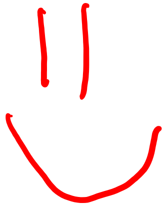
Announcements



Assignments

- HW9
  - Out Friday
  - Due Wed, 12/9, 11:59 pm
  - The two slip days are free (last possible submission Fri, 12/11, 11:59 pm)

Final Exam

- Mon, 12/14
- Stay tuned to Piazza for more details

Introduction to Machine Learning

Clustering

Instructor: Pat Virtue

Learning Paradigms

Paradigm	**Data**
Supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$ $\quad$ $\mathbf{x} \sim p^*(\cdot)$ and $y = c^*(\cdot)$
$\hookrightarrow$ Regression	$y^{(i)} \in \mathbb{R}$
$\hookrightarrow$ Classification	$y^{(i)} \in \{1, \ldots, K\}$
$\hookrightarrow$ Binary classification	$y^{(i)} \in \{+1, -1\}$
$\hookrightarrow$ Structured Prediction	$\mathbf{y}^{(i)}$ is a vector
Unsupervised	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ $\quad$ $\mathbf{x} \sim p^*(\cdot)$
Semi-supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N_1} \cup \{\mathbf{x}^{(j)}\}_{j=1}^{N_2}$
Online	$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), (\mathbf{x}^{(3)}, y^{(3)}), \ldots\}$
Active Learning	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ and can query $y^{(i)} = c^*(\cdot)$ at a cost
Imitation Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}), (s^{(2)}, a^{(2)}), \ldots\}$
Reinforcement Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}, r^{(1)}), (s^{(2)}, a^{(2)}, r^{(2)}), \ldots\}$

Slide credit: CMU MLD, Matt Gormley

Clustering, Informal Goals

Goal: Automatically partition unlabeled data into groups of similar datapoints.

Question: When and why would we want to do this?

Useful for:

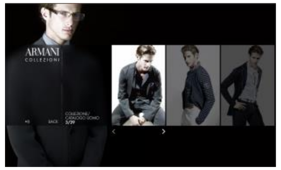
- Automatically organizing data.
- Understanding hidden structure in data.
- Preprocessing for further analysis.
    - Representing high-dimensional data in a low-dimensional space (e.g., for visualization purposes).

Slide credit: CMU MLD Nina Balcan

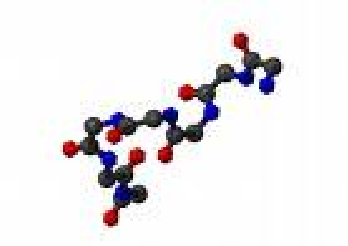
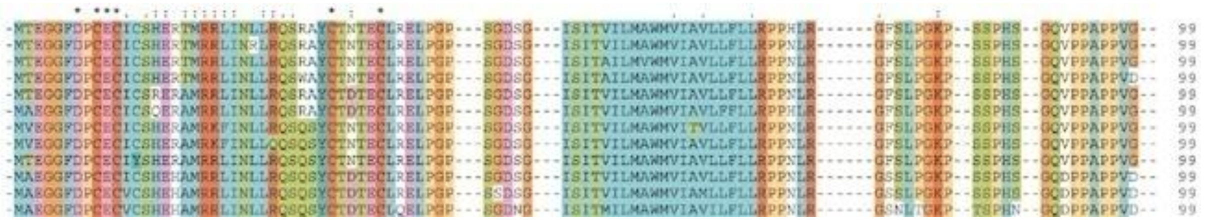
Applications (Clustering comes up everywhere…)

Cluster news articles or web pages or search results by topic.

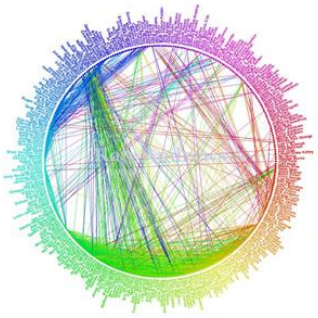




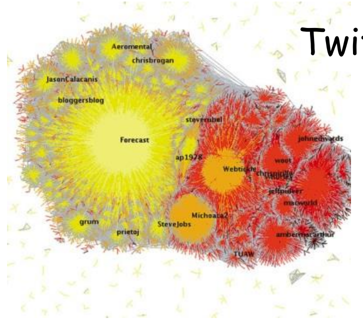
- Cluster protein sequences by function or genes according to expression profile.

- Cluster users of social networks by interest (community detection).



Facebook network

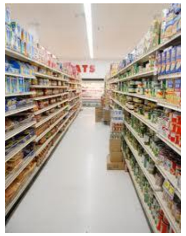


Twitter Network

Slide credit: CMU MLD Nina Balcan

Applications (Clustering comes up everywhere…)

Cluster customers according to purchase history.



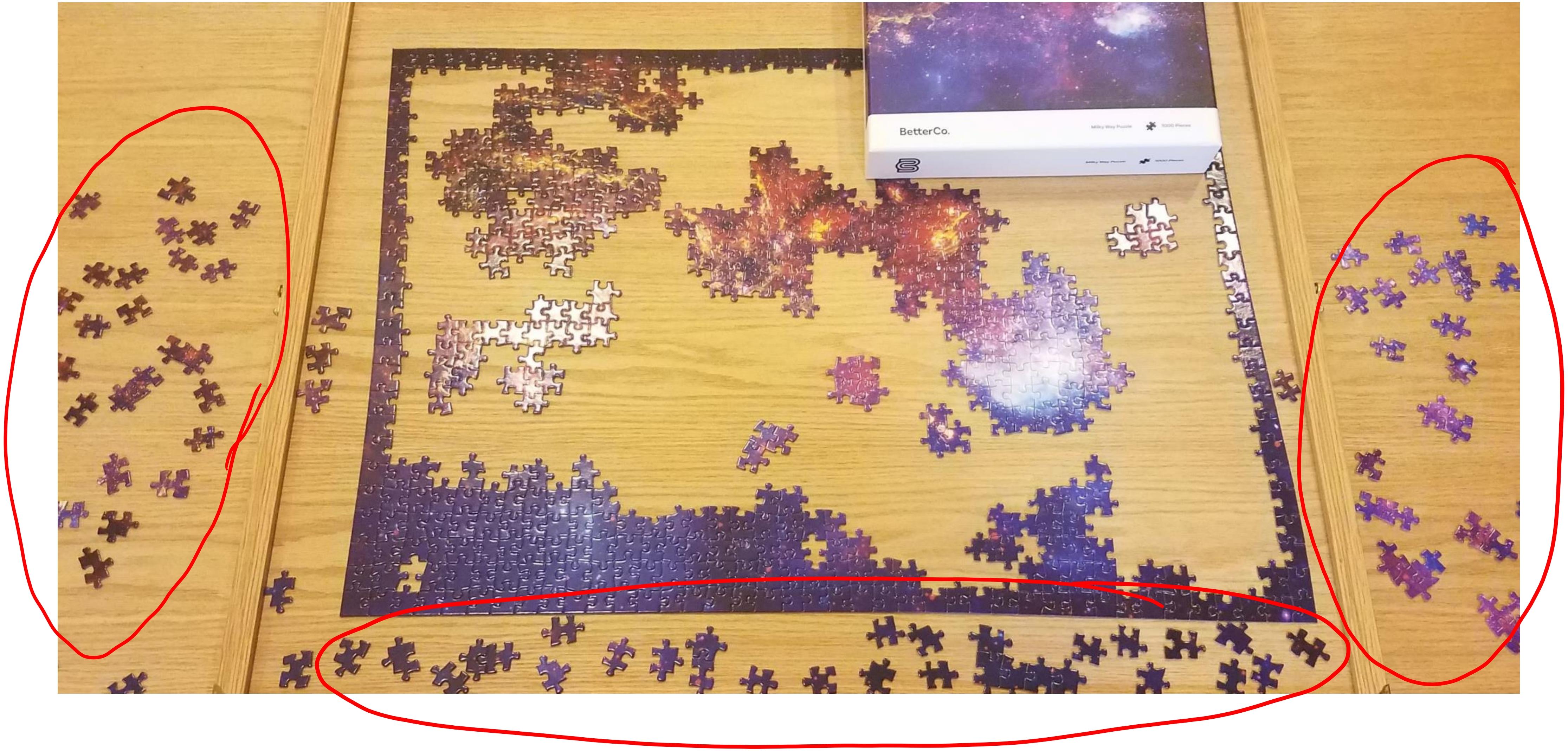
- Cluster galaxies or nearby stars (e.g. Sloan Digital Sky Survey)

- And many many more applications….

Slide credit: CMU MLD Nina Balcan

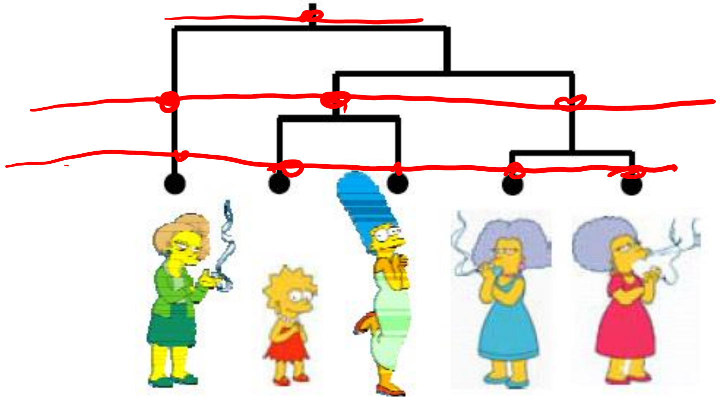
Clustering Applications

Jigsaw puzzles!



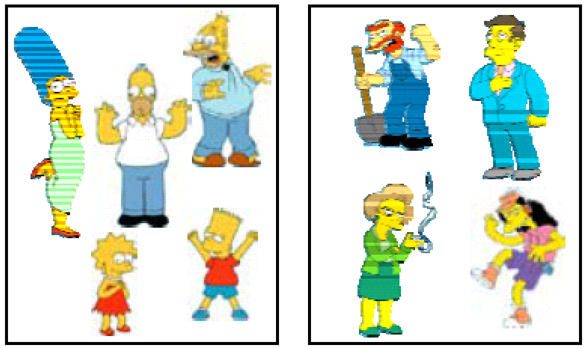
Clustering Algorithms

- Hierarchical algorithms
  - Bottom-up: Agglomerative Clustering
  - Top-down: Divisive



- Partition algorithms
  - K means clustering
  - Mixture-Model based clustering

probabilistic



Hierarchical Clustering

- Bottom-Up Agglomerative Clustering

  Starts with each object in a separate cluster, and repeat:
  - Joins the most similar pair of clusters,
  - Update the similarity of the new cluster to others

  until there is only one cluster.

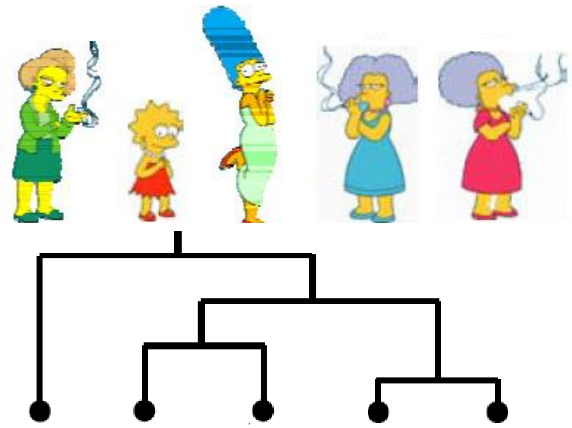
  Greedy - less accurate but simple to implement

- Top-Down divisive

  Starts with all the data in a single cluster, and repeat:
  - Split each cluster into two using a partition algorithm

  Until each object is a separate cluster.

  More accurate but complex to implement



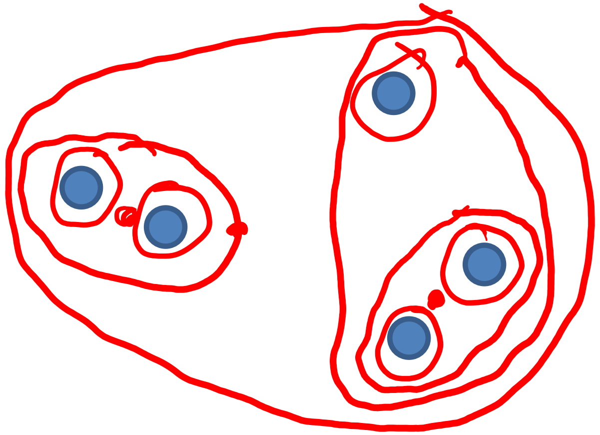
Hierarchical Clustering

- Bottom-Up Agglomerative Clustering

  Starts with each object in a separate cluster, and repeat:
  - Joins the most similar pair of clusters,
  - Update the similarity of the new cluster to others

  until there is only one cluster.

  Greedy – less accurate but simple to implement

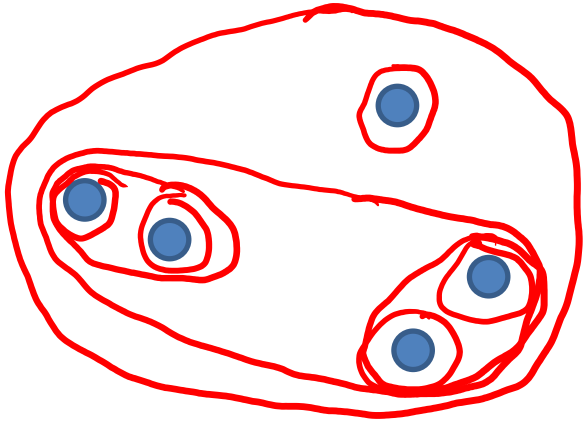


- Top-Down divisive

  Starts with all the data in a single cluster, and repeat:
  - Split each cluster into two using a partition algorithm

  Until each object is a separate cluster.

  More accurate but complex to implement



Partitioning Algorithms

- Partitioning method: Construct a partition of *N* objects into a set of *K* clusters
- Given: a set of objects and the number *K*
- Find: a partition of *K* clusters that optimizes the chosen partitioning criterion
  - Globally optimal: exhaustively enumerate all partitions
  - Effective heuristic method: K-means algorithm

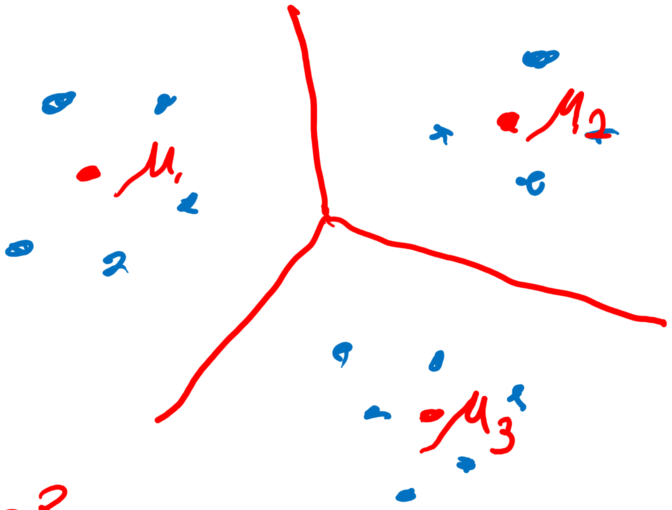
Slide credit: CMU MLD Aarti Singh

Course Feedback

See Piazza for two surveys

- FCE (Faculty Course Evaluation)
- TA survey
  - Fill out once for each TA that you can provide feedback for

K-Means



Algorithm

Input – Data, $\boldsymbol{x}^{(i)}$, Desired number of clusters, K

Initialize – the K cluster centers (randomly if necessary)

Iterate –

1. Assign points to the nearest cluster centers

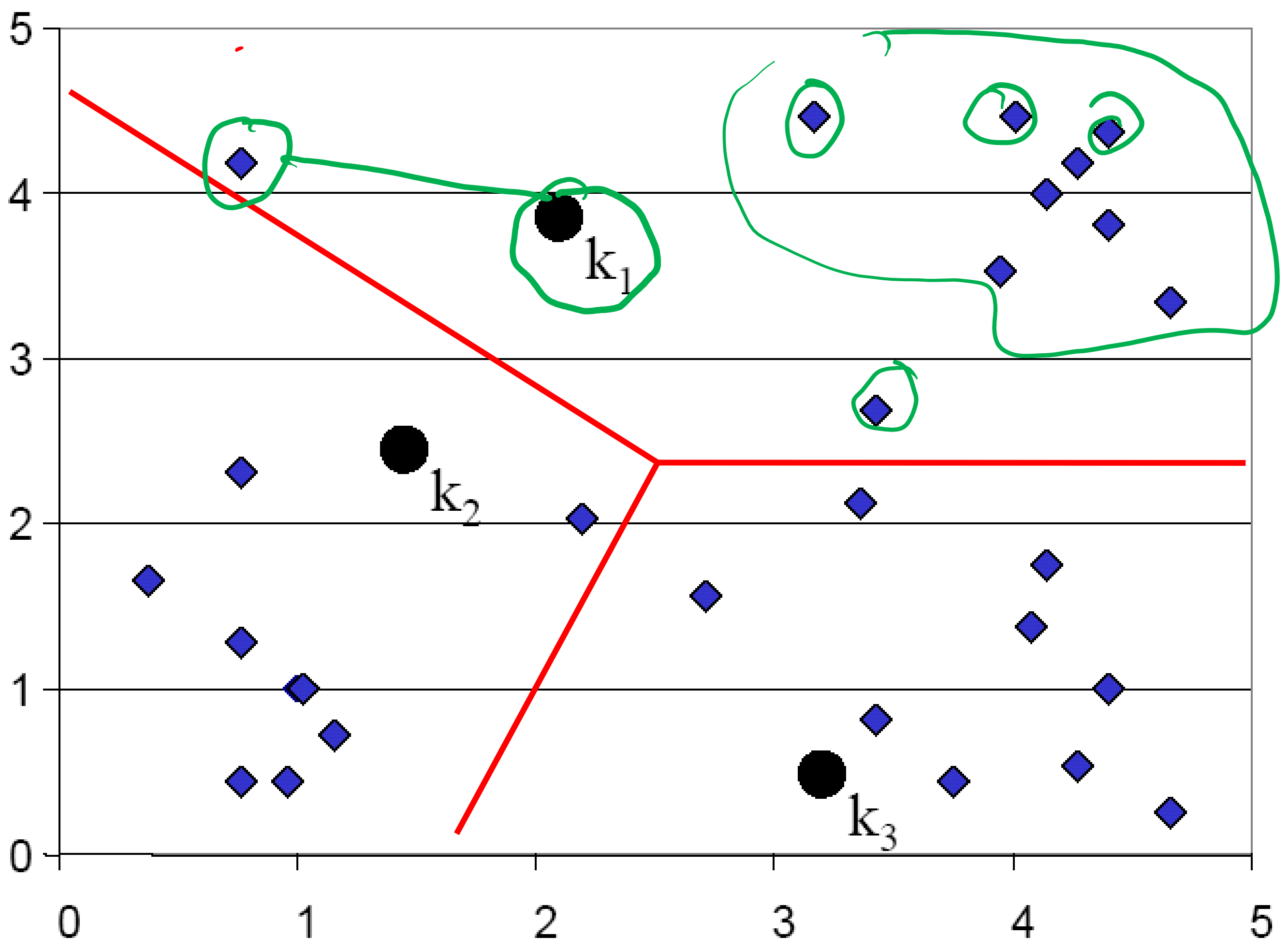
2. Re-estimate the K cluster centers (aka the centroid or mean), by assuming the memberships found above are correct.

Termination –

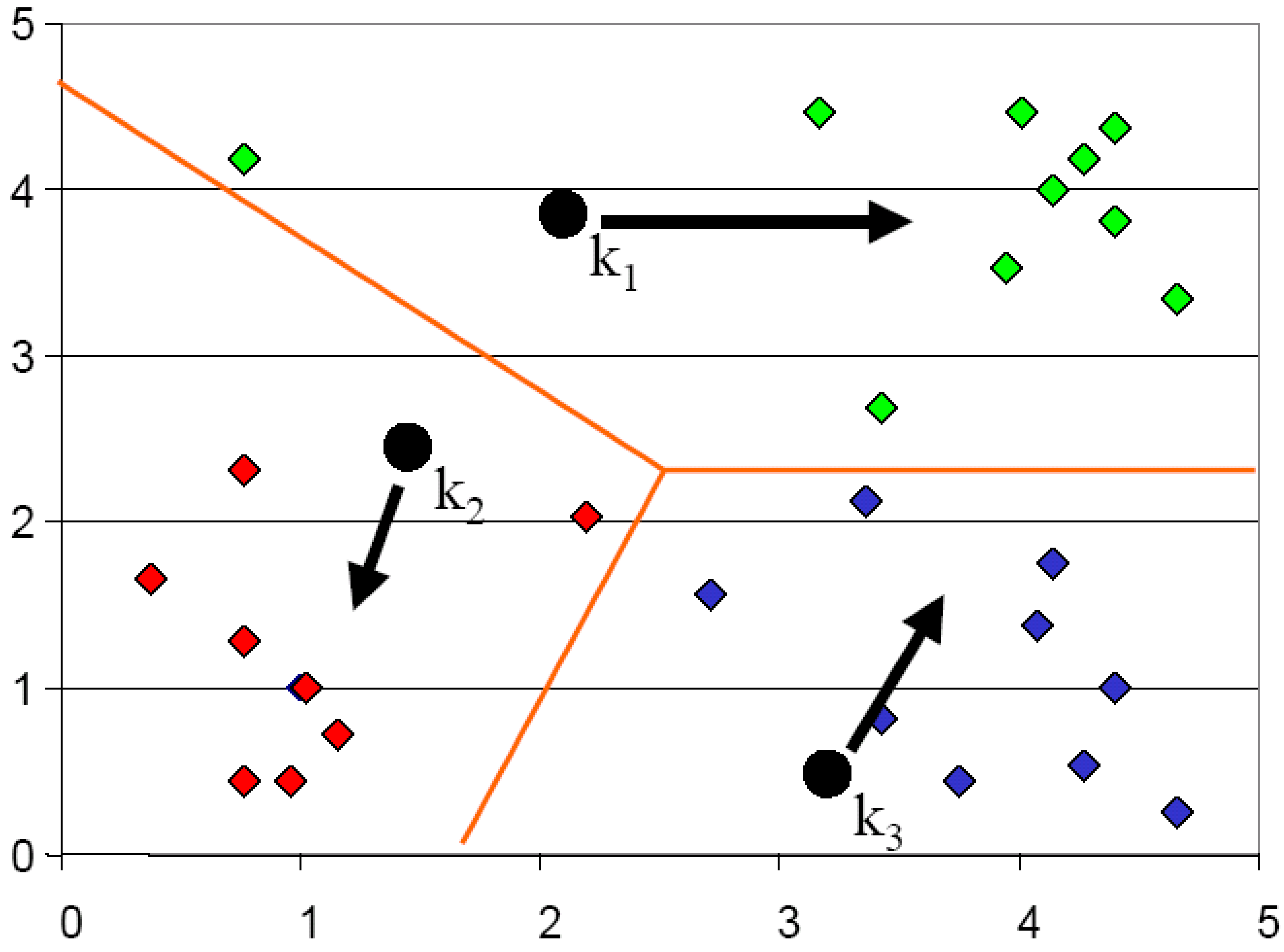
If none of the objects changed membership in the last iteration, exit. Otherwise go to 1.

Slide credit: CMU MLD Aarti Singh

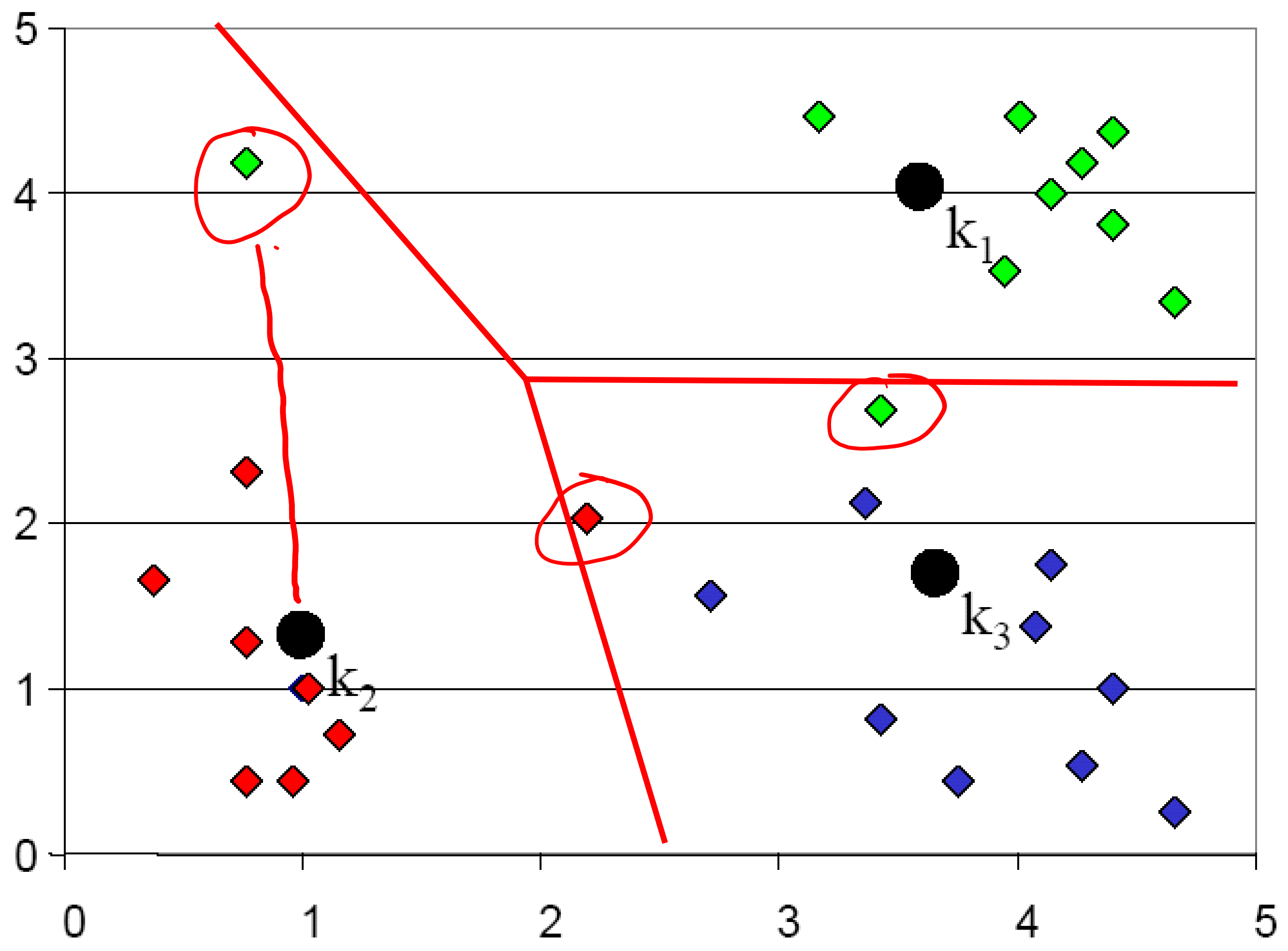
K-means Clustering: Assign points



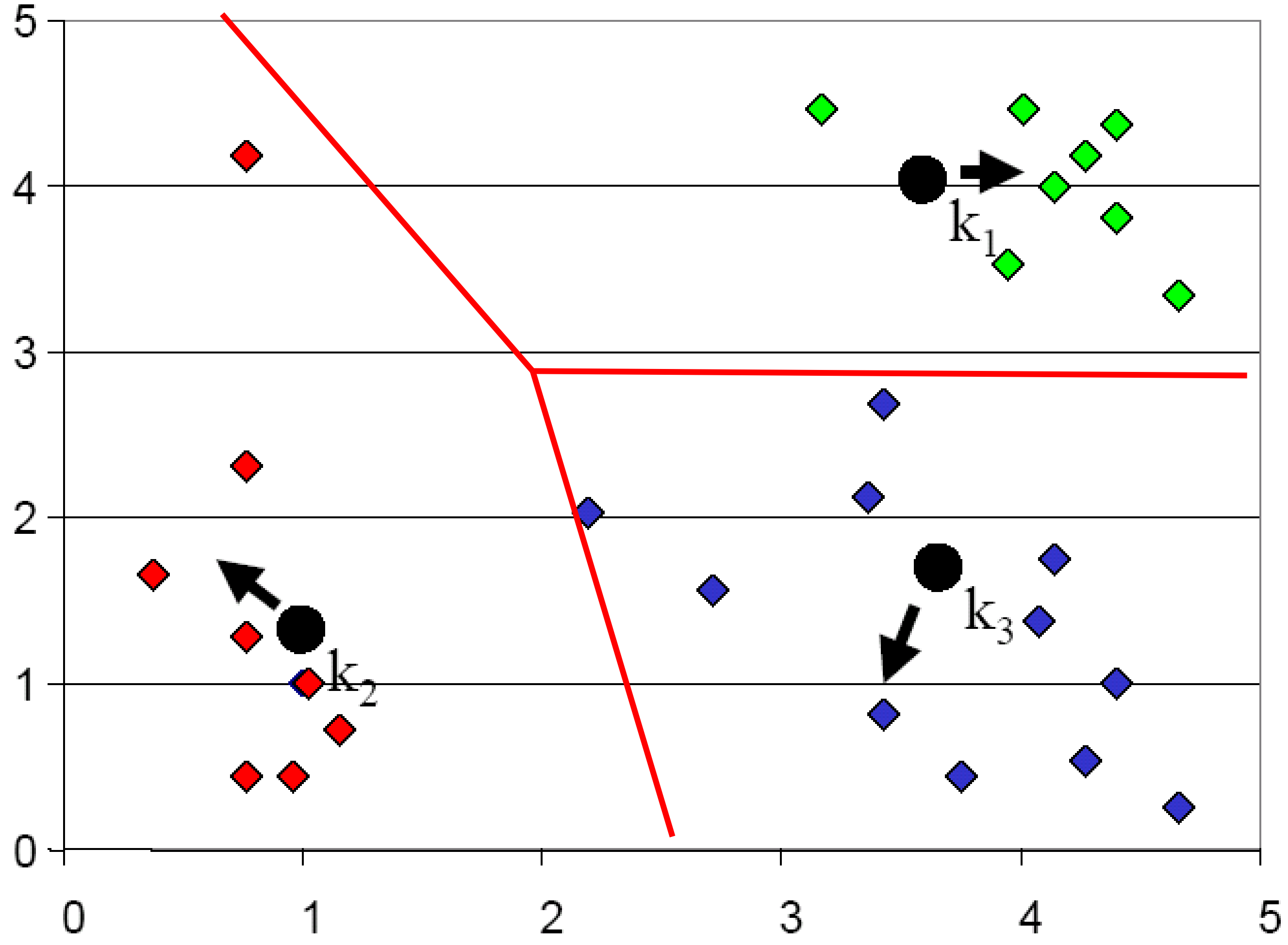
K-means Clustering: Update centers



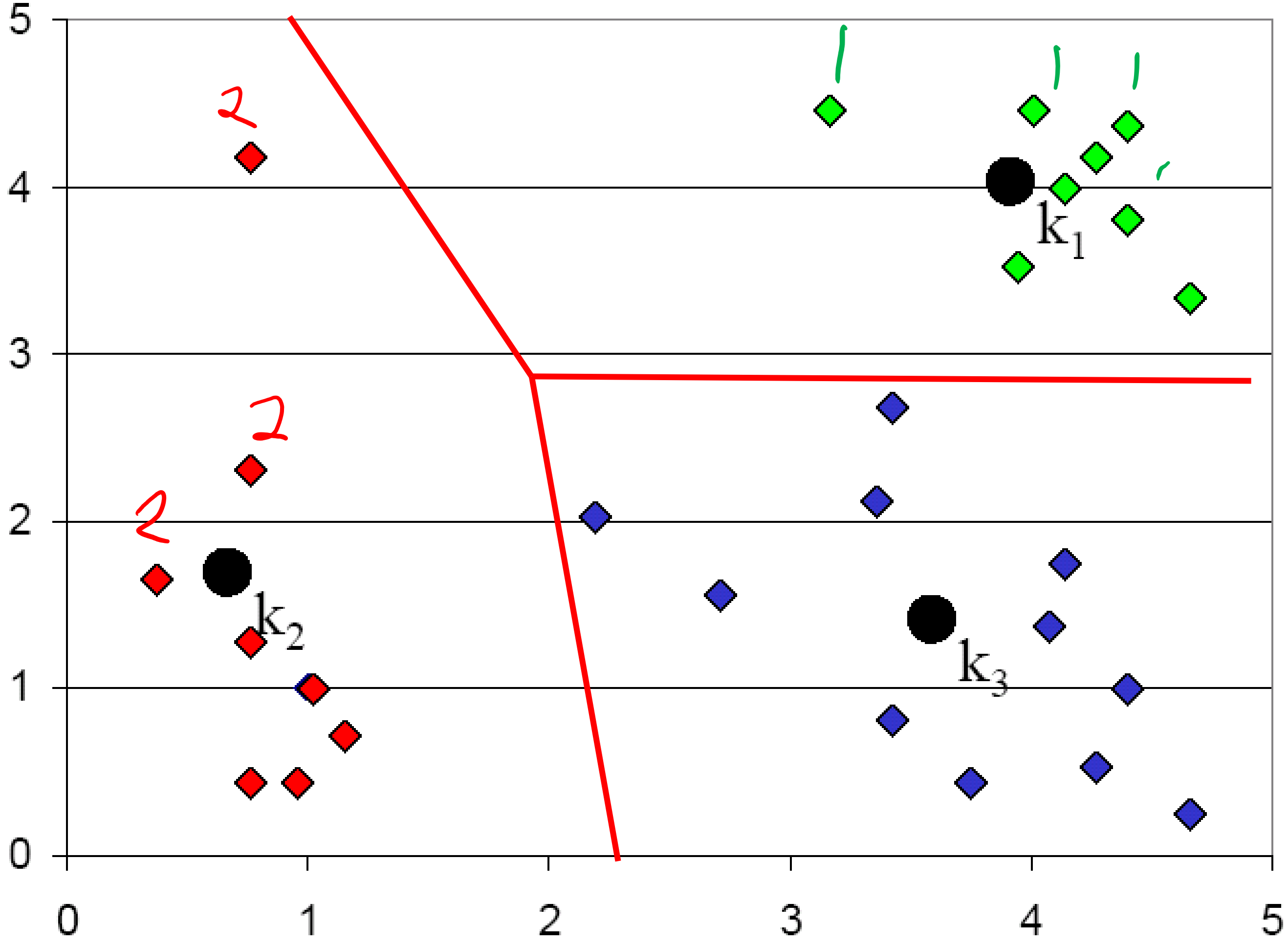
K-means Clustering: Assign points



K-means Clustering: Update centers



K-means Clustering: Assign points



K-means Optimization

Optimization recipe

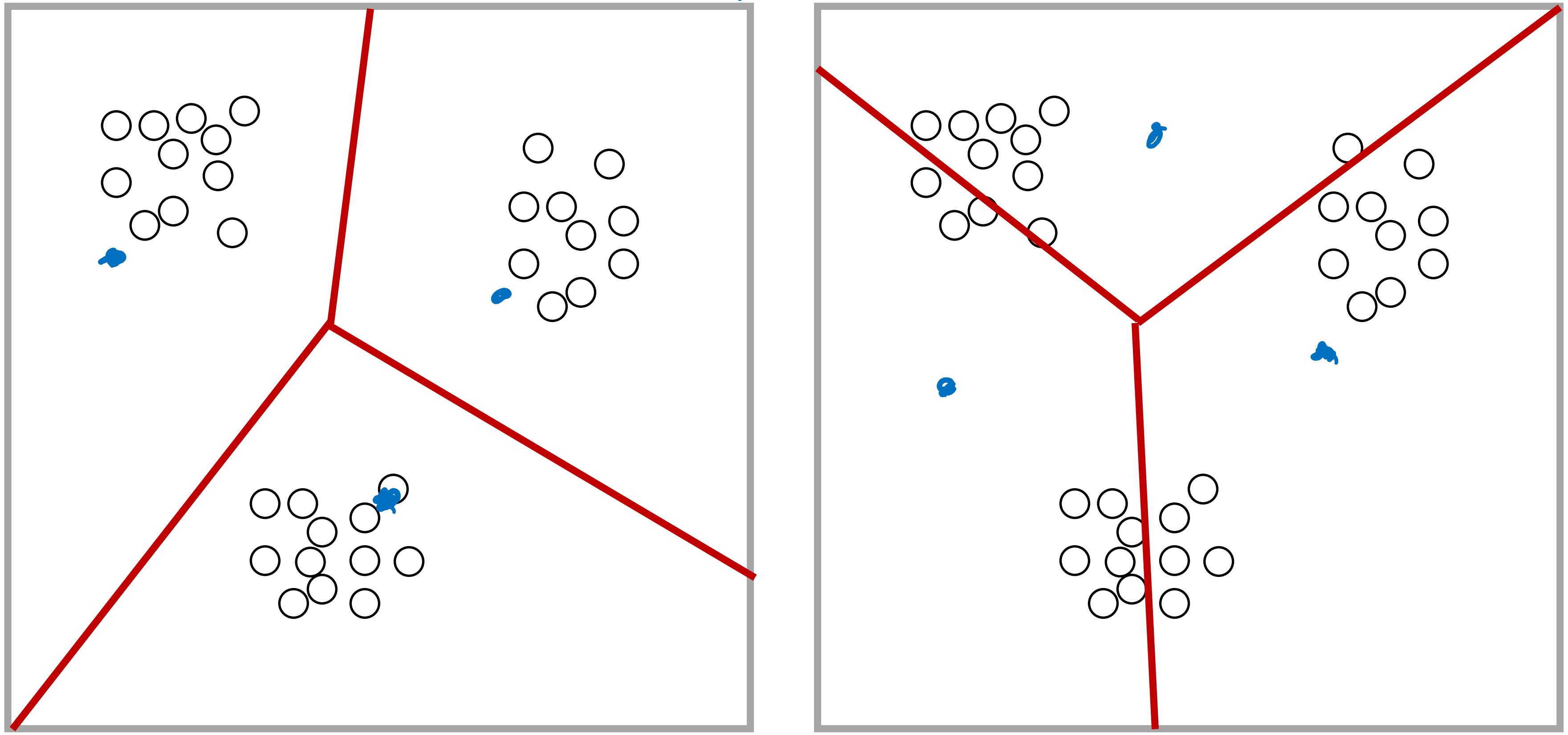
1. Formulate objective
2. Minimize objective

Slide credit: CMU MLD Matt Gormley

K-means Optimization

Question: Which of these partitions is "better"?

$J(\mu_1, \mu_2, \mu_3)$



K-means Optimization

Input: K, $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}$, $\boldsymbol{x}^{(i)} \in \mathbb{R}^M$ — Num clusters, unlabeled data

Output: $z^{(1)}, \ldots, z^{(N)}$, $z^{(i)} \in \{1 \ldots K\}$ — Cluster assignments per point

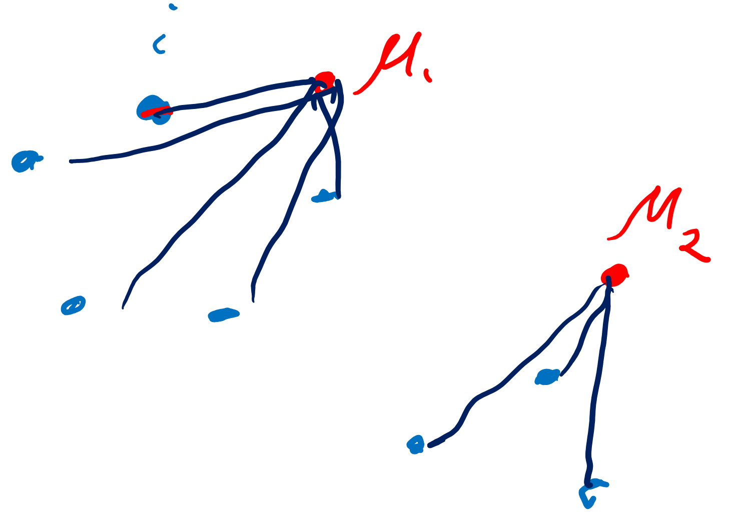
Output: $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$, $\boldsymbol{\mu}_k \in \mathbb{R}^M$ — Cluster centers

$$\underset{\vec{\mu}_1 \ldots \vec{\mu}_K}{\arg\min} \sum_{i=1}^{N} \min_{j \in \{1..K\}} \| x^{(i)} - \mu_j \|_2^2$$

$$\| v_1 - v_2 \|_2^2$$

$$\underset{\mu_1 \ldots \mu_K}{\arg\min} \sum_{i=1}^{N} \min_{z^{(i)} \in \{1 \ldots K\}} \| x^{(i)} - \mu_{z^{(i)}} \|_2^2$$

$$\underset{\mu_1 .. \mu_K, z^{(1)}, \ldots, z^{(N)}}{\arg\min} \sum_{i=1}^{N} \| x^{(i)} - \mu_{z^{(i)}} \|_2^2$$



K-means Optimization

Computational complexity

$$\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{z} = \underset{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{z}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{z^{(i)}} \right\|_2^2$$

$\vec{x}, \vec{a} \in \mathbb{R}^M$

K

if $M \geq 2$ and $K \geq 2$, NP-hard ← not convex

Easy $K=1$

$$\mu_1 = \underset{\mu_1}{\operatorname{argmin}} \sum_{i=1}^{N} \|x^{(i)} - \mu_1\|_2^2$$

$$= \frac{1}{N} \sum_{i=1}^{N} \vec{x}^{(i)}$$

K-means Optimization

Alternating minimization

a) $\boldsymbol{z} = \underset{\boldsymbol{z}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{z^{(i)}} \right\|_2^2$ fix μ

b) $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K = \underset{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{z^{(i)}} \right\|_2^2$ fix z

a) $z^{(1)} = \underset{z^{(1)} \in \{1..K\}}{\operatorname{argmin}} \| x^{(1)} - \mu_{z^{(1)}} \|_2$

$\vdots$

$z^{(N)}$

b) $\mu_1 = \underset{\mu_1}{\operatorname{argmin}} \sum_{i : z^{(i)} = 1} \| x^{(i)} - \mu_1 \|_2^2$

$\mu_K =$

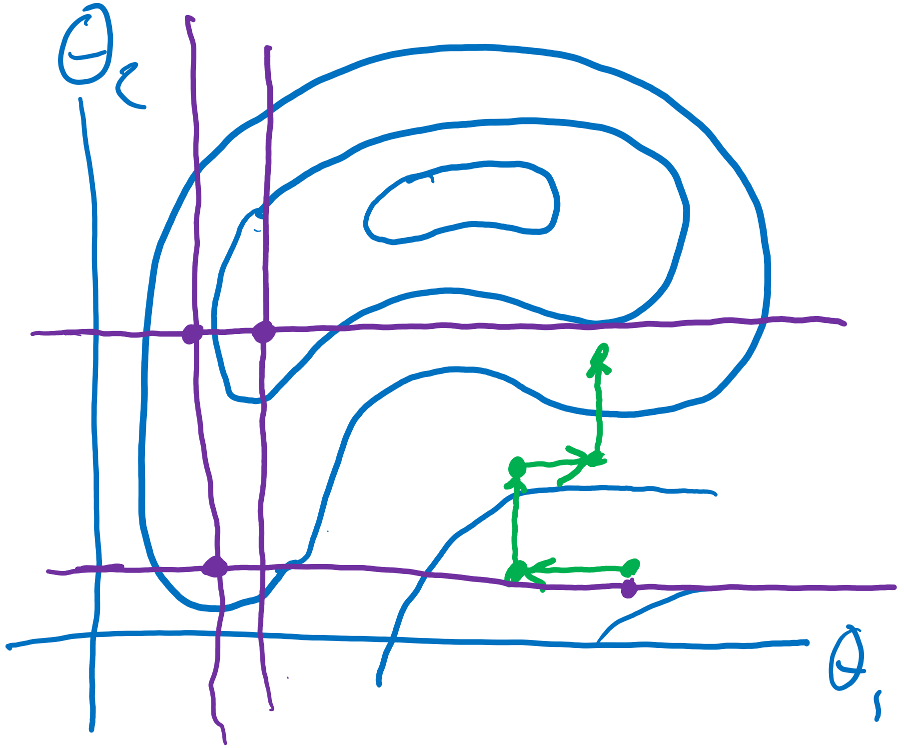
Alternating minimization

Coordinate descent

Two different approaches

$$\min_{\theta_1,\theta_2} J(\theta_1,\theta_2)$$

1. Step based on derivative for one parameter
   a. $\theta_1 \leftarrow \theta_1 - \eta\, \partial J/\partial\theta_1$ fix θ_2
   b. $\theta_2 \leftarrow \theta_2 - \eta\, \partial J/\partial\theta_2$ fix θ_1
2. Find minimum for one parameter
   a. $\theta_1 \leftarrow \underset{\theta_1}{\operatorname{argmin}} J(\theta_1,\theta_2)$ fix θ_2
   b. $\theta_2 \leftarrow \underset{\theta_2}{\operatorname{argmin}} J(\theta_1,\theta_2)$ fix θ_1



Alternating minimization

Block coordinate descent

Two different approaches

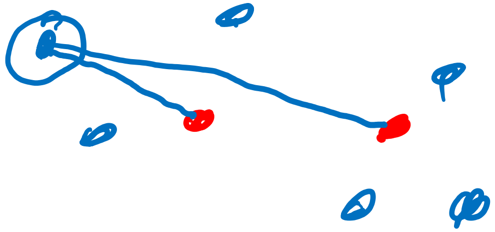
N K

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \; J(\boldsymbol{\alpha}, \boldsymbol{\beta})$$

1. Step based on gradient for one set of parameters (step size η)
   a. $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} - \eta \nabla_{\boldsymbol{\alpha}} J$
   b. $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} - \eta \nabla_{\boldsymbol{\beta}} J$

2. Find minimum for one set of parameter (no hyperparameters!)
   a. $\boldsymbol{\alpha} \leftarrow \operatorname{argmin}_{\boldsymbol{\alpha}} \mathrm{J}(\boldsymbol{\alpha}, \boldsymbol{\beta})$
   b. $\boldsymbol{\beta} \leftarrow \operatorname{argmin}_{\boldsymbol{\beta}} \mathrm{J}(\boldsymbol{\alpha}, \boldsymbol{\beta})$

Computational Complexity

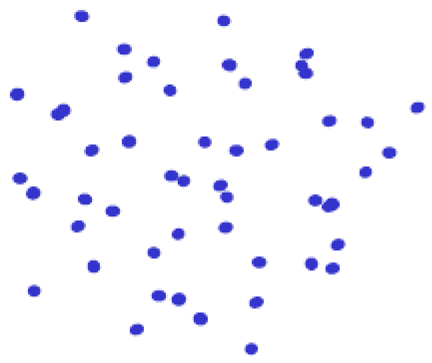


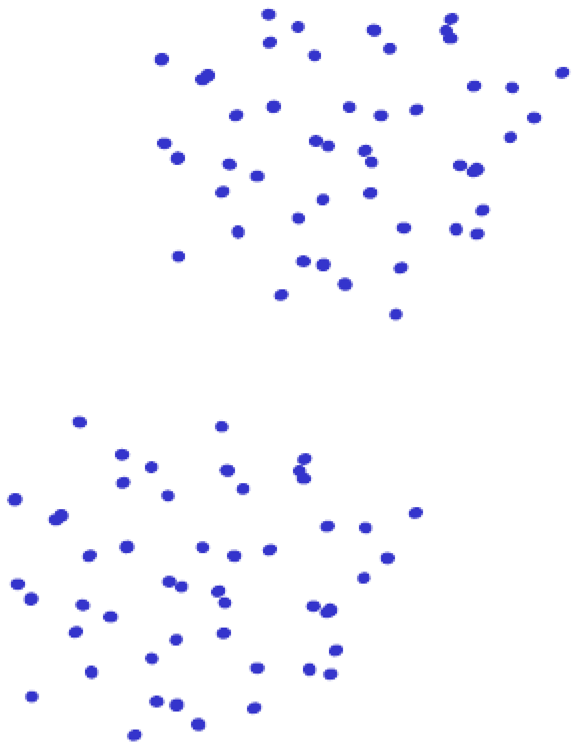
- At each iteration,
  - Computing cluster centers: Each object gets added once to some cluster: O(N)
  - Computing distance between each of the N objects and the K cluster centers is O(KN)

- Assume these two steps are each done once for l iterations: O(lKN)

Issues: Seed Choice

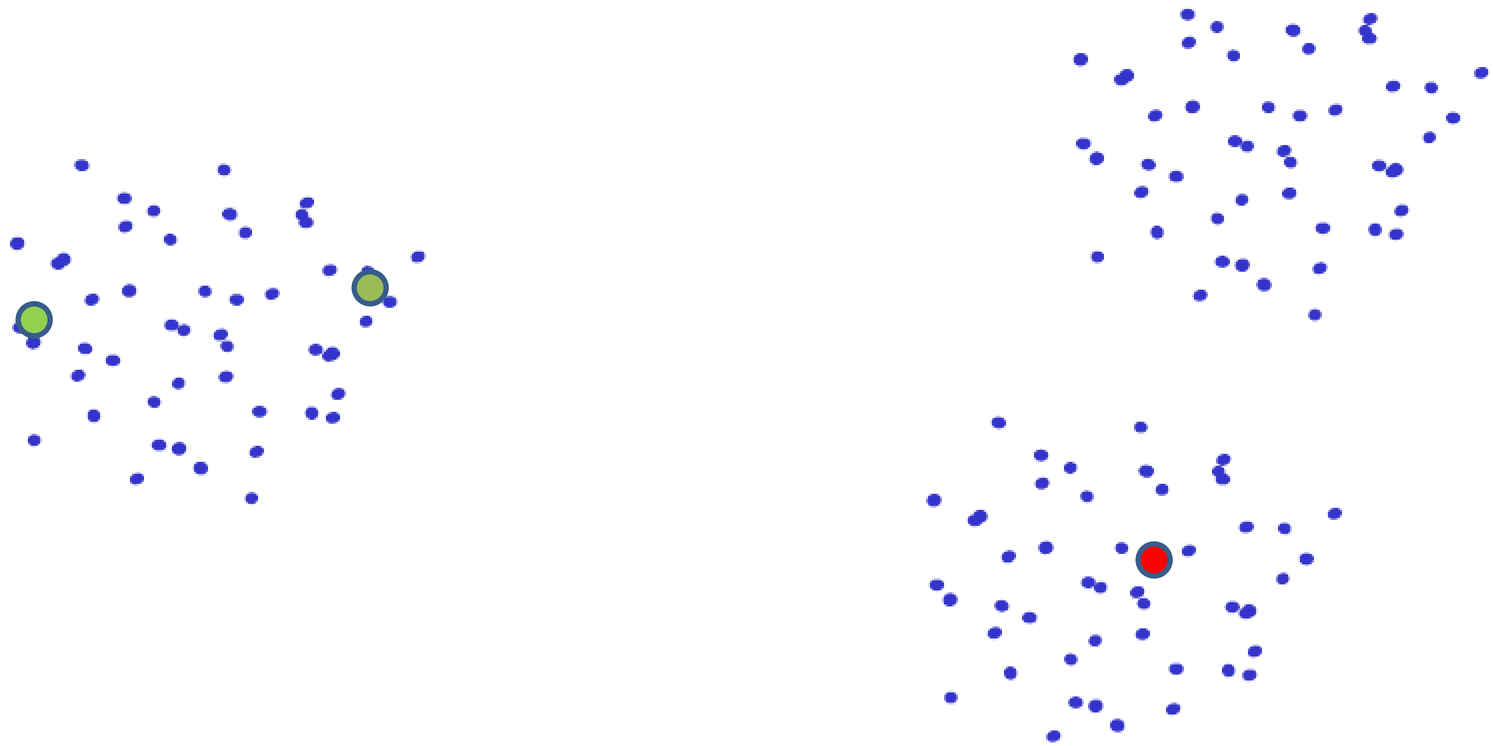
- Results are quite sensitive to seed selection.





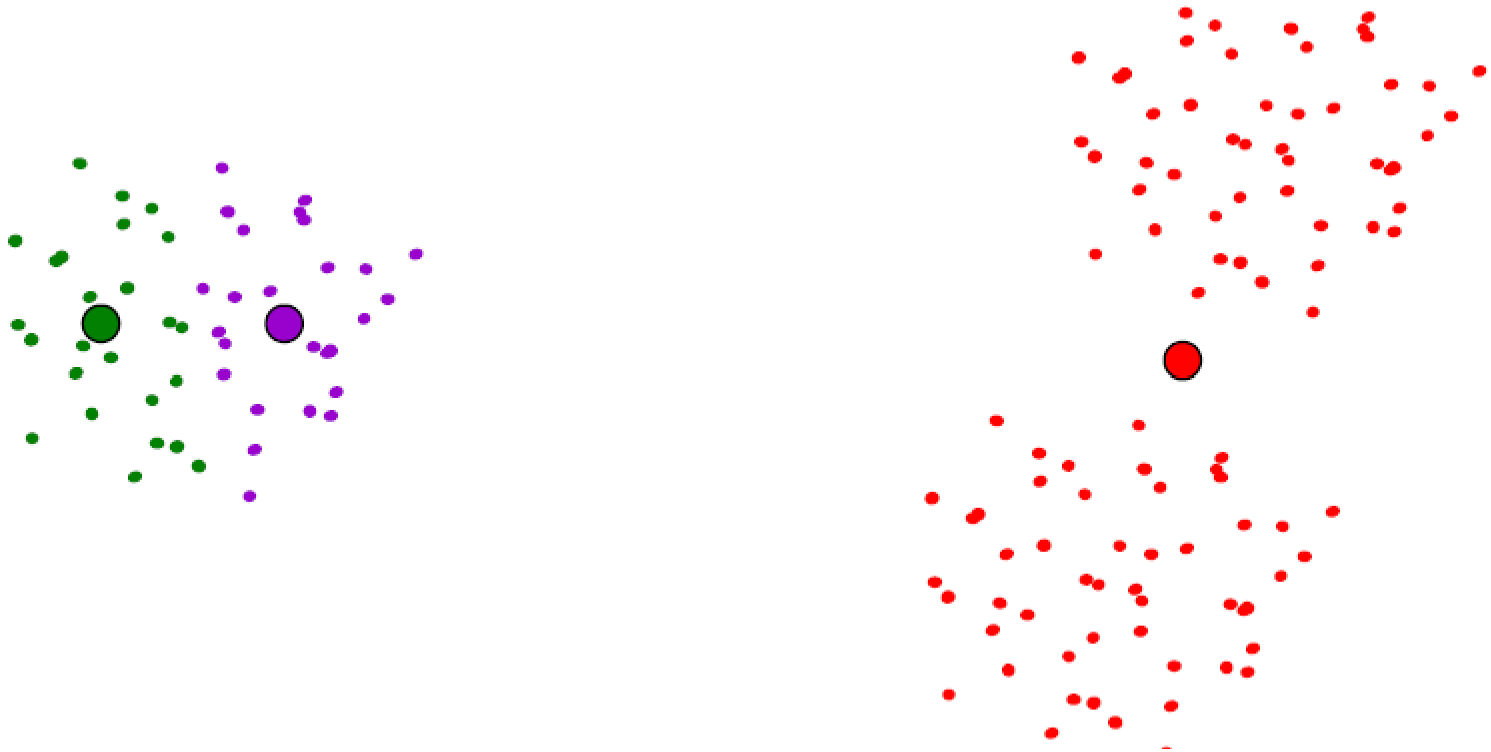
Issues: Seed Choice

- Results are quite sensitive to seed selection.



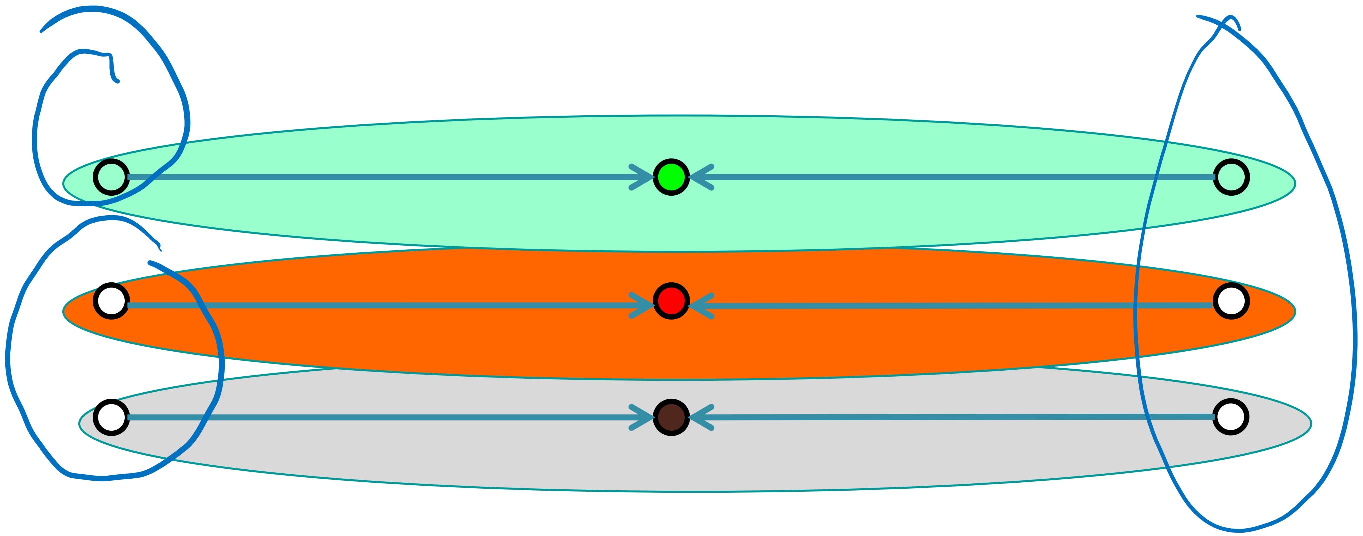
Issues: Seed Choice

- Results are quite sensitive to seed selection.



Slide credit: CMU MLD Aarti Singh

Issues: Seed Choice



K-means always converges, but it may converge at a local optimum that is different from the global optimum, and in fact could be arbitrarily worse in terms of its objective.

Issues: Seed Choice

- Results can vary based on random seed selection.

- Some seeds can result in poor convergence rate, or convergence to sub-optimal clustering.
  - Try out multiple starting points (very important!!!)
  - k-means ++ algorithm of Arthur and Vassilvitskii

    key idea: choose centers that are far apart

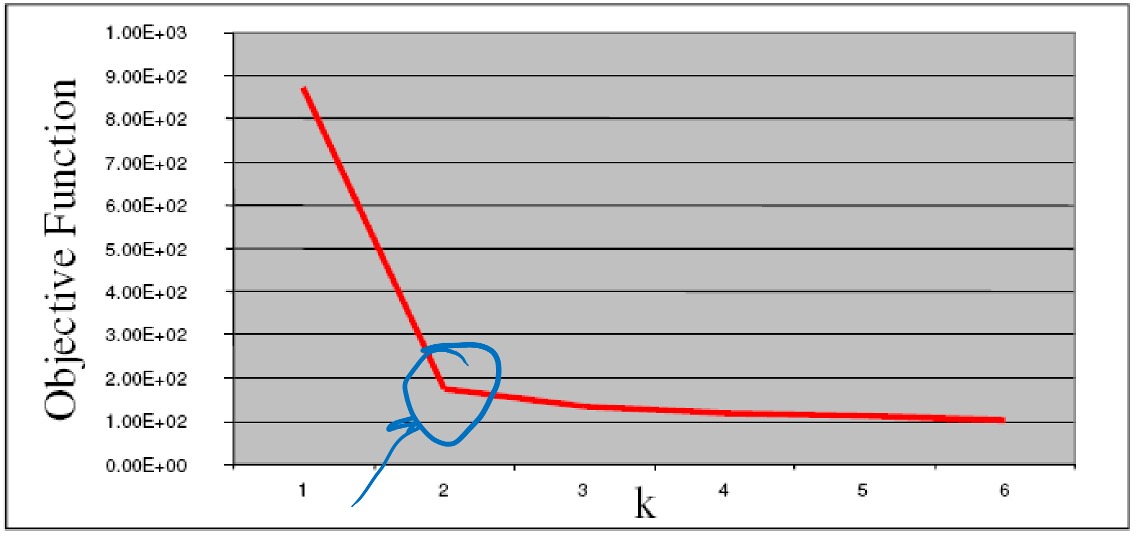
    (probability of picking a point as cluster center $\propto$ distance from nearest center picked so far)

Other Issues

- Number of clusters K
  - Objective function

$$\sum_{j=1}^{m} ||\mu_{C(j)} - x_j||^2$$

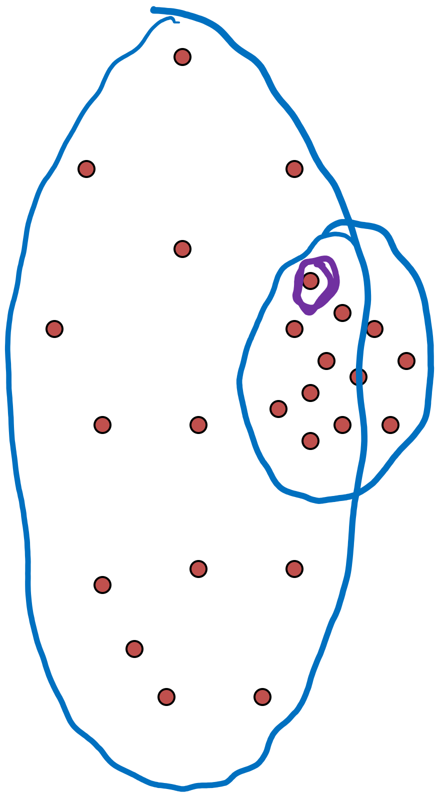
  - Look for "Knee" in objective function



  - Can you pick K by minimizing the objective over K?

Slide credit: CMU MLD Aarti Singh

(One) bad case for K-means



- Clusters may overlap
- Some clusters may be “wider” than others
- Clusters may not be linearly separable