Data-driven Clustering via Parameterized Lloyds Families

Travis Dick
Joint work with Maria-Florina Balcan and Colin White
Carnegie Mellon University
NeurIPS 2018

Data-driven Clustering

Data-driven Clustering

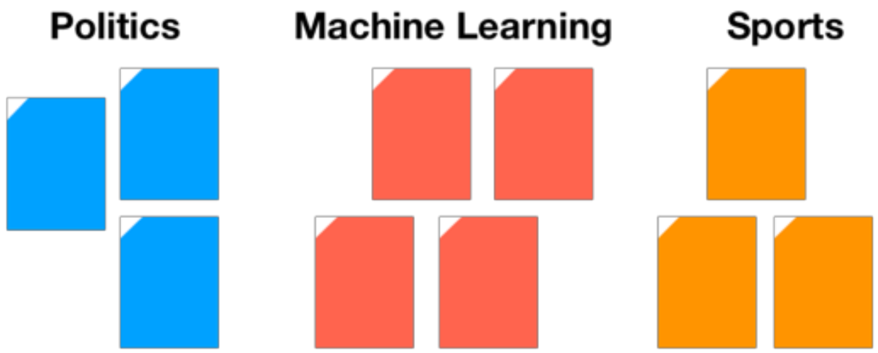
- Clustering aims to divide a dataset into self-similar clusters.

Data-driven Clustering

- Clustering aims to divide a dataset into self-similar clusters.
- Goal: find some unknown natural clustering.

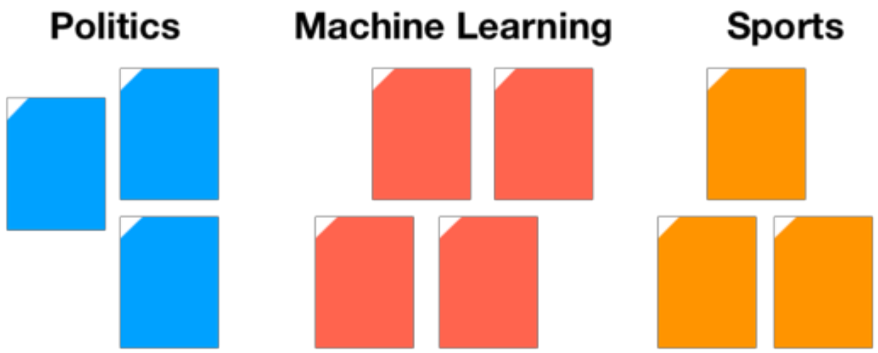
Data-driven Clustering

- Clustering aims to divide a dataset into self-similar clusters.
- Goal: find some unknown natural clustering.



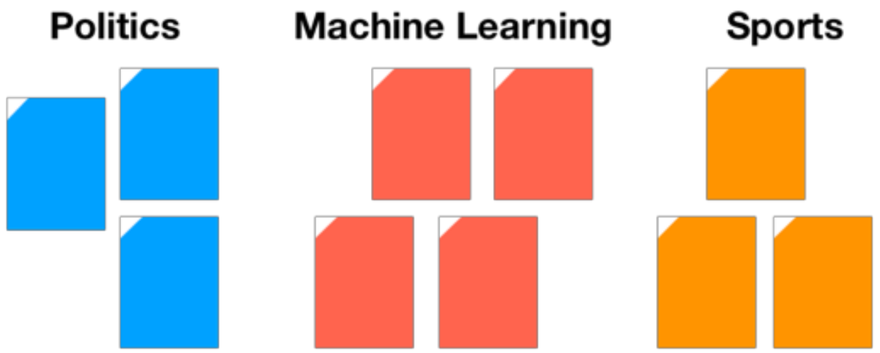
Data-driven Clustering

- Clustering aims to divide a dataset into self-similar clusters.
- Goal: find some unknown natural clustering.



Data-driven Clustering

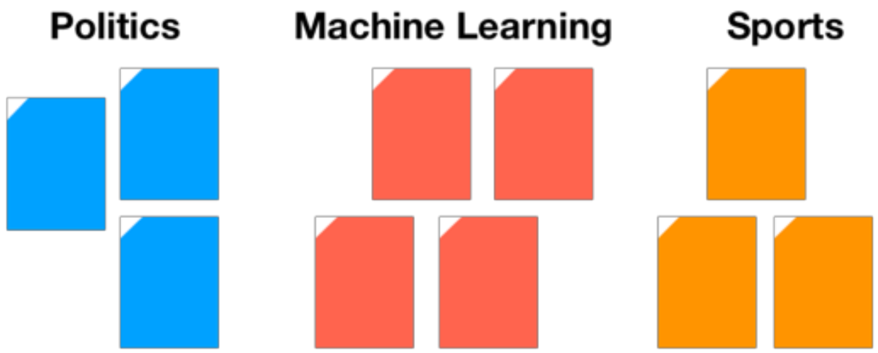
- Clustering aims to divide a dataset into self-similar clusters.
- Goal: find some unknown natural clustering.



- However, most clustering algorithms minimize a clustering cost function.

Data-driven Clustering

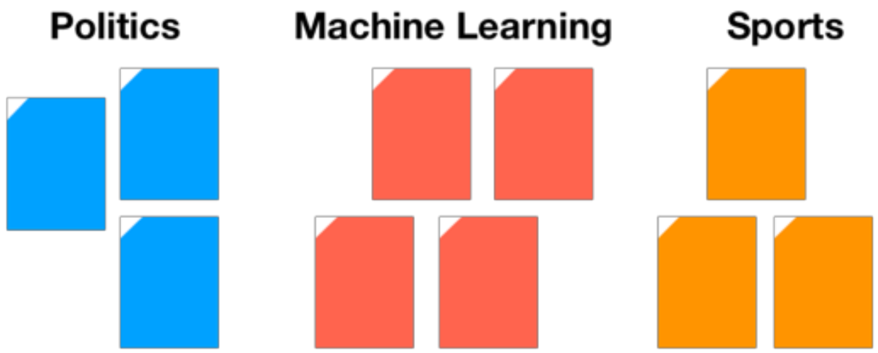
- Clustering aims to divide a dataset into self-similar clusters.
- Goal: find some unknown natural clustering.



- However, most clustering algorithms minimize a clustering cost function.
- Hope that low-cost clusterings recover the natural clusters.

Data-driven Clustering

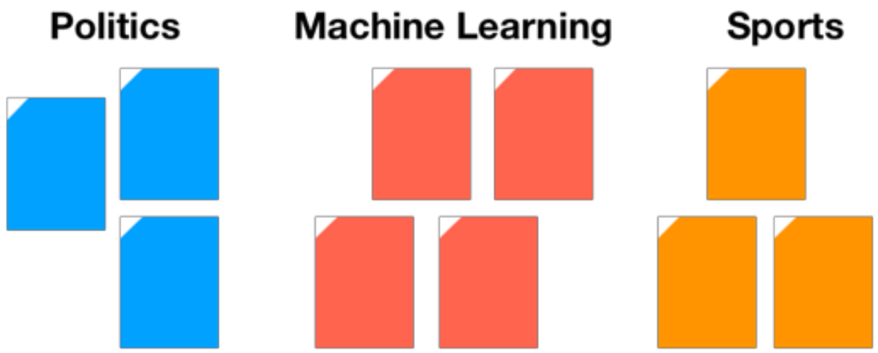
- Clustering aims to divide a dataset into self-similar clusters.
- Goal: find some unknown natural clustering.



- However, most clustering algorithms minimize a clustering cost function.
- Hope that low-cost clusterings recover the natural clusters.
- There are many algorithms and many objectives.

Data-driven Clustering

- Clustering aims to divide a dataset into self-similar clusters.
- Goal: find some unknown natural clustering.

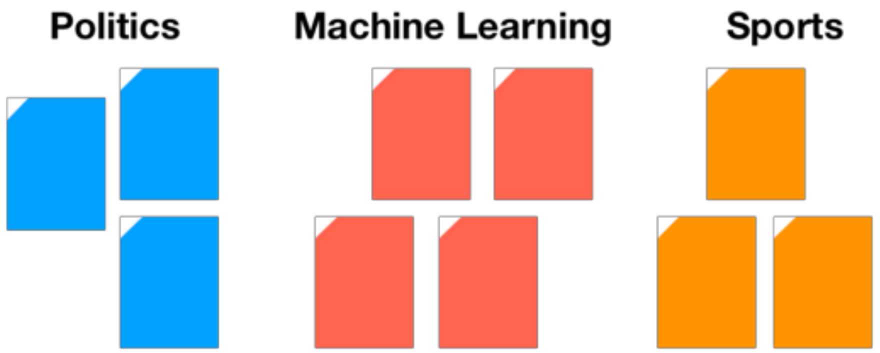


- However, most clustering algorithms minimize a clustering cost function.
- Hope that low-cost clusterings recover the natural clusters.
- There are many algorithms and many objectives.

How do we choose the best algorithm for a specific application?

Data-driven Clustering

- Clustering aims to divide a dataset into self-similar clusters.
- Goal: find some unknown natural clustering.



- However, most clustering algorithms minimize a clustering cost function.
- Hope that low-cost clusterings recover the natural clusters.
- There are many algorithms and many objectives.

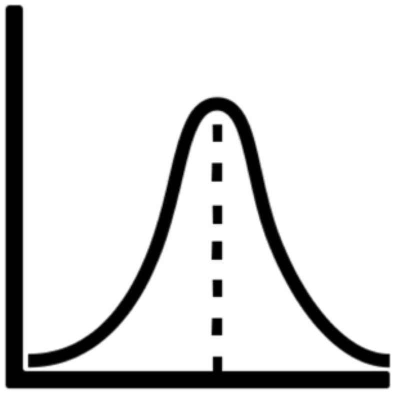
How do we choose the best algorithm for a specific application?

Can we automate this process?

Learning Model

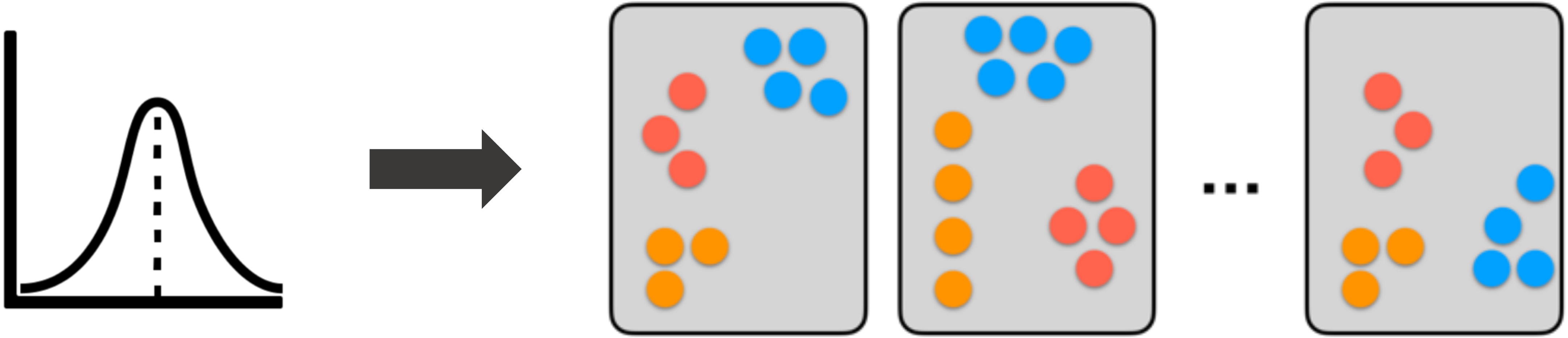
Learning Model

- An unknown distribution $\mathcal{P}$ over clustering instances.



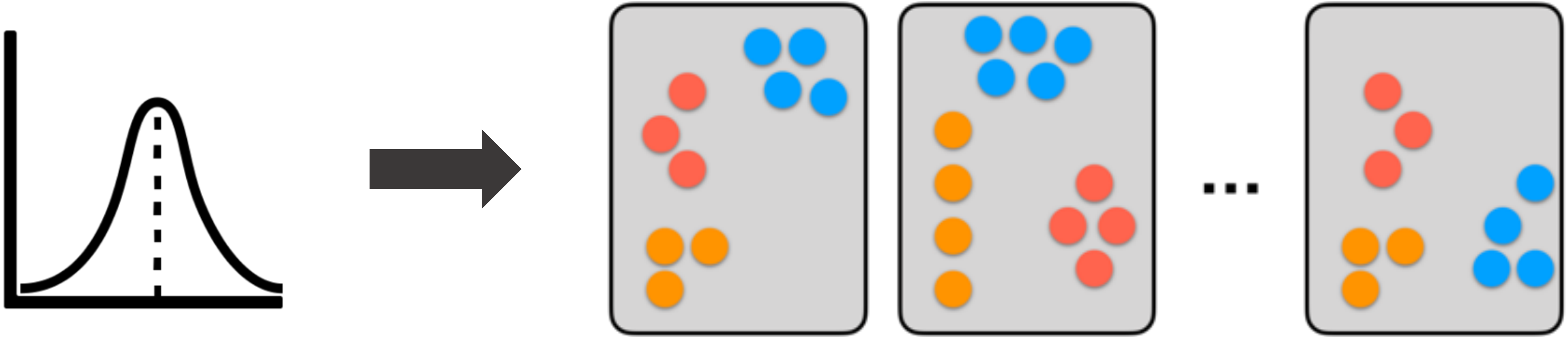
Learning Model

- An unknown distribution $\mathcal{P}$ over clustering instances.
- Given a sample $x_1, \dots, x_n \sim \mathcal{P}$ annotated by their target clusterings.



Learning Model

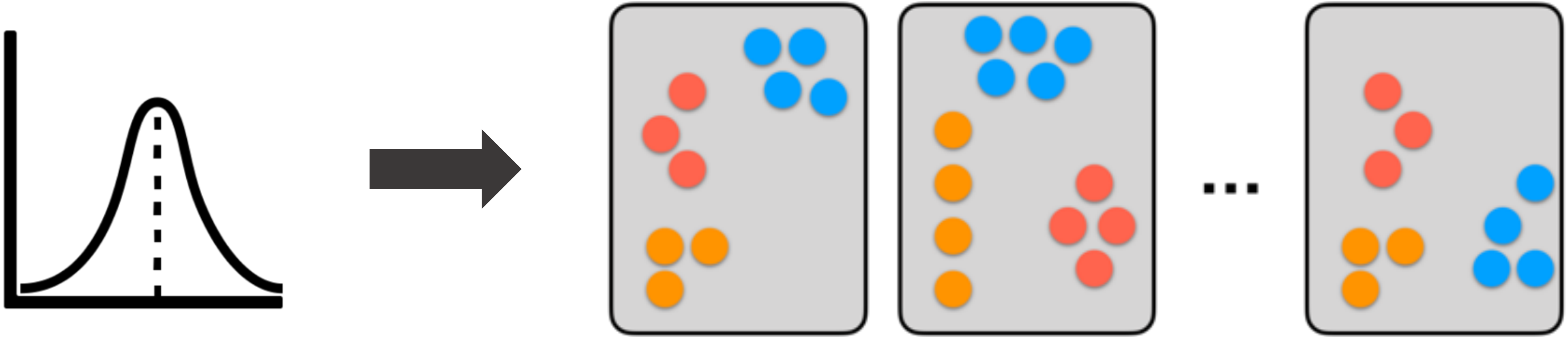
- An unknown distribution $\mathcal{P}$ over clustering instances.
- Given a sample $x_1, \dots, x_n \sim \mathcal{P}$ annotated by their target clusterings.



- Find an algorithm $\mathcal{A}$ that produces clusterings similar to the target clusterings.

Learning Model

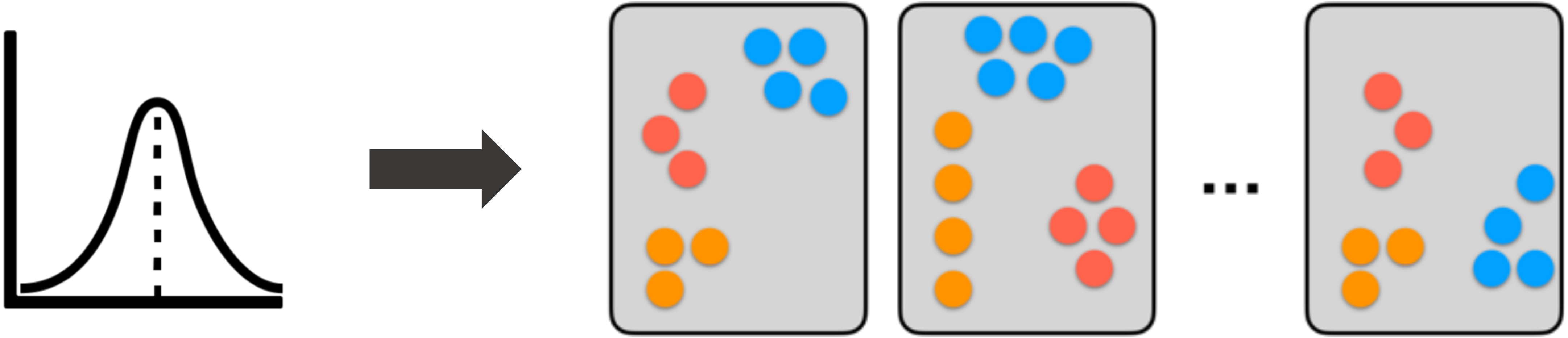
- An unknown distribution $\mathcal{P}$ over clustering instances.
- Given a sample $x_1, \ldots, x_n \sim \mathcal{P}$ annotated by their target clusterings.



- Find an algorithm $\mathcal{A}$ that produces clusterings similar to the target clusterings.
- Want $\mathcal{A}$ to also work well for new instances from $\mathcal{P}$!

Learning Model

- An unknown distribution $\mathcal{P}$ over clustering instances.
- Given a sample $x_1, \dots, x_n \sim \mathcal{P}$ annotated by their target clusterings.

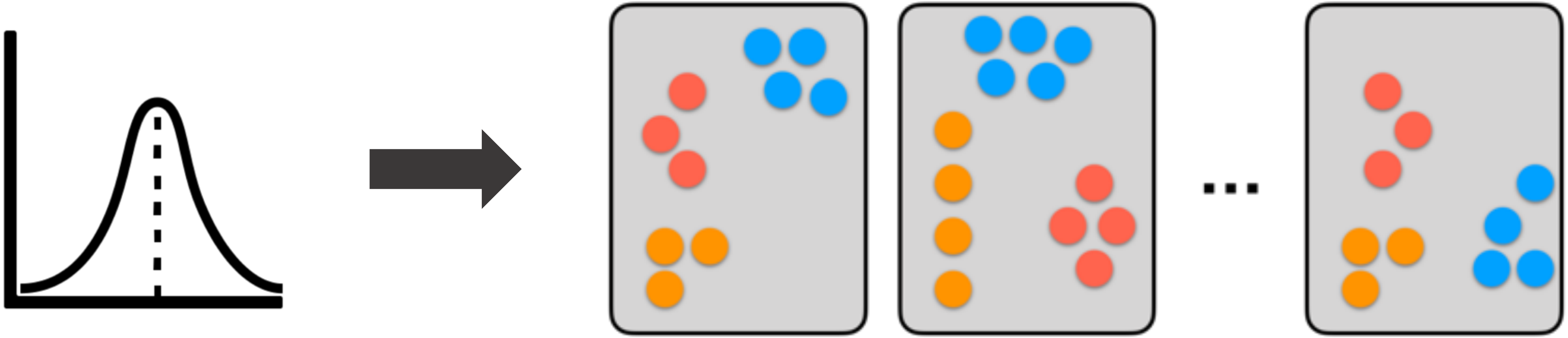


- Find an algorithm $\mathcal{A}$ that produces clusterings similar to the target clusterings.
- Want $\mathcal{A}$ to also work well for new instances from $\mathcal{P}$!

- In this work:
  1. Introduce large parametric family of clustering algorithms, (α, β)-Lloyds.

Learning Model

- An unknown distribution $\mathcal{P}$ over clustering instances.
- Given a sample $x_1, \ldots, x_n \sim \mathcal{P}$ annotated by their target clusterings.

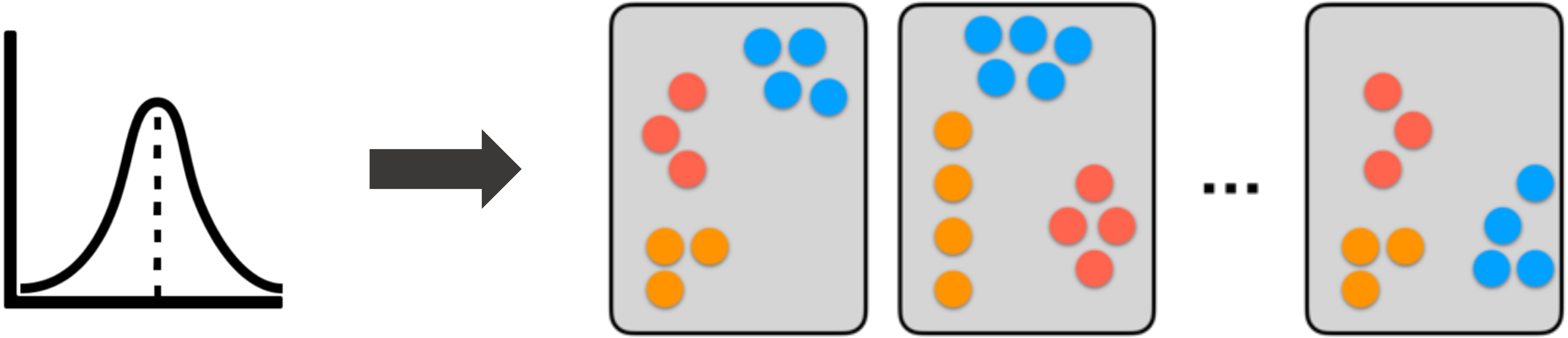


- Find an algorithm $\mathcal{A}$ that produces clusterings similar to the target clusterings.
- Want $\mathcal{A}$ to also work well for new instances from $\mathcal{P}$!

- In this work:
  1. Introduce large parametric family of clustering algorithms, (α, β)-Lloyds.
  2. Efficient procedures for finding best parameters on a sample.

Learning Model

- An unknown distribution $\mathcal{P}$ over clustering instances.
- Given a sample $x_1, \dots, x_n \sim \mathcal{P}$ annotated by their target clusterings.



- Find an algorithm $\mathcal{A}$ that produces clusterings similar to the target clusterings.
- Want $\mathcal{A}$ to also work well for new instances from $\mathcal{P}$!

- In this work:
  1. Introduce large parametric family of clustering algorithms, (α, β)-Lloyds.
  2. Efficient procedures for finding best parameters on a sample.
  3. Generalization: optimal parameters on sample are nearly optimal on $\mathcal{P}$.

Lloyds Method

Lloyds Method

- Maintains k centers $c_1, \ldots, c_k$ that define clusters.

Lloyds Method

- Maintains k centers $c_1, \ldots, c_k$ that define clusters.
- Perform local search to improve the k-means cost of the centers.

Lloyds Method

- Maintains k centers $c_1, \dots, c_k$ that define clusters.
- Perform local search to improve the k-means cost of the centers.
    1. Assign each point to nearest center.

Lloyds Method

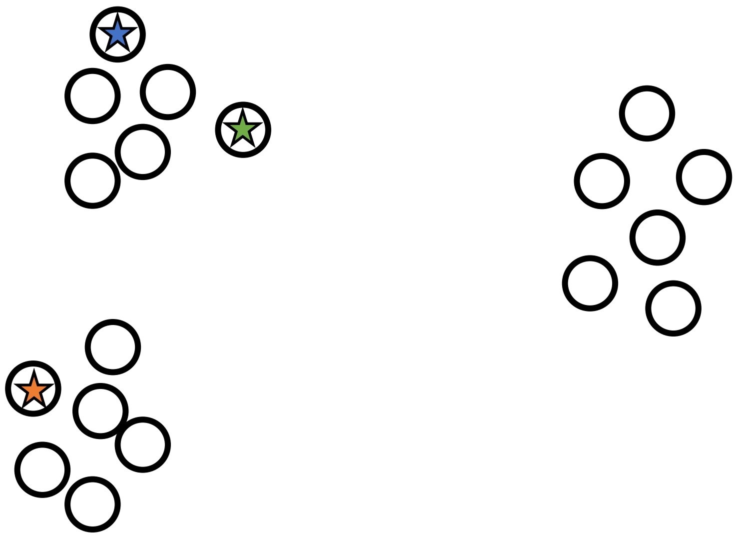
- Maintains k centers $c_1, \dots, c_k$ that define clusters.
- Perform local search to improve the k-means cost of the centers.
  1. Assign each point to nearest center.
  2. Update each center to be the mean of assigned points.

Lloyds Method

- Maintains k centers $c_1, \dots, c_k$ that define clusters.
- Perform local search to improve the k-means cost of the centers.
  1. Assign each point to nearest center.
  2. Update each center to be the mean of assigned points.
  3. Repeat until convergence.

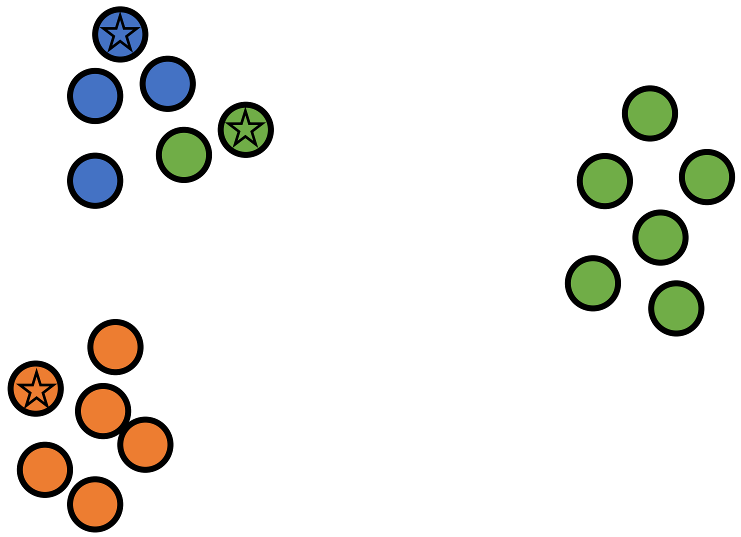
Lloyds Method

- Maintains k centers $c_1, \ldots, c_k$ that define clusters.
- Perform local search to improve the k-means cost of the centers.
  1. Assign each point to nearest center.
  2. Update each center to be the mean of assigned points.
  3. Repeat until convergence.



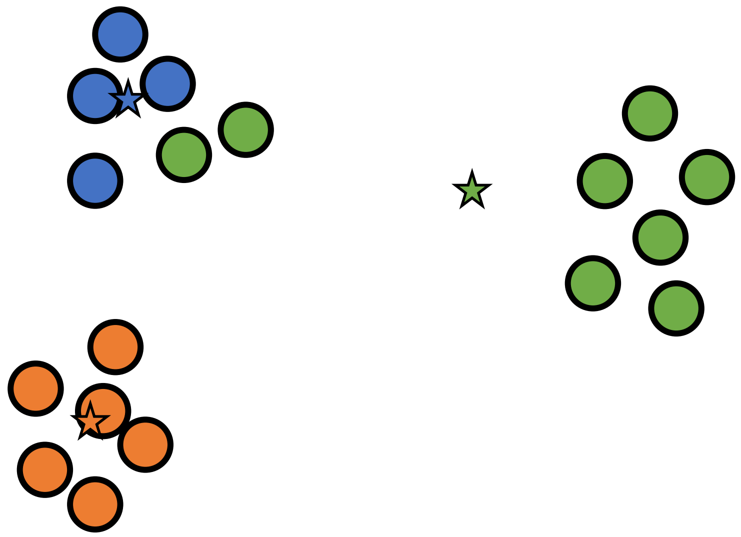
Lloyds Method

- Maintains k centers $c_1, \dots, c_k$ that define clusters.
- Perform local search to improve the k-means cost of the centers.
    1. Assign each point to nearest center.
    2. Update each center to be the mean of assigned points.
    3. Repeat until convergence.



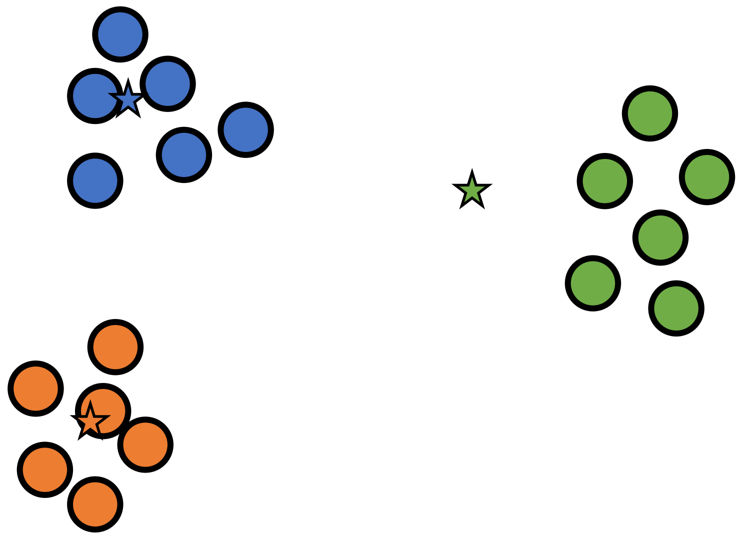
Lloyds Method

- Maintains k centers $c_1, \dots, c_k$ that define clusters.
- Perform local search to improve the k-means cost of the centers.
  1. Assign each point to nearest center.
  2. Update each center to be the mean of assigned points.
  3. Repeat until convergence.



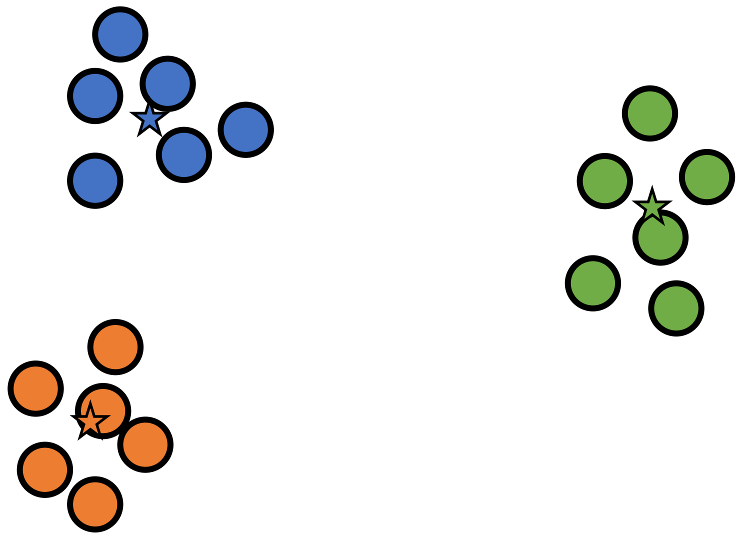
Lloyds Method

- Maintains k centers $c_1, \dots, c_k$ that define clusters.
- Perform local search to improve the k-means cost of the centers.
  1. Assign each point to nearest center.
  2. Update each center to be the mean of assigned points.
  3. Repeat until convergence.



Lloyds Method

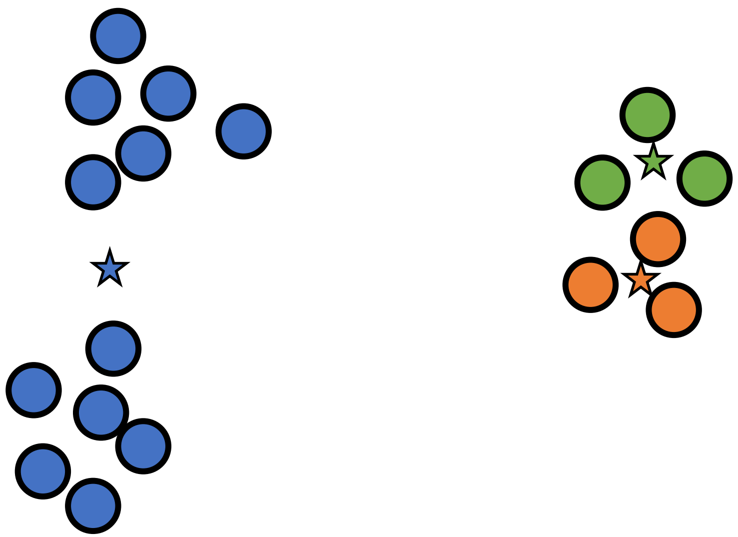
- Maintains k centers $c_1, \dots, c_k$ that define clusters.
- Perform local search to improve the k-means cost of the centers.
  1. Assign each point to nearest center.
  2. Update each center to be the mean of assigned points.
  3. Repeat until convergence.



Initial Centers are Important!

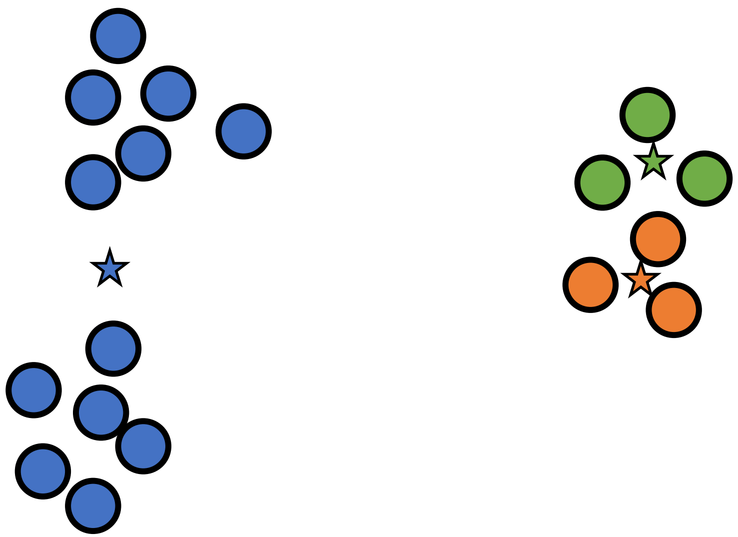
Initial Centers are Important!

- Lloyd's method can get stuck if initial centers are chosen poorly



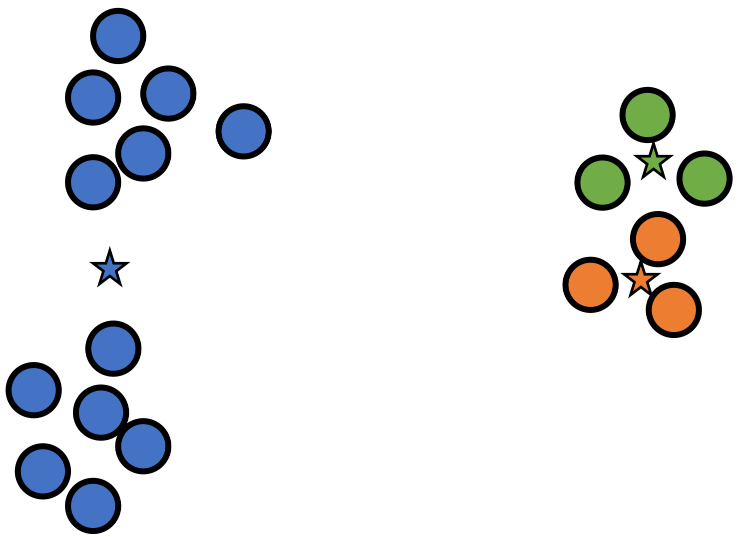
Initial Centers are Important!

- Lloyd's method can get stuck if initial centers are chosen poorly
- Initialization is a well-studied problem with many proposed procedures (e.g., k-means++)



Initial Centers are Important!

- Lloyd's method can get stuck if initial centers are chosen poorly
- Initialization is a well-studied problem with many proposed procedures (e.g., k-means++)
- Best method will depend on properties of the clustering instances.



The (α, β)-Lloyds Family

The (α, β)-Lloyds Family

Initialization: Parameter α

The (α, β)-Lloyds Family

Initialization: Parameter α

- Use d^{α}-sampling (generalizing d^{2}-sampling of k-means++)

The (α, β)-Lloyds Family

Initialization: Parameter α

- Use d^{α}-sampling (generalizing d^2-sampling of k-means++)
- Choose initial centers from dataset S randomly.

The (α, β)-Lloyds Family

Initialization: Parameter α

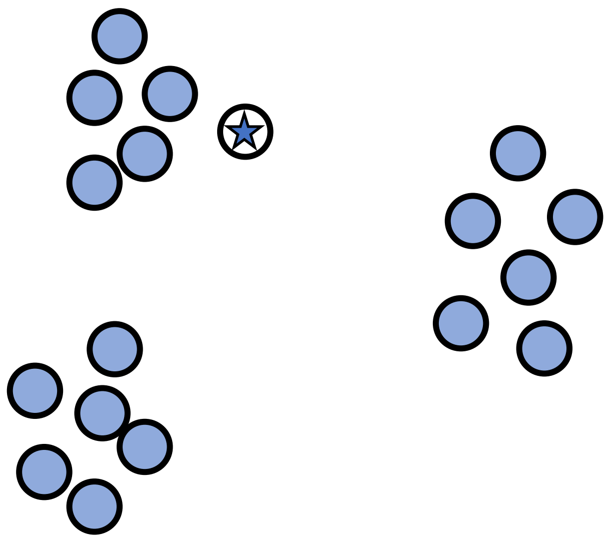
- Use d^α-sampling (generalizing d^2-sampling of k-means++)
- Choose initial centers from dataset S randomly.
- Probability that point $x \in S$ is center c_i is proportional to $d(x, \{c_1, \dots, c_{i-1}\})^\alpha$.

The (α, β)-Lloyds Family

Initialization: Parameter α

- Use d^{α}-sampling (generalizing d^{2}-sampling of k-means++)
- Choose initial centers from dataset S randomly.
- Probability that point $x \in S$ is center c_i is proportional to $d(x, \{c_1, \dots, c_{i-1}\})^{\alpha}$.

$\alpha = 0$: random initialization



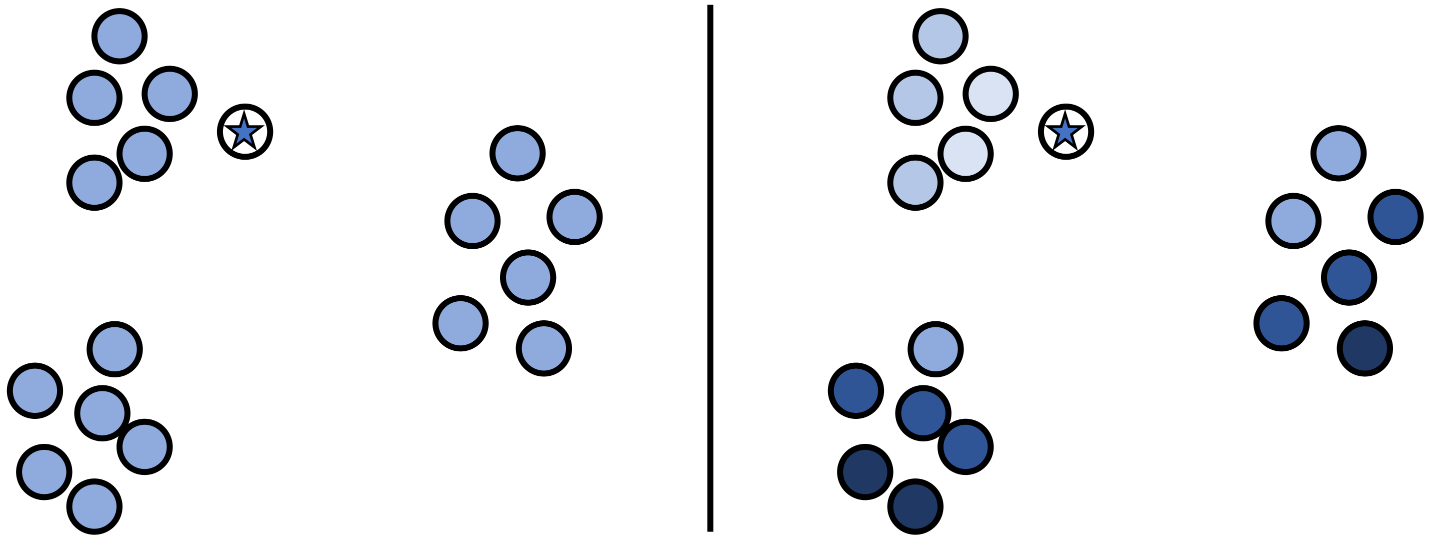
The (α, β)-Lloyds Family

Initialization: Parameter α

- Use d^α-sampling (generalizing d^2-sampling of k-means++)
- Choose initial centers from dataset S randomly.
- Probability that point $x \in S$ is center c_i is proportional to $d(x, \{c_1, \ldots, c_{i-1}\})^\alpha$.

$\alpha = 0$: random initialization

$\alpha = 2$: k-means++



The (α, β)-Lloyds Family

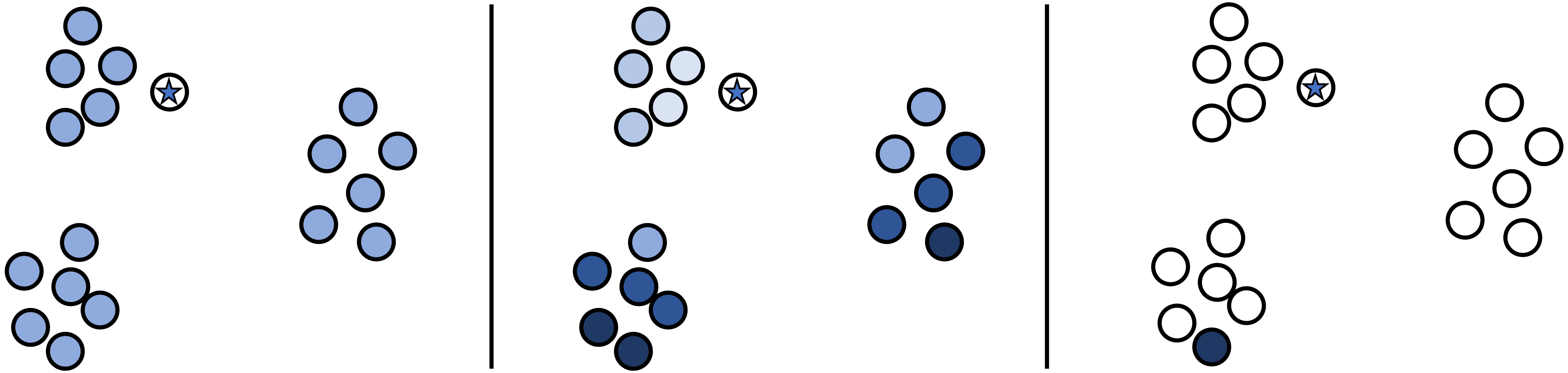
Initialization: Parameter α

- Use d^{α}-sampling (generalizing d^2-sampling of k-means++)
- Choose initial centers from dataset S randomly.
- Probability that point $x \in S$ is center c_i is proportional to $d(x, \{c_1, \ldots, c_{i-1}\})^{\alpha}$.

$\alpha = 0$: random initialization

$\alpha = 2$: k-means++

$\alpha = \infty$: farthest first

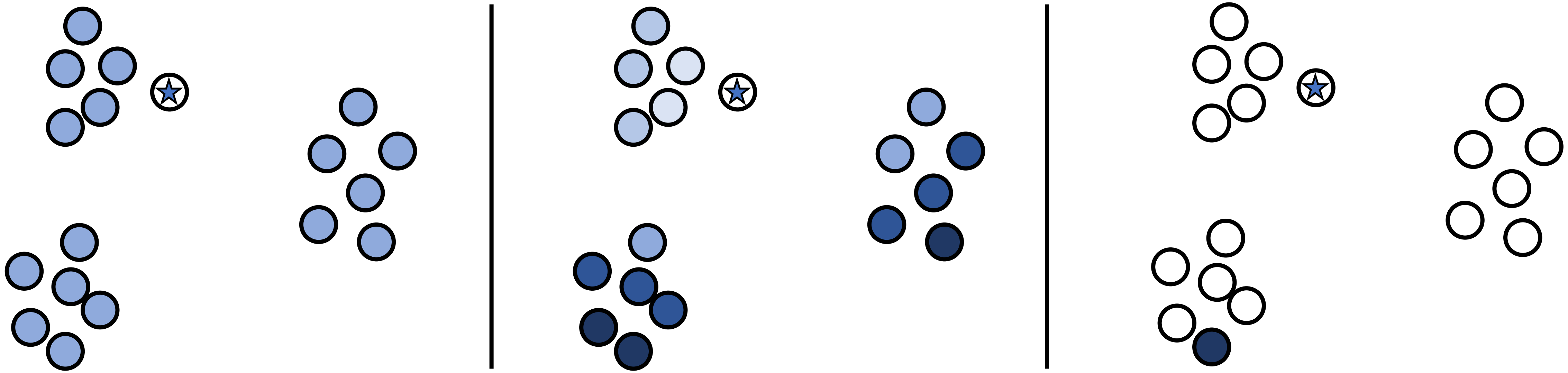


The (α, β)-Lloyds Family

Initialization: Parameter α

- Use d^α-sampling (generalizing d^2-sampling of k-means++)
- Choose initial centers from dataset S randomly.
- Probability that point $x \in S$ is center c_i is proportional to $d(x, \{c_1, \ldots, c_{i-1}\})^\alpha$.

$\alpha = 0$: random initialization $\alpha = 2$: k-means++ $\alpha = \infty$: farthest first



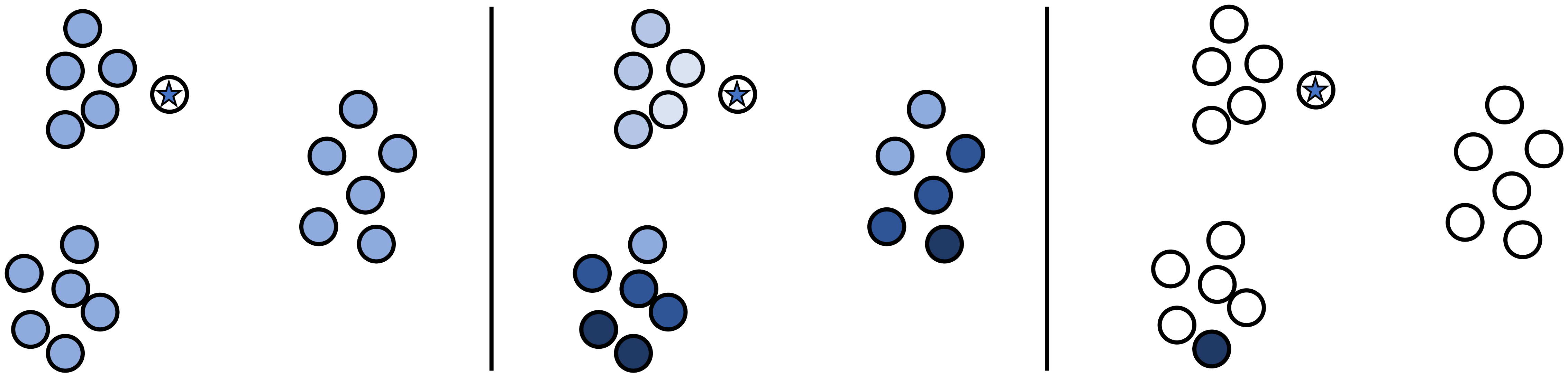
Local search: Second parameter β tweaks the local search. Details in paper.

The (α, β)-Lloyds Family

Initialization: Parameter α

- Use d^α-sampling (generalizing d^2-sampling of k-means++)
- Choose initial centers from dataset S randomly.
- Probability that point $x \in S$ is center c_i is proportional to $d(x, \{c_1, \ldots, c_{i-1}\})^\alpha$.

$\alpha = 0$: random initialization $\alpha = 2$: k-means++ $\alpha = \infty$: farthest first



Local search: Second parameter β tweaks the local search. Details in paper.

Question: For a distribution $\mathcal{P}$ over tasks, what parameters give best performance?

Results

Results

Efficient Tuning on Sample:

Results

Efficient Tuning on Sample:

- Efficient algorithm for finding parameters on sample with best agreement to targets.

Results

Efficient Tuning on Sample:

- Efficient algorithm for finding parameters on sample with best agreement to targets.
- “Algorithmically feasible to tune parameters on sample.”

Results

Efficient Tuning on Sample:

- Efficient algorithm for finding parameters on sample with best agreement to targets.
- "Algorithmically feasible to tune parameters on sample."

Generalization Guarantee:

Results

Efficient Tuning on Sample:

- Efficient algorithm for finding parameters on sample with best agreement to targets.
- "Algorithmically feasible to tune parameters on sample."

Generalization Guarantee:

- Analyze the intrinsic complexity of (α, β)-Lloyds

Results

Efficient Tuning on Sample:

- Efficient algorithm for finding parameters on sample with best agreement to targets.
- "Algorithmically feasible to tune parameters on sample."

Generalization Guarantee:

- Analyze the intrinsic complexity of (α, β)-Lloyds
- Show that need only roughly $\tilde{O}\left(\frac{k \log n}{\epsilon^2}\right)$ clustering instances to ensure empirical cost for all parameters within ϵ of expected cost.

Results

Efficient Tuning on Sample:

- Efficient algorithm for finding parameters on sample with best agreement to targets.
- "Algorithmically feasible to tune parameters on sample."

Generalization Guarantee:

- Analyze the intrinsic complexity of (α, β)-Lloyds
- Show that need only roughly $\tilde{O}\left(\frac{k \log n}{\epsilon^2}\right)$ clustering instances to ensure empirical cost for all parameters within ϵ of expected cost.
- "Parameters tuned on the sample will work well for new instances!"

Results

Efficient Tuning on Sample:

- Efficient algorithm for finding parameters on sample with best agreement to targets.
- “Algorithmically feasible to tune parameters on sample.”

Generalization Guarantee:

- Analyze the intrinsic complexity of (α, β)-Lloyds
- Show that need only roughly $\tilde{O}\left(\frac{k \log n}{\epsilon^2}\right)$ clustering instances to ensure empirical cost for all parameters within ϵ of expected cost.
- “Parameters tuned on the sample will work well for new instances!”

Experiments: Evaluate (α, β)-Lloyds family on real and synthetic data.