 72° 84°

Learning-Augmented Algorithms

- Better data structures: Bloom filters with lower false positive rates [Mitzenmacher18], Binary search [LinLuoWoodruff22]
- Better space-accuracy tradeoff for streaming algorithms: Frequency estimation, e.g., CountMin, CountSketch [HsuIndykKatabiVakilian19], moment estimation, distinct elements [JiangLinRuanWoodruff20], triangle counting [ChenEdenIndykLinNarayananRubinfeldSilwalWagnerWoodruffZhang22]
- Better size-accuracy tradeoff for sketching: Low-rank approximation [IndykVakilianYuan19]

Learning-Augmented Algorithms

- Warm-start to search algorithms: Max-flow [ChenSilwalVakilianZhang22], [DaviesMoseleyVassilvitskiiWang23], matchings [DinitzImLavastidaMoseleyVassilvitskii21]
- Better accuracy-sample complexity tradeoff: Support size estimation [EdenIndykNarayananRubinfeldSilwalWagner21]
- Better online algorithms: Set cover [BamasMaggioriSvensson20], [GrigorescuLinSilwalSongZhou23], Scheduling [LattanziLavastidaMoseleyVassilvitskii20], [ScullyGrosofMitzenmacher22]
- Better privacy-utility tradeoffs for DP: Quantile estimation [KhodakAminDickVassilvitskii23]
- Beating NP-hardness?

https://algorithms-with-predictions.github.io

Algorithms with Predictions

PAPER LIST FURTHER MATERIAL ABOUT

'07 '09 '10 '17 '18 '19 '20 '21 '22

Newest first

122 papers

Graph Searching with Predictions Banerjee, Cohen-Addad, Gupta, Li arXiv '22 exploration online search

Scheduling with Predictions Cho, Henderson, Shmoys arXiv '22 online scheduling

On the Power of Learning-Augmented BSTs Chen, Chen arXiv '22 data structure search

Algorithms with Prediction Portfolios Dinitz, Im, Lavastida, Moseley, Vassilvitskii arXiv '22 load balancing matching multiple predictions online scheduling

Private Algorithms with Private Predictions Amin, Dick, Khodak, Vassilvitskii arXiv '22 differential privacy

Paging with Succinct Predictions Antoniadis, Boyar, Eliáš, Favrholdt, Hoeksma, Larsen, Polak, Simon arXiv '22 caching/paging online

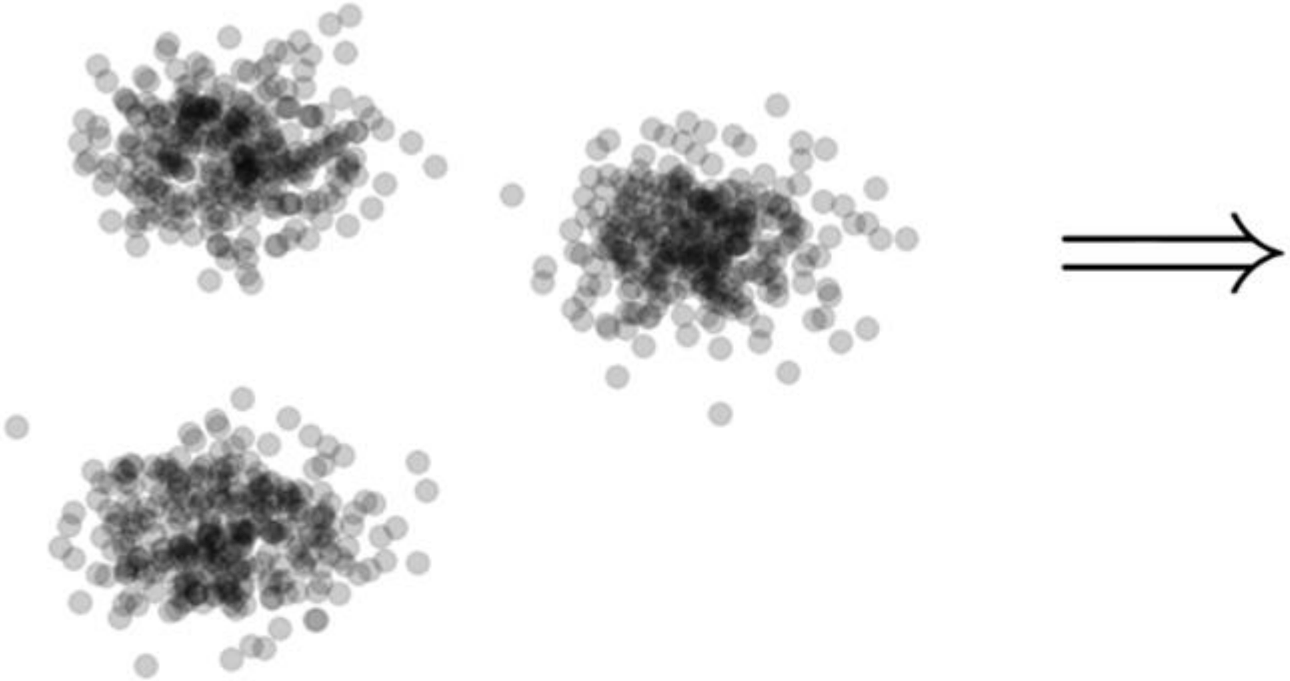
Proportionally Fair Online Allocation of Public Goods with Predictions Banerjee, Gkatzelis, Hossain, Jin, Micha, Shah arXiv '22 allocation online

Canadian Traveller Problem with Predictions Bampis, Escoffier, Xefteris arXiv '22 WAOA '22 online routing

Learning-Augmented Algorithms for Online Linear and Semidefinite Programming Grigorescu, Lin, Silwal, Song, Zhou arXiv '22 covering problems online SDP

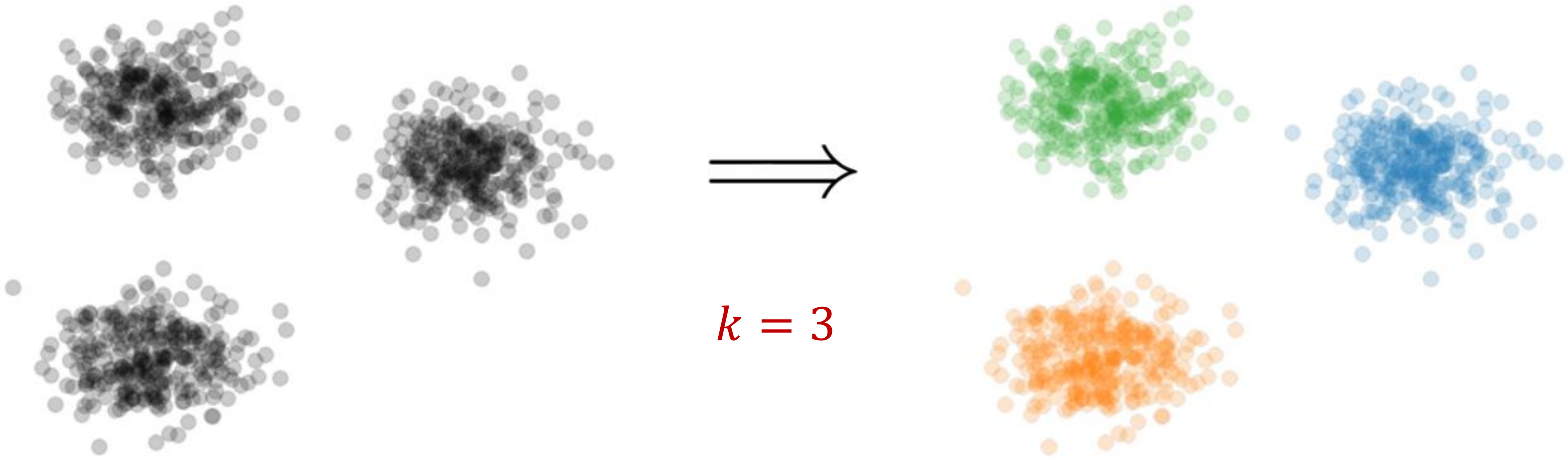
Clustering

- Goal: Given input dataset X, partition X so that “similar” points are in the same cluster and “different” points are in different clusters



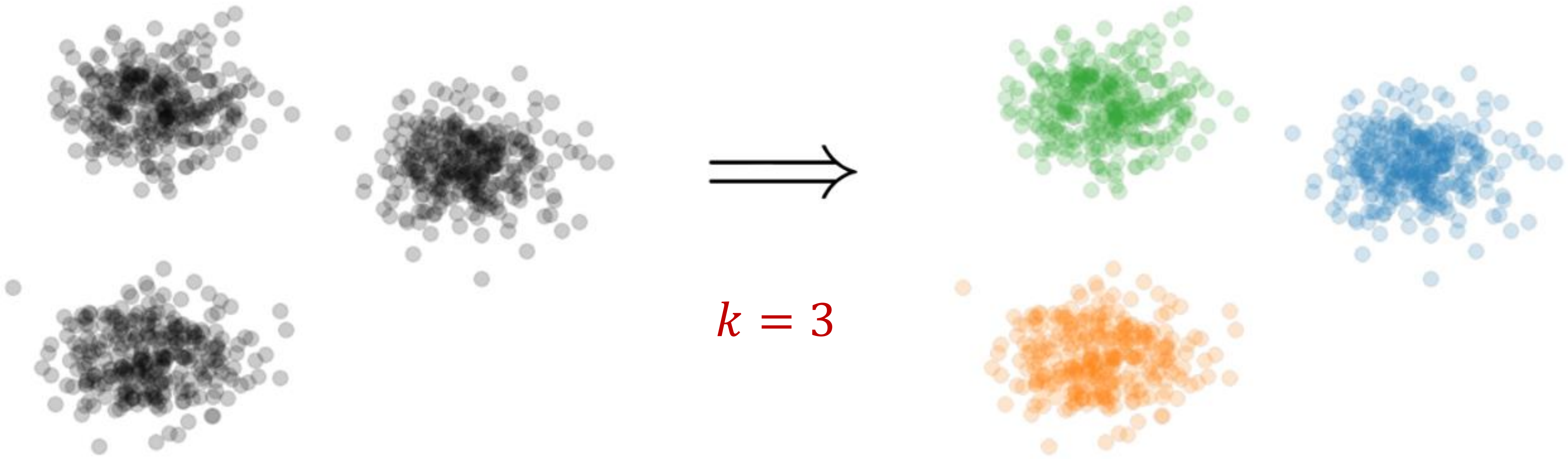
k-Clustering

- Goal: Given input dataset X, partition X so that "similar" points are in the same cluster and "different" points are in different clusters
- There can be at most k different clusters



k-Clustering

- Question: How do we measure the "quality" of each clustering?



k-Clustering

- Question: How do we measure the "quality" of each clustering?
- Assign a "center" c_i to each cluster
- Have a cost function induced by c_i for all of the points P_i assigned to cluster i

k-Clustering

- Question: How do we measure the "quality" of each clustering?

- Assign a "center" c_i to each cluster

- Have a cost function induced by c_i for all of the points P_i assigned to cluster i
  - Assume points are in metric space with distance function $\mathrm{dist}(\cdot,\cdot)$
  - Define $\mathrm{Cost}(P_i, c_i)$ to be a function of $\{\mathrm{dist}(x, c_i)\}_{x \in P_i}$

k-Clustering

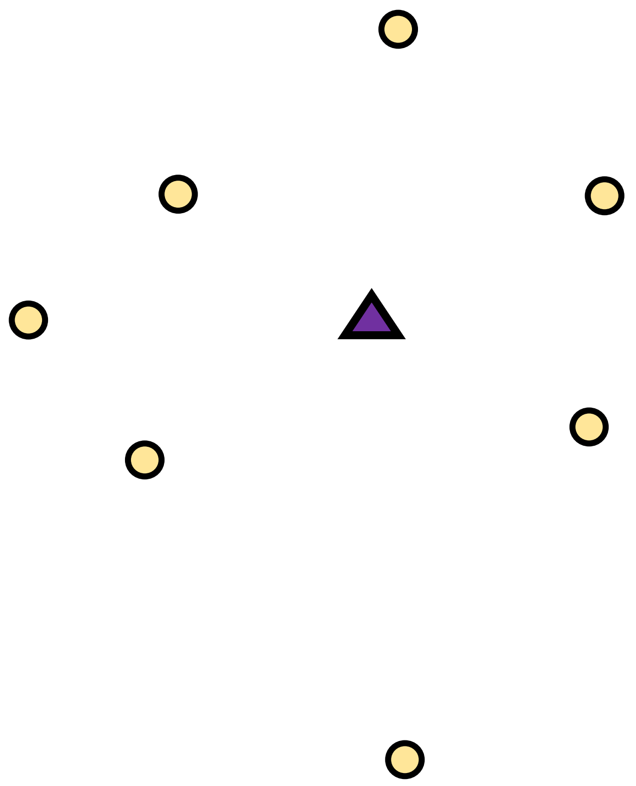
- Question: How do we measure the "quality" of each clustering?

- Have a cost function induced by c_i for all of the points P_i assigned to cluster i
  - Define $\text{Cost}(P_i, c_i)$ to be a function of $\{\text{dist}(x, c_i)\}_{x \in P_i}$

- Suppose the set of centers is $C = \{c_1, \dots, c_k\}$
  - Define clustering cost $\text{Cost}(X, C)$ to be a function of $\{\text{dist}(x, C)\}_{x \in C}$

k-Clustering

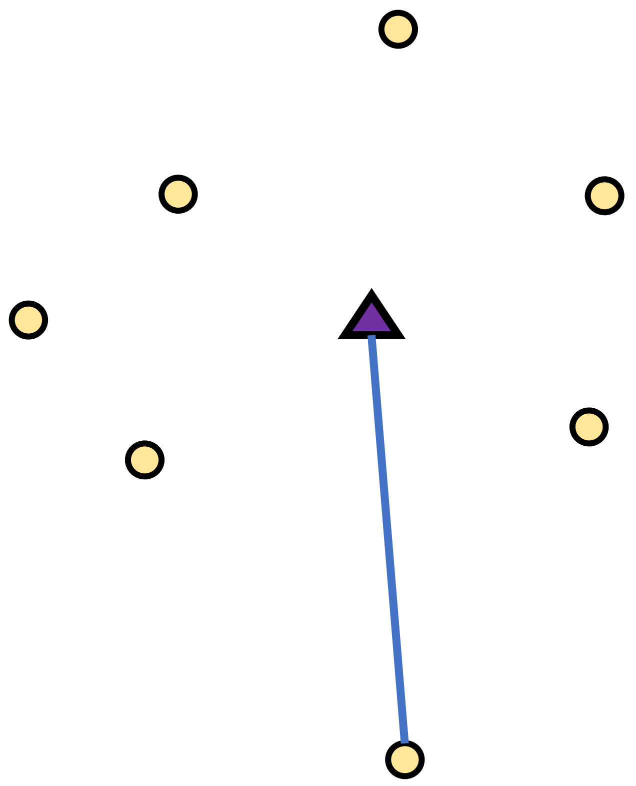
- Define clustering cost $\mathrm{Cost}(X, C)$ to be a function of $\{\mathrm{dist}(x, C)\}_{x \in C}$



k-Clustering

- Define clustering cost $\mathrm{Cost}(X, C)$ to be a function of $\{\mathrm{dist}(x, C)\}_{x \in C}$

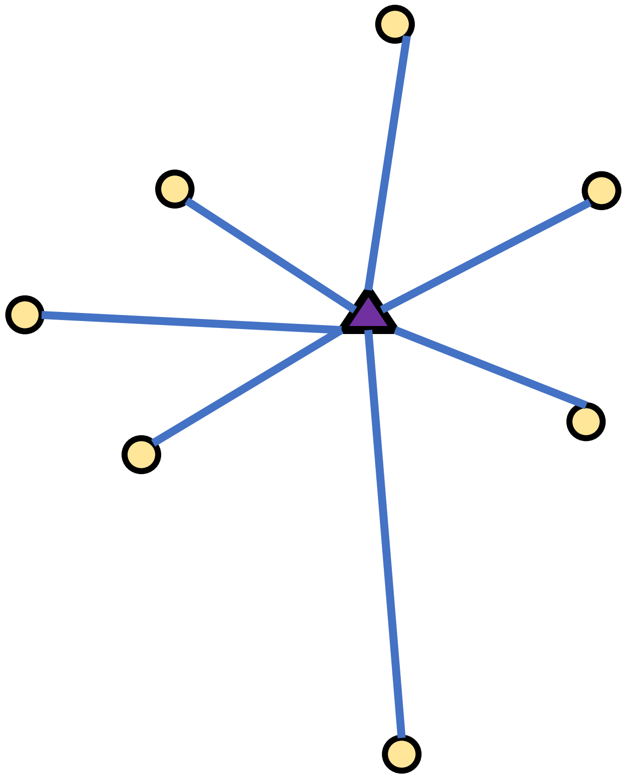
- k-center: $\mathrm{Cost}(X, C) = \max_{x \in X} \mathrm{dist}(x, C)$



k-Clustering

- Define clustering cost $\mathrm{Cost}(X, C)$ to be a function of $\{\mathrm{dist}(x, C)\}_{x \in C}$

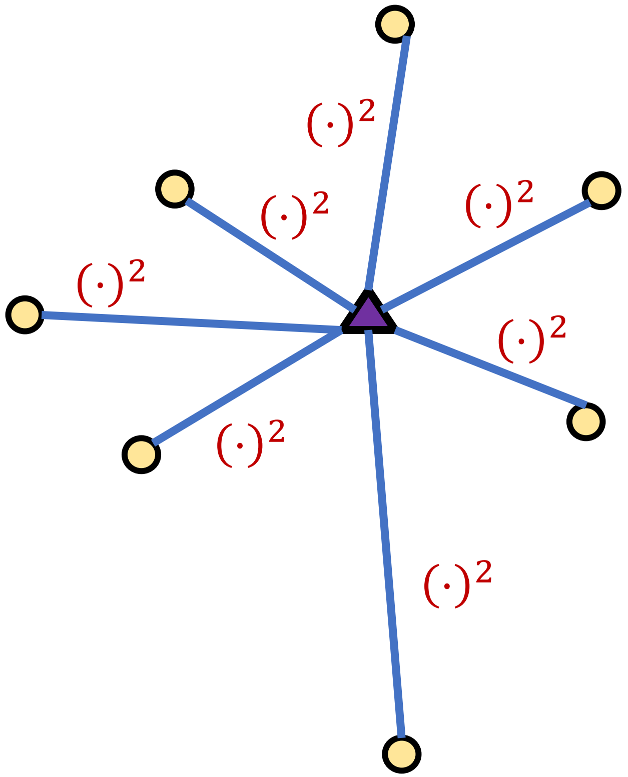
- k-center: $\mathrm{Cost}(X, C) = \max_{x \in X} \mathrm{dist}(x, C)$
- k-median: $\mathrm{Cost}(X, C) = \sum_{x \in X} \mathrm{dist}(x, C)$



k-Clustering

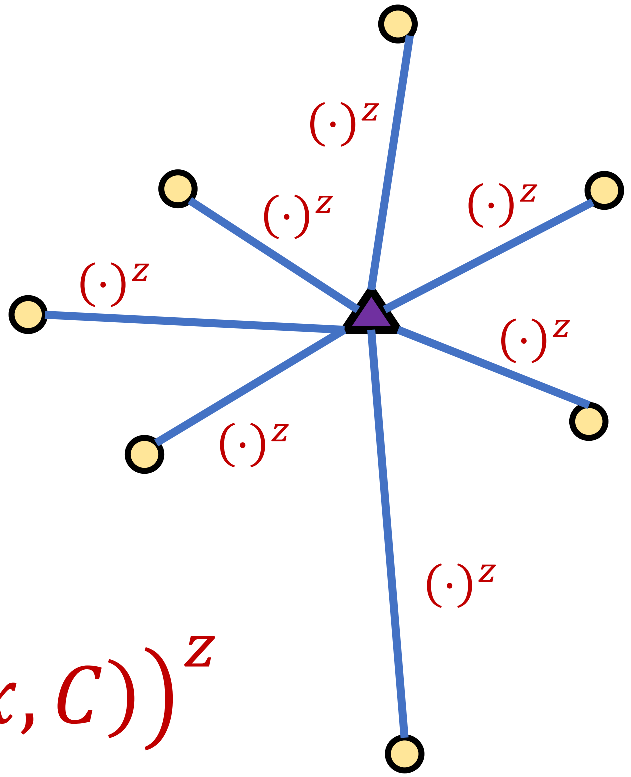
- Define clustering cost $\mathrm{Cost}(X, C)$ to be a function of $\{\mathrm{dist}(x, C)\}_{x \in C}$

- k-center: $\mathrm{Cost}(X, C) = \max_{x \in X} \mathrm{dist}(x, C)$
- k-median: $\mathrm{Cost}(X, C) = \sum_{x \in X} \mathrm{dist}(x, C)$
- k-means: $\mathrm{Cost}(X, C) = \sum_{x \in X} \left(\mathrm{dist}(x, C)\right)^2$



k-Clustering

- Define clustering cost $\mathrm{Cost}(X, C)$ to be a function of $\{\mathrm{dist}(x, C)\}_{x \in C}$



- k-center: $\mathrm{Cost}(X, C) = \max_{x \in X} \mathrm{dist}(x, C)$
- k-median: $\mathrm{Cost}(X, C) = \sum_{x \in X} \mathrm{dist}(x, C)$
- k-means: $\mathrm{Cost}(X, C) = \sum_{x \in X} \left(\mathrm{dist}(x, C)\right)^2$
- (k, z)-clustering: $\mathrm{Cost}(X, C) = \sum_{x \in X} \left(\mathrm{dist}(x, C)\right)^z$

Euclidean k-Clustering

- For Euclidean k-clustering, input points $X = x_1, \dots, x_n$ are in $\mathbb{R}^d$ (for us, they will be in $[\Delta]^d := \{1,2, \dots, \Delta\}^d$)

- $\mathrm{dist}(x, y) = \|x - y\| = \sqrt{(x_1 - y_1)^2 + \cdots + (x_d - y_d)^2}$ is the Euclidean distance

- (k, z)-clustering problem:

$$\min_{C:|C|\leq k} \mathrm{Cost}(X, C) = \min_{C:|C|\leq k} \Sigma_{x\in X}\big(\mathrm{dist}(x, C)\big)^z$$

Learning-Augmented Clustering

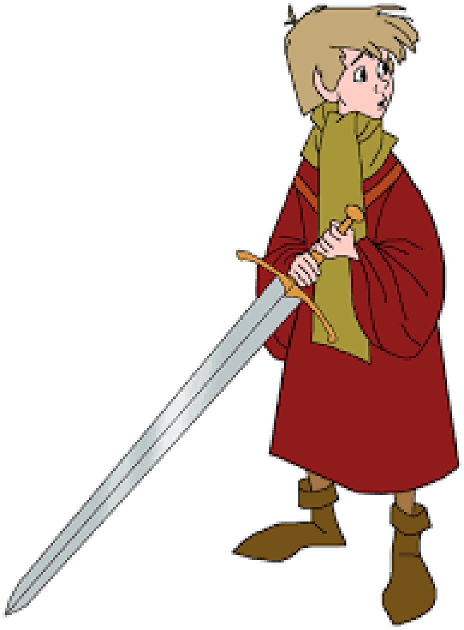
- Goal: Given dataset X in d dimensions, output a set C of k centers to minimize

$$\sum_{x \in X} \min_{c \in C} \|x - c\|_2^2$$

- NP-hard to even approximate within a factor of 1.07 [Cohen-AddadC.S.20, LeeSchmidtWright17]
- Beyond worst-case: Clustering on inputs from some “nice” distribution, similar inputs or inputs with auxiliary information
- Hope: ML can guide the clustering, so we can overcome worst-case with advice!

Predictor

- Suppose Π outputs noisy labels according to a $(1+\alpha)$ approximate clustering C and error rate $\lambda \leq \alpha$



What is the label of x_1?

x_1 belongs to cluster 3

What is the label of x_2?

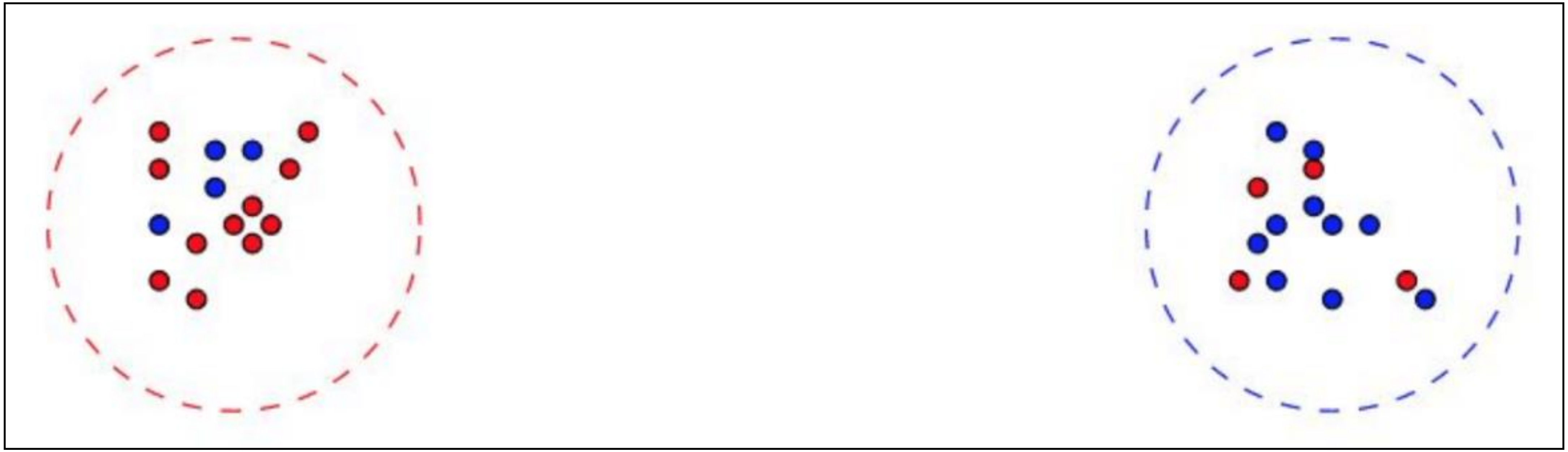
x_2 belongs to cluster 7

Theoretical Guarantee

- Suppose Π outputs noisy labels according to a $(1+\alpha)$ approximate clustering C and error rate $\lambda \leq \alpha$
- Main result [EFSWZ22]: Algorithm that outputs a $(1+O(\alpha))$ approximate k-means clustering in nearly linear time
- "Predictions can overcome complexity hardness barriers!"

Naïve Approach Does Not Work

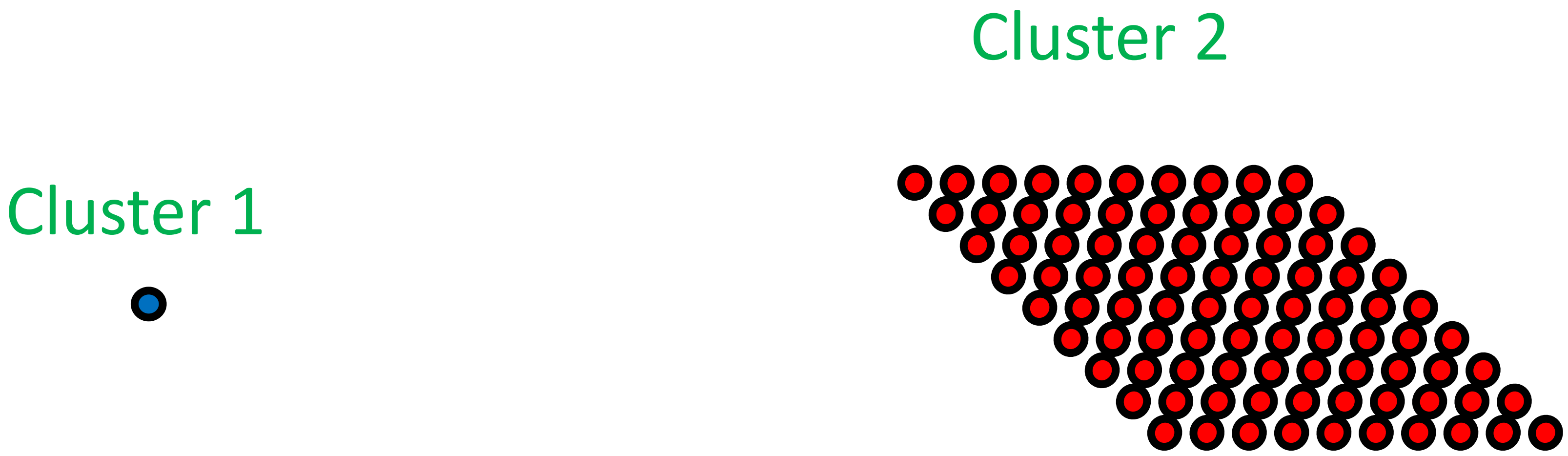
- Not enough to blindly follow predictions!



- Optimal cost ≈ 0
- Predictor with arbitrary small error has large cost!

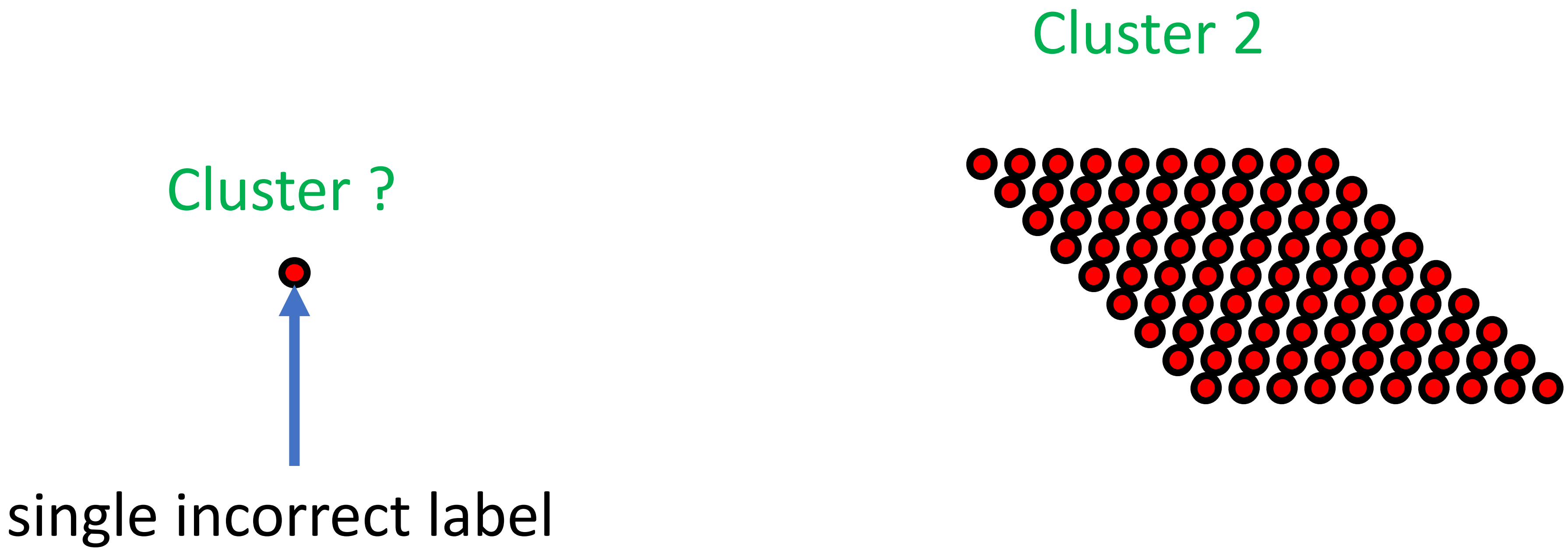
Naïve Approach Does Not Work

- Can a predictor even help?



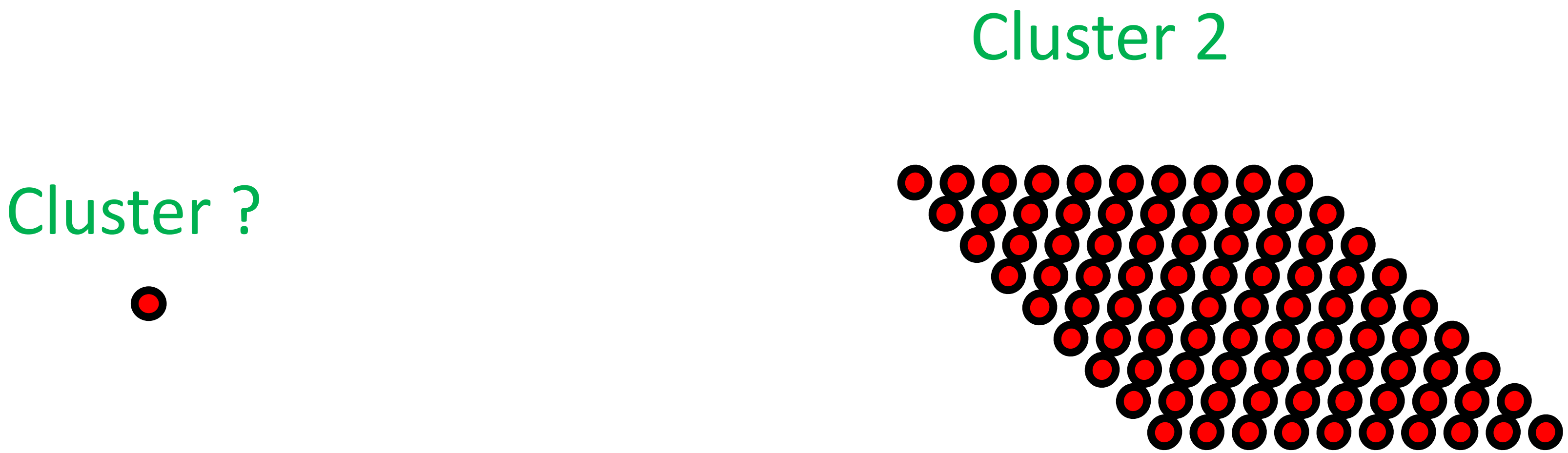
Naïve Approach Does Not Work

- Can a predictor even help?



Naïve Approach Does Not Work

- Can a predictor even help?



- MUST have assumptions about the accuracy on each cluster

Precision and Recall

- [EFSWZ22]: Assume cluster sizes are "balanced"

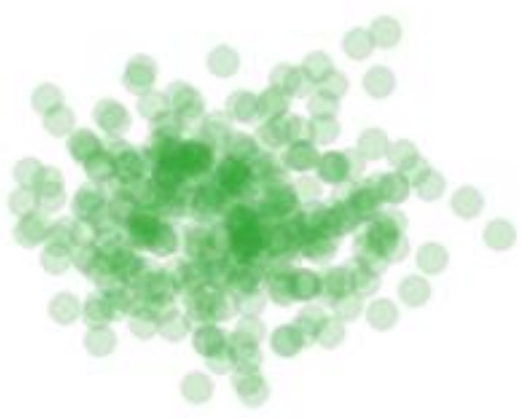
- [NCN23]: Let P_i be the optimal cluster with label i and Q_i be the points that are labeled i. Then $|Q_i \setminus P_i| + |P_i \setminus Q_i| \leq \alpha \cdot |P_i|$.

Precision

Recall

Algorithmic Intuition

- Our approach: Closed-form solution for best center of a *fixed* set of points



$$\operatorname{argmin}_c\,[\mathrm{cost}(c, P)] = \frac{1}{|P|}\sum_{p\in P} p$$

$$\operatorname{argmin}_c \sum_{p\in P} \|p - c\|_2^2 = \frac{1}{|P|}\sum_{p\in P} p$$

Algorithmic Intuition

- Consider each *dimension* separately

Algorithm 1 Learning-augmented k-means clustering

Input: A point set X with labels given by a predictor Π with label error rate λ

Output: $(1+O(\alpha))$-approximate k-means clustering of X

```
for i = 1 to i = k do
    Let Y_i be the set of points with label i.
    Run CrdEst for each of the d coordinates of Y_i.
    Let C'_i be the coordinate-wise outputs of CrdEst.
end for
Return C'_1, ..., C'_k.
```

Algorithmic Intuition

- Consider each *label* separately

Algorithm 1 Learning-augmented k-means clustering

Input: A point set X with labels given by a predictor Π with label error rate λ

Output: $(1+O(\alpha))$-approximate k-means clustering of X

for $i = 1$ to $i = k$ **do**

  Let Y_i be the set of points with label i.

  Run CrdEst for each of the d coordinates of Y_i.

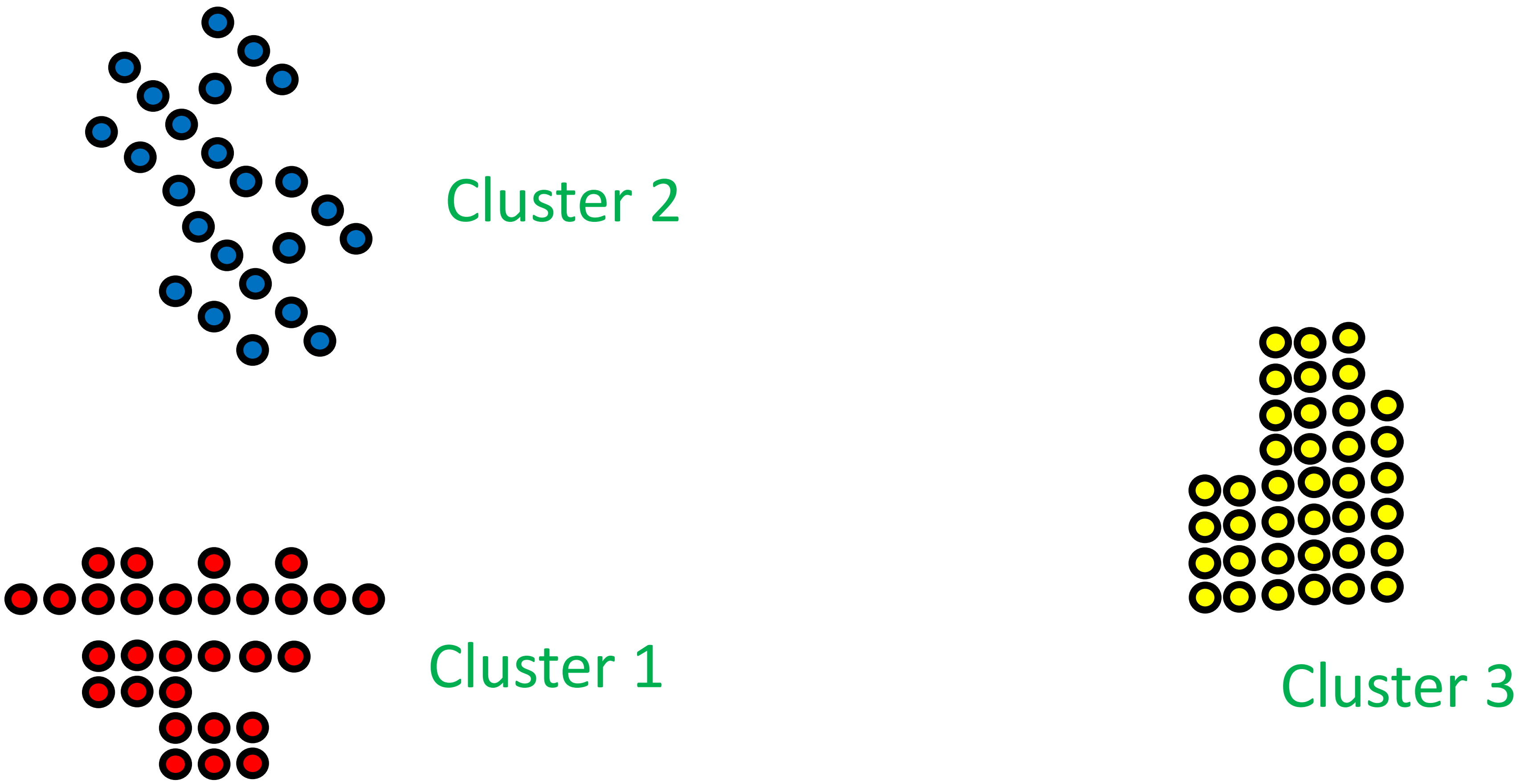
  Let C'_i be the coordinate-wise outputs of CrdEst.

end for

Return $C'_1, \ldots, C'_k$.

Algorithmic Intuition

- Example:

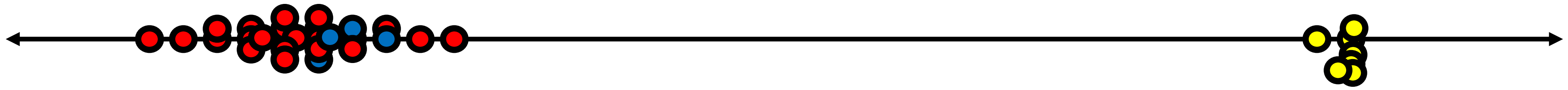


Algorithmic Intuition

- Example: Consider the points with predicted label 1

Algorithmic Intuition

- Example: Consider the points with predicted label 1
- Consider each dimension separately

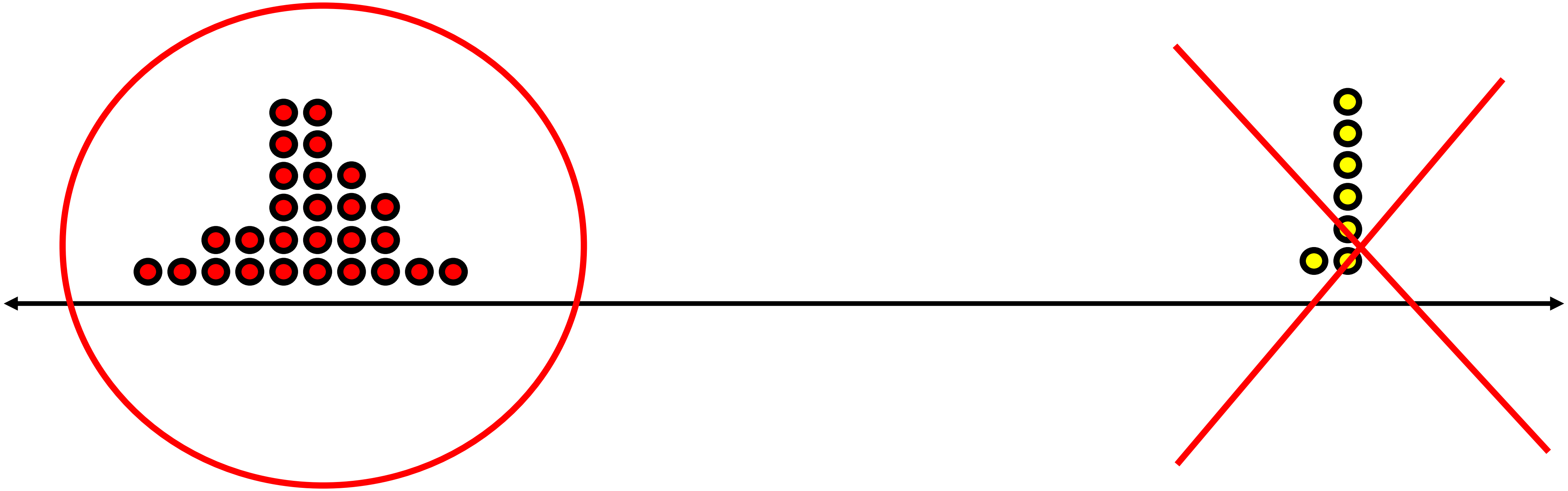


Algorithmic Intuition

- Example: Consider the histogram of points with predicted label 1

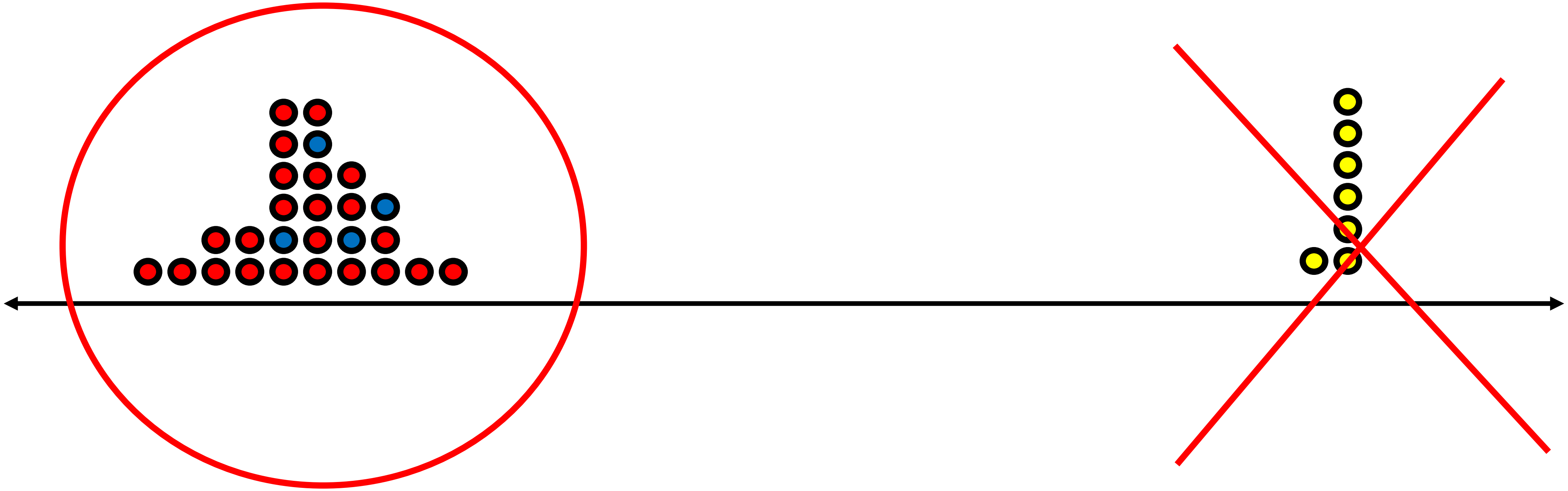
Algorithmic Intuition

- Example: Consider the histogram of points with predicted label 1
- Is it true that "pruning" away the outliers removes all incorrect points?



Algorithmic Intuition

- Is it true that “pruning” away the outliers removes all incorrect points? NO!



Algorithmic Intuition

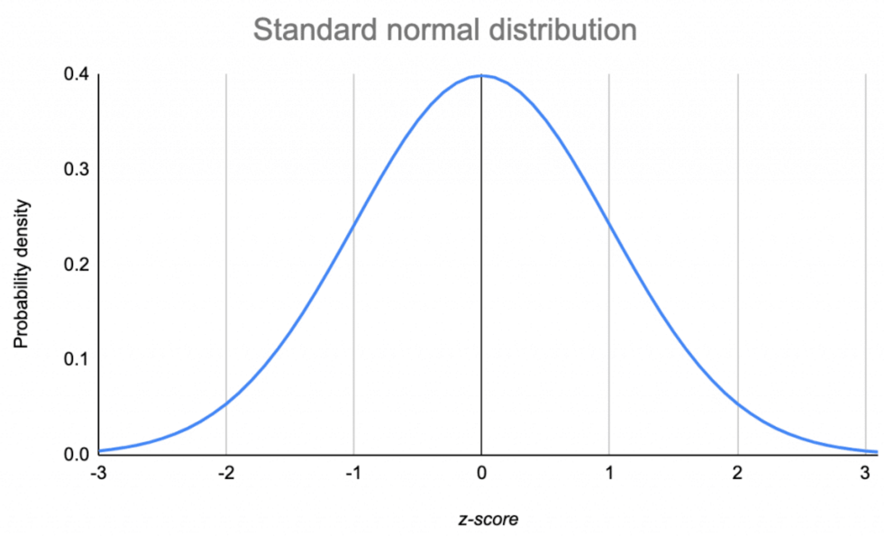
- Example: Consider the points with label 1

Algorithmic Intuition

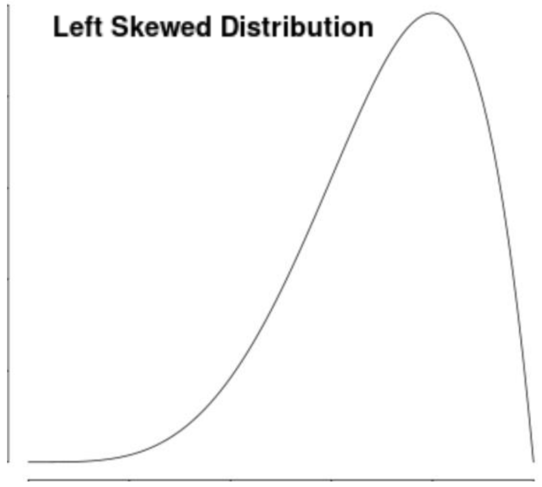
- Consider each label and each dimension separately

- Our approach: Use ideas from robust mean estimation

$(1-\alpha)P$



αQ



Algorithmic Intuition

- Case 1: Q is “far” from P

Algorithmic Intuition

- Case 1: Q is "far" from P

- Can detect handle this case by "pruning" the distribution

Algorithmic Intuition

- Case 2: Q is "close" to P

Algorithmic Intuition

- Case 2: Q is “close” to P

- Q cannot heavily affect the empirical mean P

Algorithmic Intuition

- Algorithm: Find the mean of the shortest interval that contains $(1 - O(\alpha))$ fraction of the points

Algorithm

- Algorithm: Find the mean of the shortest interval that contains $(1 - O(\alpha))$ fraction of the points

---

Algorithm 2 Coordinate-wise estimation CrdEst

Input: Points $x_1, \ldots, x_{2m} \in \mathbb{R}$, corruption level $\lambda \leq \alpha$

1: Randomly partition the points into two groups X_1, X_2 of size m.

2: Let $I = [a, b]$ be the shortest interval containing $m(1 - 5\alpha)$ points of X_1.

3: $Z \leftarrow X_2 \cap I$

4: $z \leftarrow \frac{1}{|Z|} \sum_{x \in Z} x$

5: **Return** z

Analysis Overview

- Robust mean estimation gives additive α error to the *location* of the mean

- How does this affect the k-means clustering cost?

Analysis Overview

- Analysis: Robust mean gives $(1+\alpha)$-approximation to the 1-means clustering cost
- Recall: Consider each label and each dimension separately

Analysis Overview

- Analysis: Robust mean gives $(1+\alpha)$-approximation to the k-means clustering cost

- Lemma: Let P, Q be sets of real numbers with $|P| \geq (1-\alpha)n$ and $|Q| \leq \alpha n$. Let $X = P \cup Q$, let C_X and C_P be the means of X and P. Then

$$\mathrm{Cost}(X, C_P) \leq (1+\alpha)\mathrm{Cost}(X, C_X)$$

- [InabaKatohIImai94]:

$$\mathrm{Cost}(X, C_P) \leq \mathrm{Cost}(X, C_X) + |X| \cdot |C_P - C_X|^2$$

Algorithm 1 Learning-augmented k-means clustering

Input: A point set X with labels given by a predictor Π with label error rate λ

Output: $(1+O(\alpha))$-approximate k-means clustering of X

```
for i = 1 to i = k do
    Let Y_i be the set of points with label i.
    Run CrdEst for each of the d coordinates of Y_i.
    Let C'_i be the coordinate-wise outputs of CrdEst.
end for
Return C'_1, ..., C'_k.
```

Algorithm 2 Coordinate-wise estimation CrdEst

Input: Points $x_1, \ldots, x_{2m} \in \mathbb{R}$, corruption level $\lambda \le \alpha$

```
Randomly partition the points into two groups X_1, X_2 of size m.
Let I = [a, b] be the shortest interval containing m(1 − 5α) points of X_1.
Z ← X_2 ∩ I
z ← (1/|Z|) Σ_{x∈Z} x
Return z
```

Algorithm [NCN23]

Algorithm 1 Deterministic Learning-augmented k-Means Clustering

Require: Data set P of m points, Partition $P = P_1 \cup \ldots P_k$ from a predictor, accuracy parameter α

for $i \in [k]$ **do**

  for $j \in [d]$ **do**

    Let $\omega_{i,j}$ be the collection of all subsets of $(1-\alpha)m_i$ contiguous points in $P_{i,j}$.

    $I_{i,j} \leftarrow \text{argmin}_{Z \in \omega_{i,j}} \text{cost}(Z, \overline{Z}) = \text{argmin}_{Z \in \omega_{i,j}} \sum_{z \in Z} z^2 - \frac{1}{|Z|} \left(\sum_{z' in Z} z'\right)^2$

  end for

  Let $\widehat{c}_i = (\overline{I_{i,j}})_{j \in [d]}$

end for

Return $\{\widehat{c}_1, \ldots, \widehat{c}_k\}$

"Find the best interval that contains $(1-\alpha)$ fraction of the points"

Additional Caveats

- Assignment of each point to clusters after finding centers in $\tilde{O}_{\varepsilon,\log k}(nd)$ time

- Dimensionality reduction to projects to lower-dimension space, use approximate nearest neighbors in lower-dimension space to assign points to clusters

Limitations

- Techniques specifically catered to k-means clustering (coordinate-wise decomposition, robust mean estimation)

- What about k-median clustering?

Algorithm [NCN23]

Algorithm 2 Learning-augmented k-Medians Clustering

Require: Data set P of m points, Partition $P = P_1 \cup \ldots P_k$ from a predictor, accuracy parameter $\alpha < 1/2$

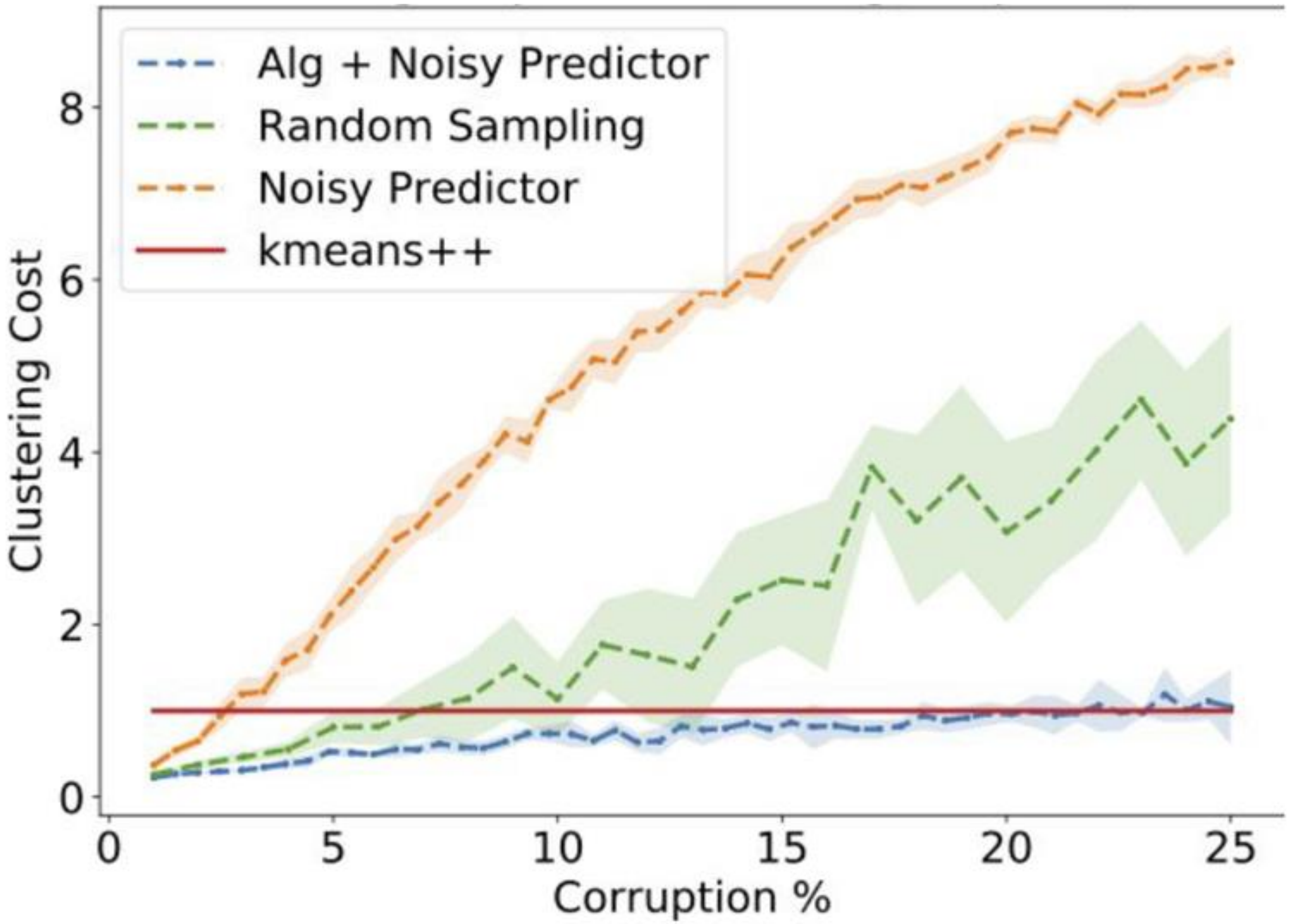
  for $i \in [k]$ **do**
    Let $R = \frac{2}{1-2\alpha} \log \frac{2k}{\delta}$
    for $j \in [R]$ **do**
      Sample $x \sim P_i$ u.a.r.
      Let P_i' be the $\lceil \alpha m_i \rceil$ points farthest from x
      $\widehat{c}_i^j \leftarrow$ median of $P_i \setminus P_i'$.
    end for
    Let $\widehat{c}_i$ be the $\widehat{c}_i^j$ with minimum cost
  end for
  Return $\{\hat{c}_1, \ldots, \hat{c}_k\}$

"Sample, prune, and find the geometric median, e.g., [CLMPS16]"

Experimental Results

- Case Study: Spectral clustering on graphs varying over time
- Dataset: Internet router graph varying over the course of a year
- Methodology: Compare to standard benchmarks while using various natural predictors, i.e., noisily perturb true labels and compare to baselines as function of error

Dataset: Internet router graph varying over the course of a year, $k = 10$



Conclusion: Our algorithm (using predictor) outperforms benchmarks such as k-means ++ for low error while staying competitive with high corruptions

Summary

- NP-hard to even approximate within a factor of 1.07 [Cohen-AddadC.S.20, LeeSchmidtWright17]
- Main result [EFSWZ22]: Algorithm that outputs a $(1 + O(\alpha))$ approximate k-means clustering in nearly linear time
- Handles clustering with *outliers*
- Not enough to blindly follow predictions!
- Our approach: Use ideas from robust mean estimation

…and Beyond!

- Related work:
- Semi-supervised active clustering (SSAC) framework: Same cluster queries, [AKB16], [KG17], [MS17], [GHS18], [ABJK18], …, correlation clustering
- Future directions:
- Spectral clustering? (Talk to me!!)
- Other predictors (multiple labels per point), relationship with robust statistics, minimizing the number of queries
- Algorithms for (k, z)-clustering, i.e., $\sum_{p\in P} \min_{c\in C} \|p - c\|_2^z$
- Algorithms for L_p-metrics, i.e., $\sum_{p\in P} \min_{c\in C} \|p - c\|_p^p$