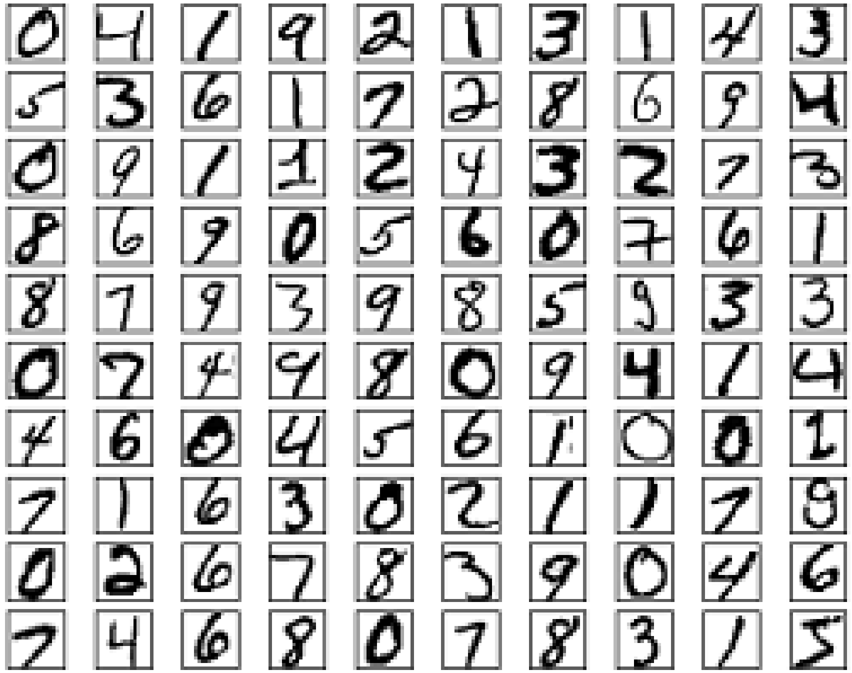
Announcements

Assignments

- HW9 (online)
  - Due Thu 4/16, 11:59 pm

Dimensionality Reduction

MNIST digit autoencoder



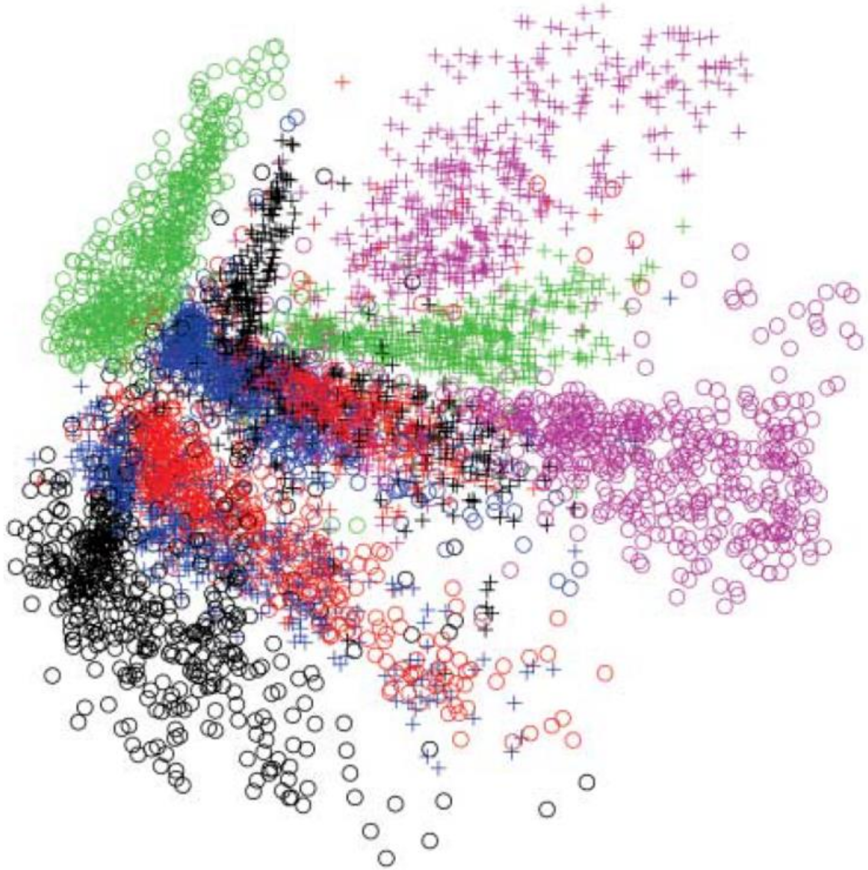


Image: Hinton & Salakhutdinov. *Science* 313.5786 (2006): 504-507.

Piazza Poll 1

Are autoencoders an example of unsupervised learning?

A.

B.

C.

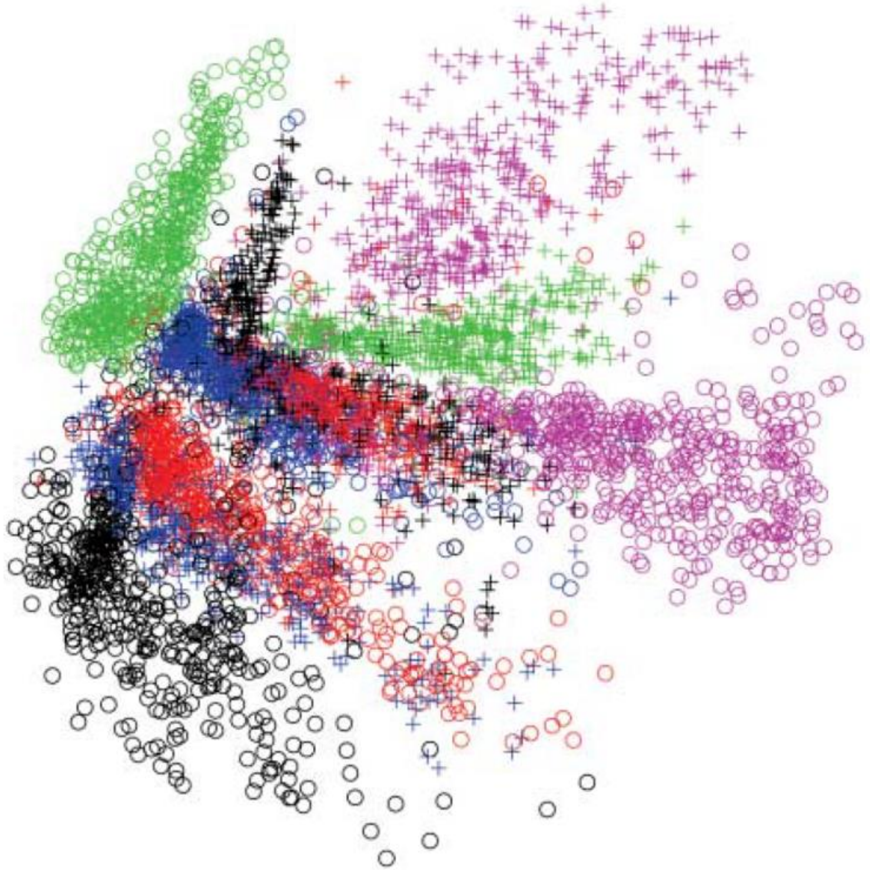
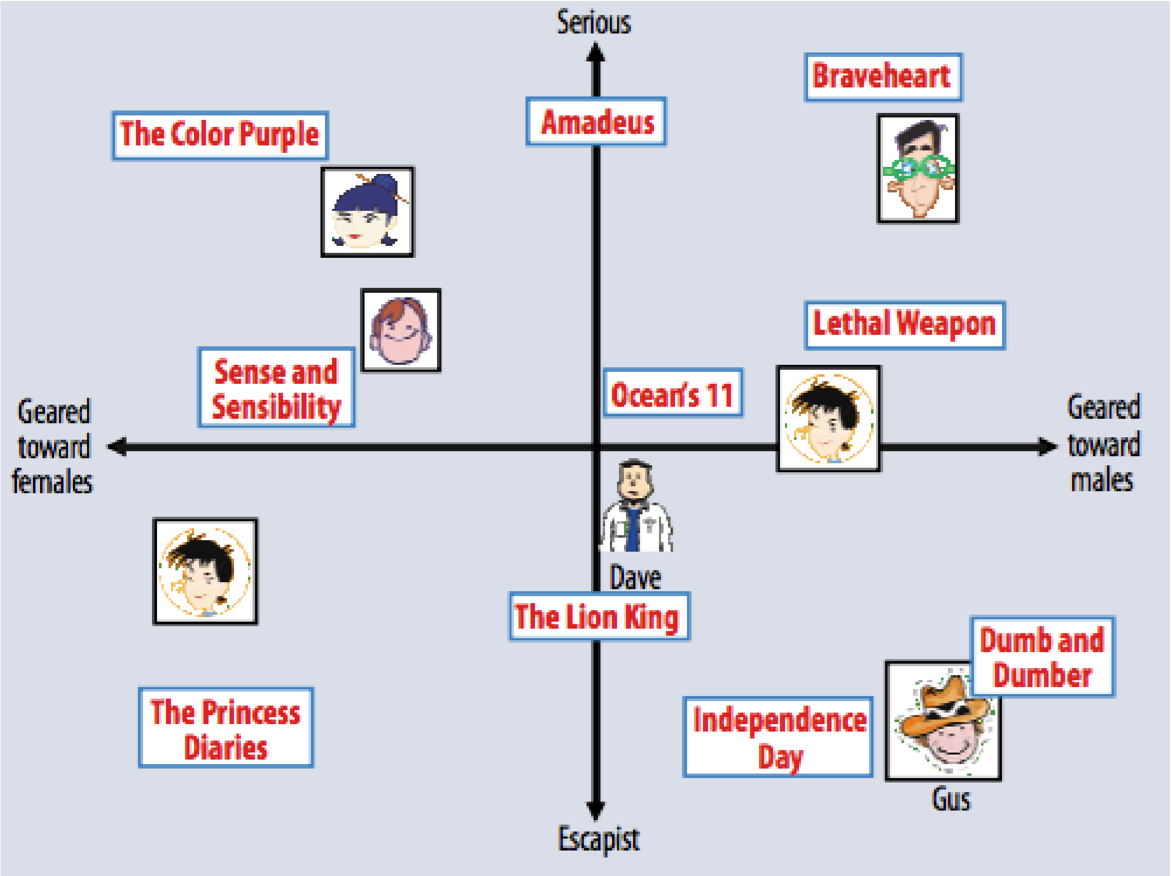


Image: Hinton & Salakhutdinov. *Science* 313.5786 (2006): 504-507.

Recommender Systems

Matrix Factorization



Figures from Koren et al. (2009)

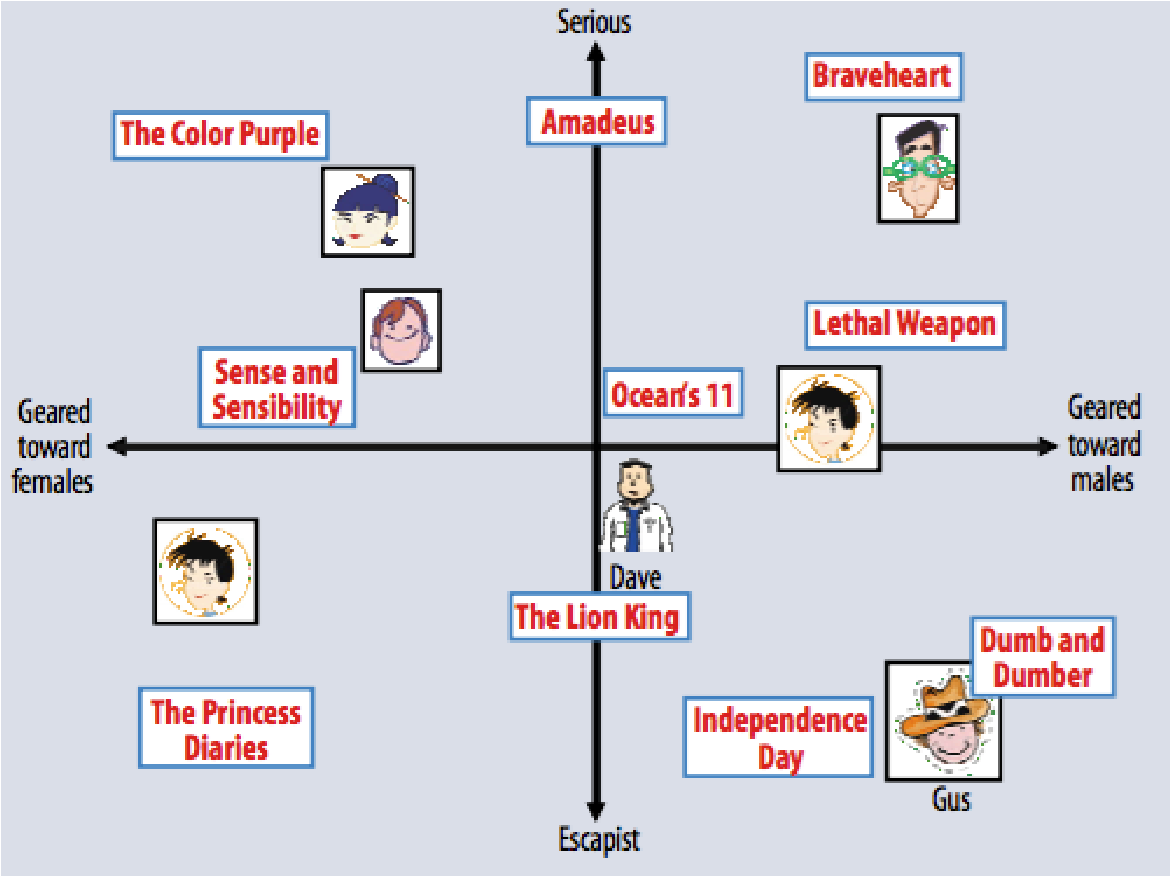
Piazza Poll 2

Are recommender systems an example of unsupervised learning?

A.

B.

C.



Figures from Koren et al. (2009)

Plan

Last week

- Dimensionality reduction
  - PCA
  - Autoencoders
- Recommender systems

This week

- Clustering

Next week

- Learning Theory

Introduction to Machine Learning

Clustering

Instructor: Pat Virtue

Clustering, Informal Goals

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

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

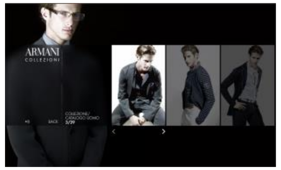
Useful for:

- 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).

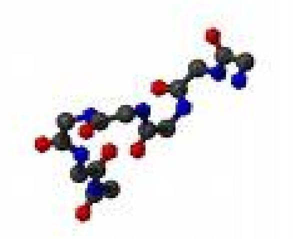
Slide credit: CMU MLD Nina Balcan

Applications (Clustering comes up everywhere…)

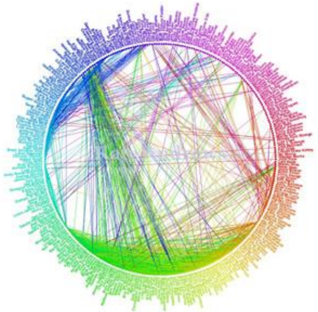
Cluster news articles or web pages or search results by topic.



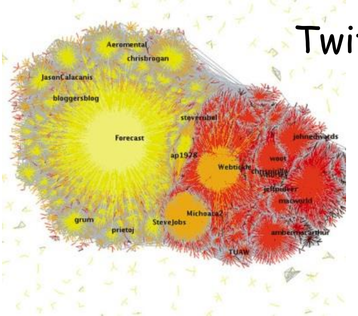
- Cluster protein sequences by function or genes according to expression profile.



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



Facebook network

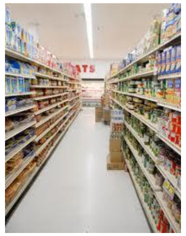


Twitter Network

Slide credit: CMU MLD Nina Balcan

Applications (Clustering comes up everywhere…)

Cluster customers according to purchase history.



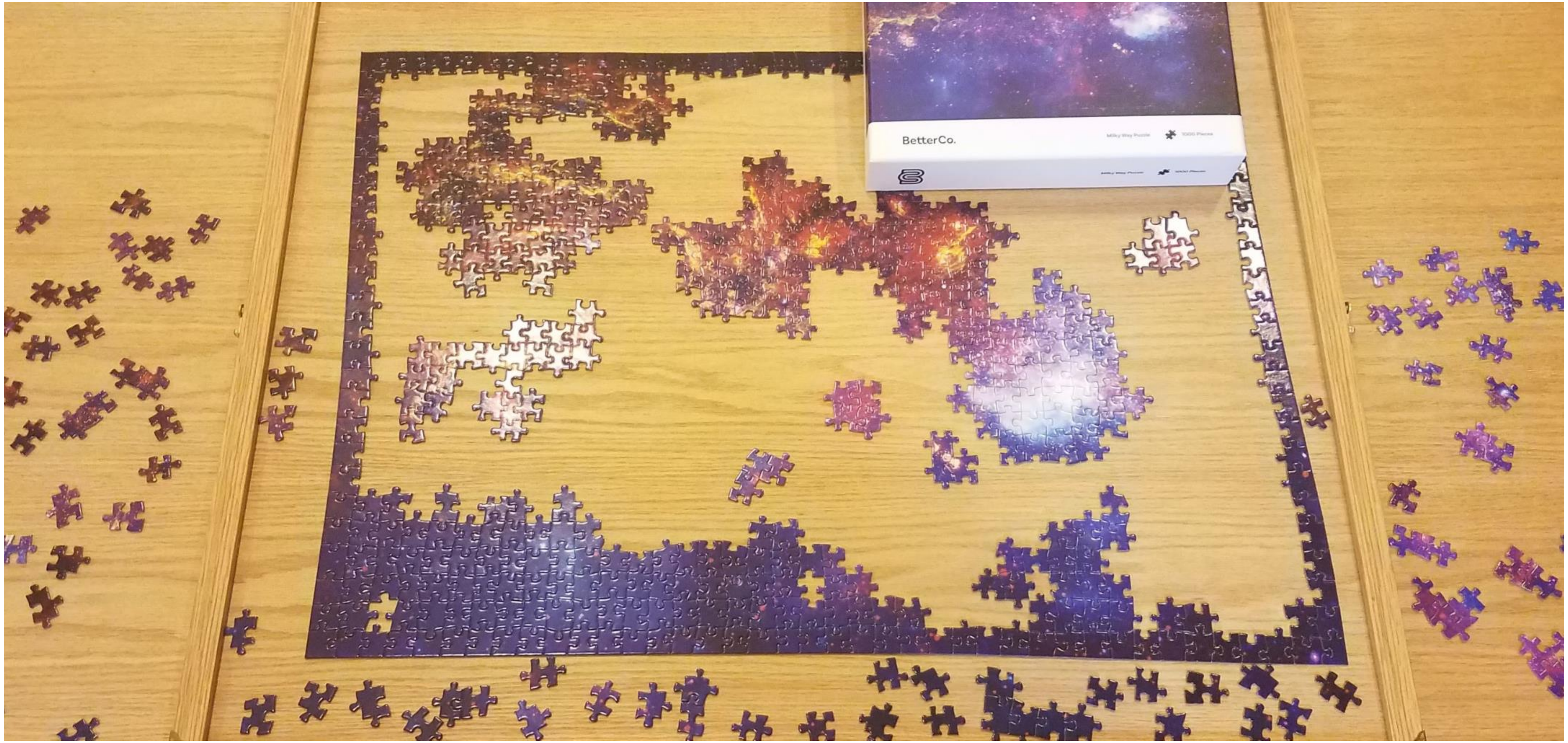
- Cluster galaxies or nearby stars (e.g. Sloan Digital Sky Survey)

- And many many more applications….

Slide credit: CMU MLD Nina Balcan

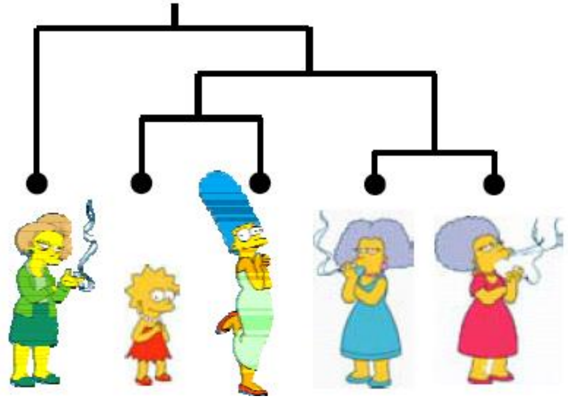
Clustering Applications

Jigsaw puzzles!

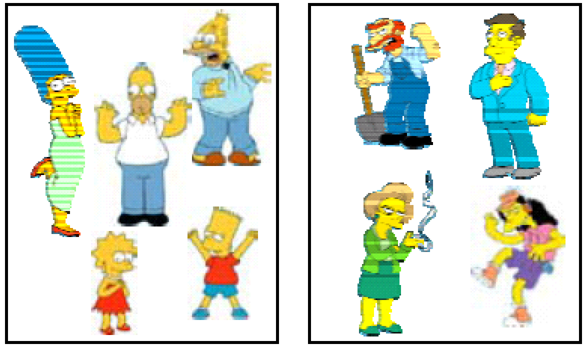


Clustering Algorithms

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



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



Slide credit: CMU MLD Aarti Singh and Eric Xing

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.

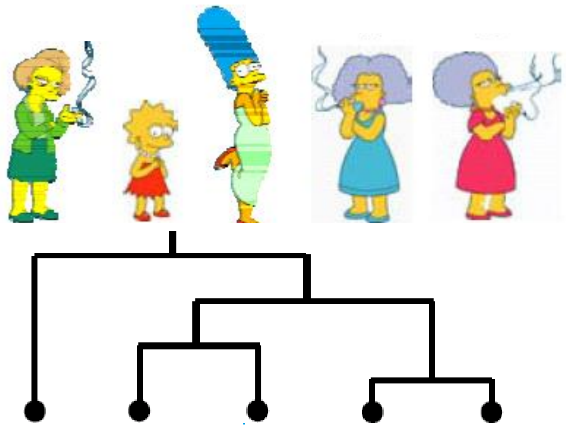
  Greedy – less accurate but simple to implement

- 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.

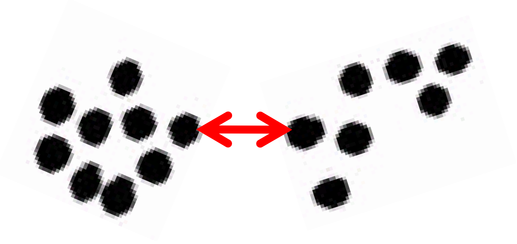
  More accurate but complex to implement



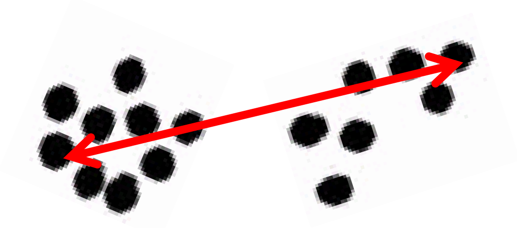
Bottom-up Agglomerative clustering

Different algorithms differ in how the similarities are defined (and hence updated) between two clusters

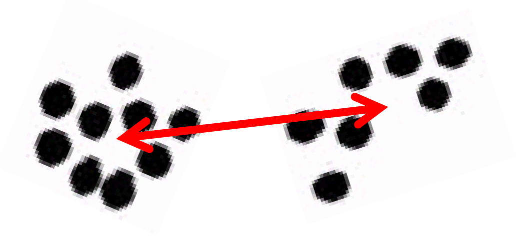
- Single-Linkage
  - Nearest Neighbor: similarity between their closest members.



- Complete-Linkage
  - Furthest Neighbor: similarity between their furthest members.



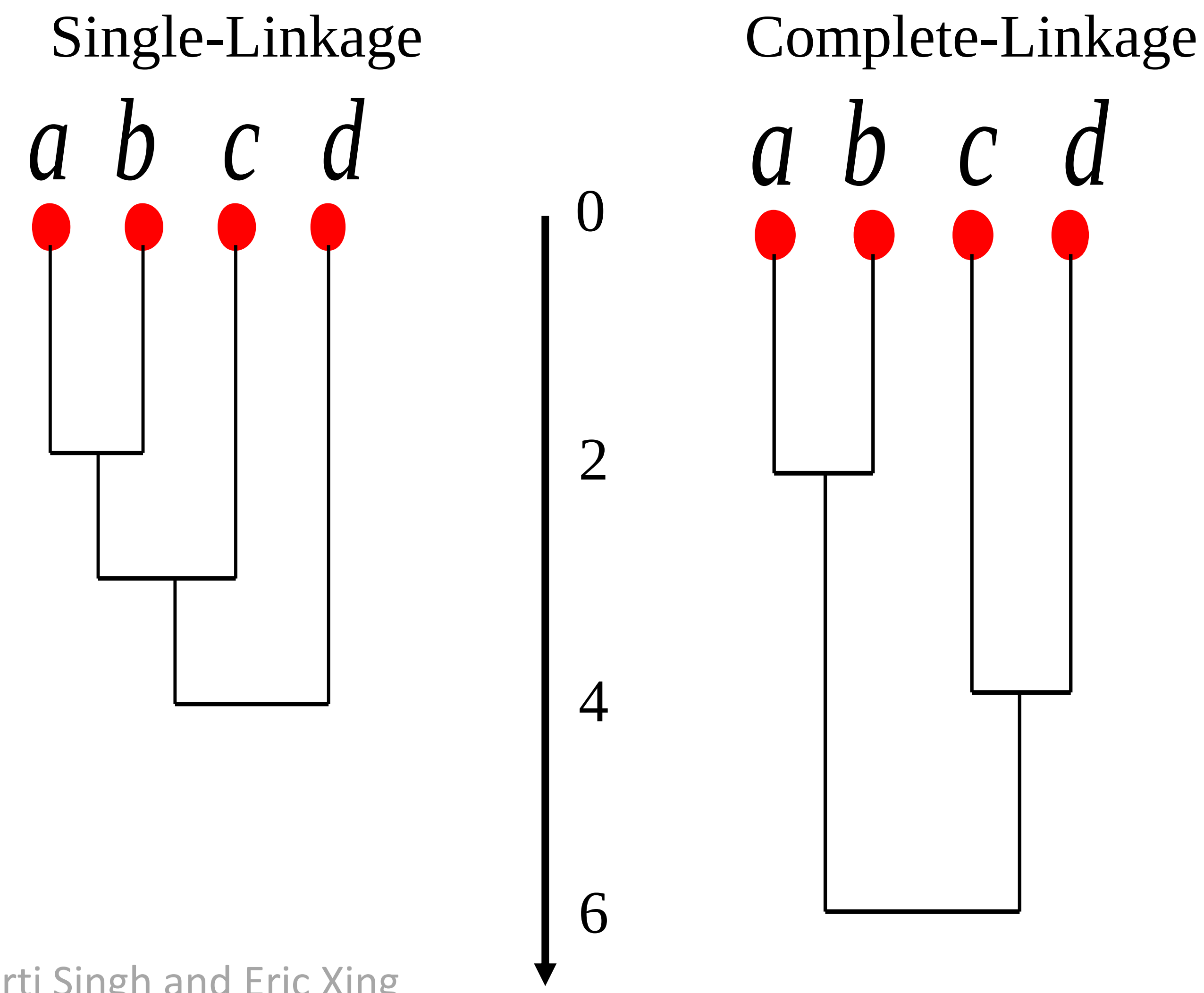
- Centroid
  - Similarity between the centers of gravity



- Average-Linkage
  - Average similarity of all cross-cluster pairs.

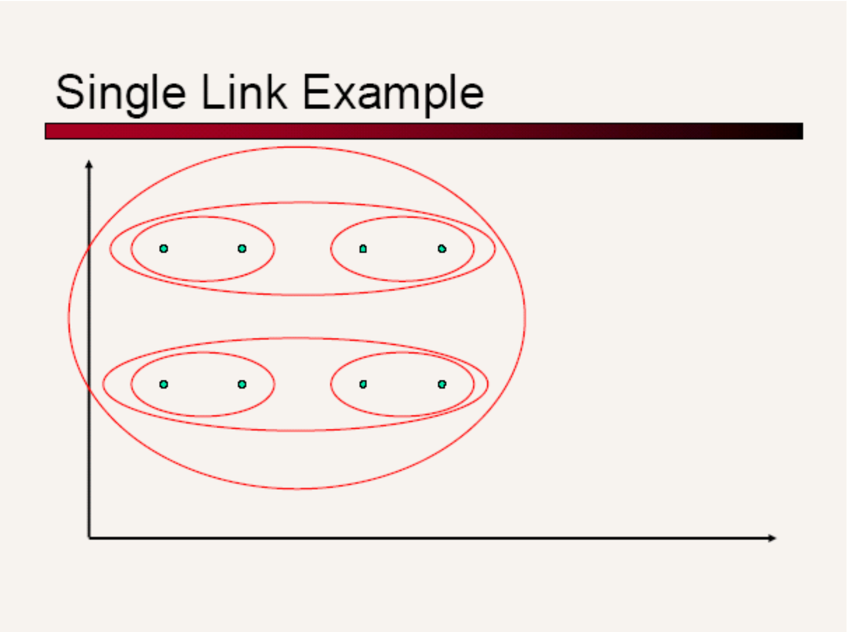
Slide credit: CMU MLD Aarti Singh

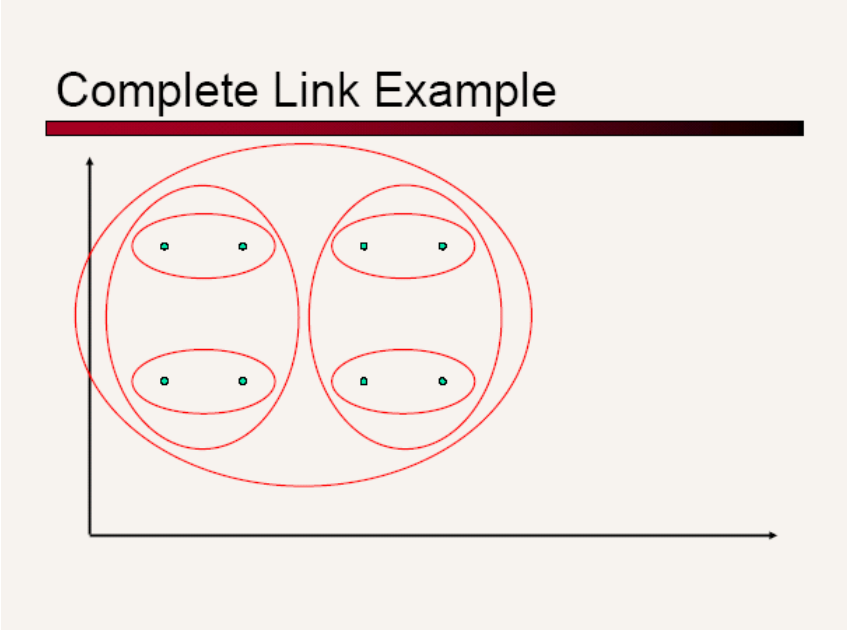
Dendrograms



Slide credit: CMU MLD Aarti Singh and Eric Xing

Another Example





Single vs. Complete Linkage

	Shape of clusters
Single-linkage	allows anisotropic and non-convex shapes
Complete-linkage	assumes isotopic, convex shapes

Slide credit: CMU MLD Aarti Singh

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

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

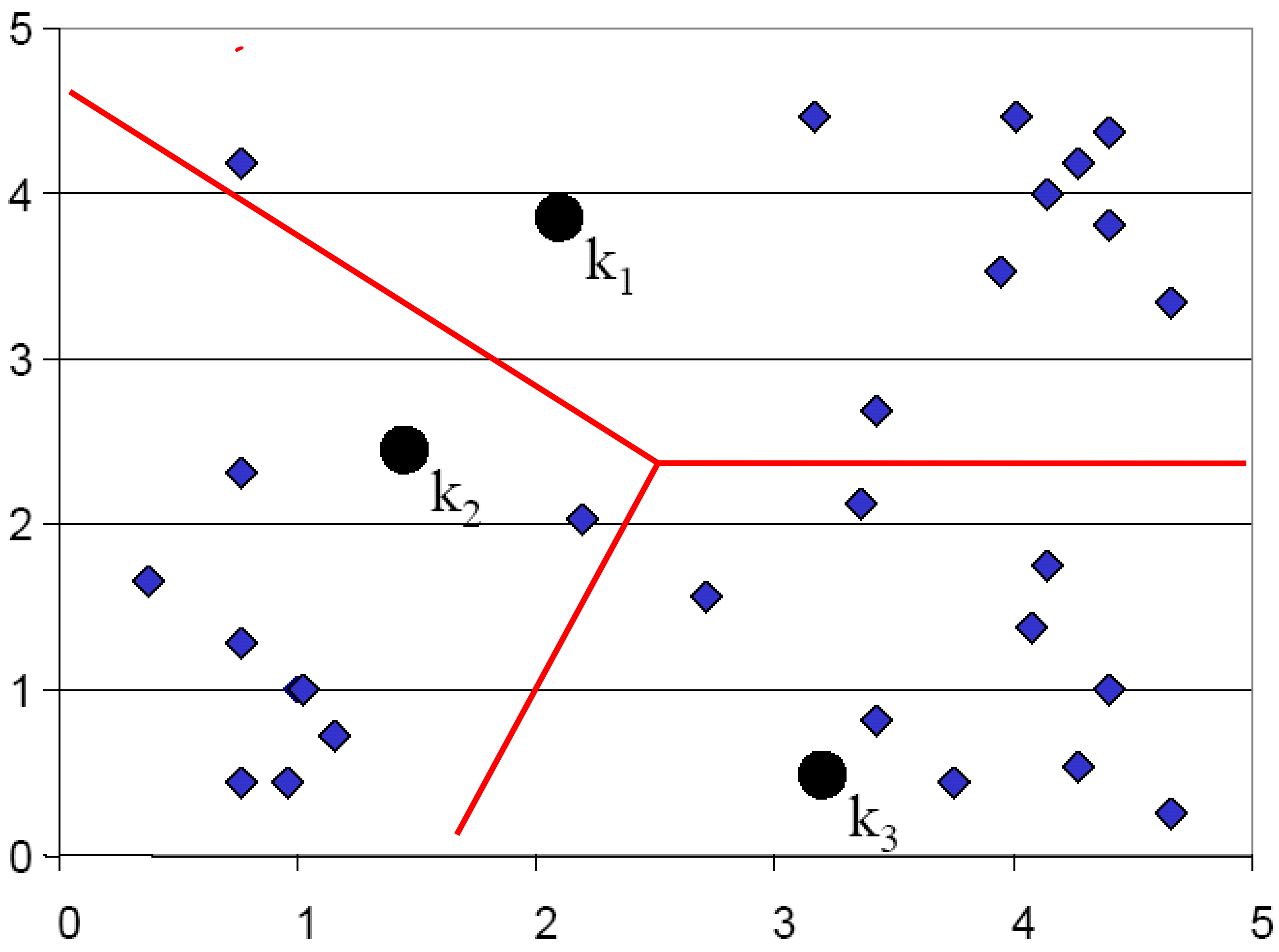
2. Re-estimate the K cluster centers (aka the centroid or mean), by assuming the memberships found above are correct.

Termination –

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

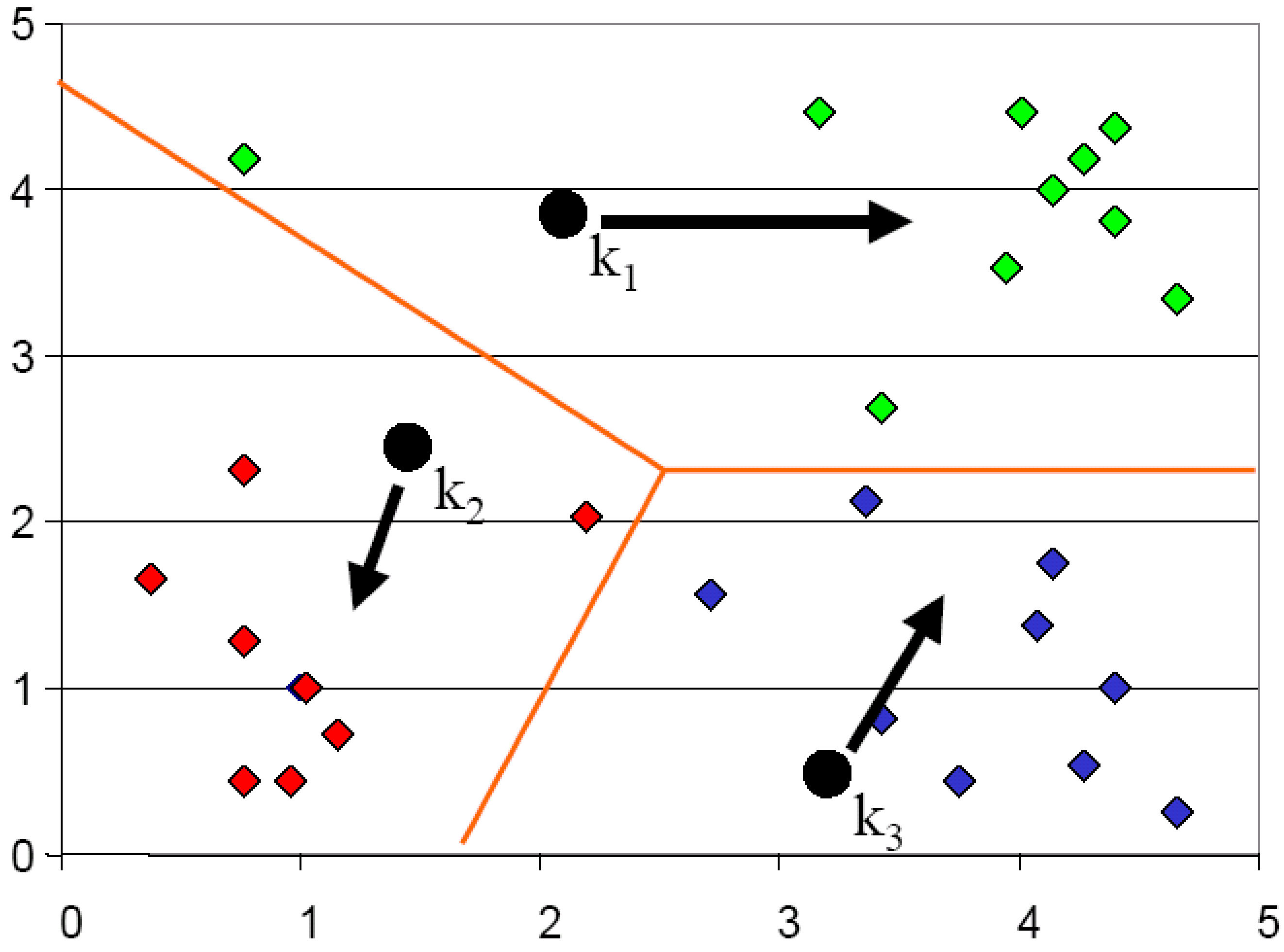
Slide credit: CMU MLD Aarti Singh

K-means Clustering: Step 1

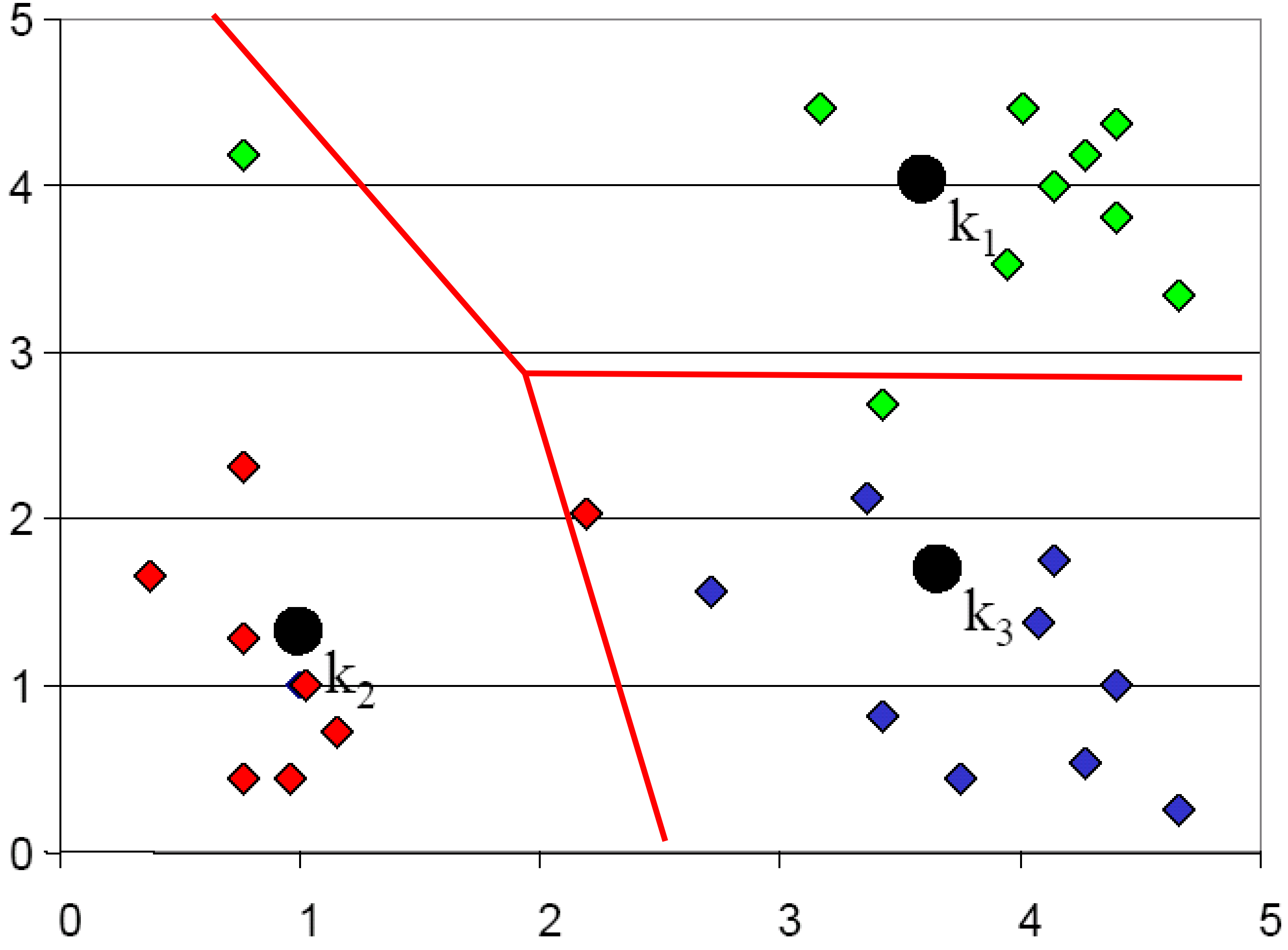


Voronoi diagram

K-means Clustering: Step 2

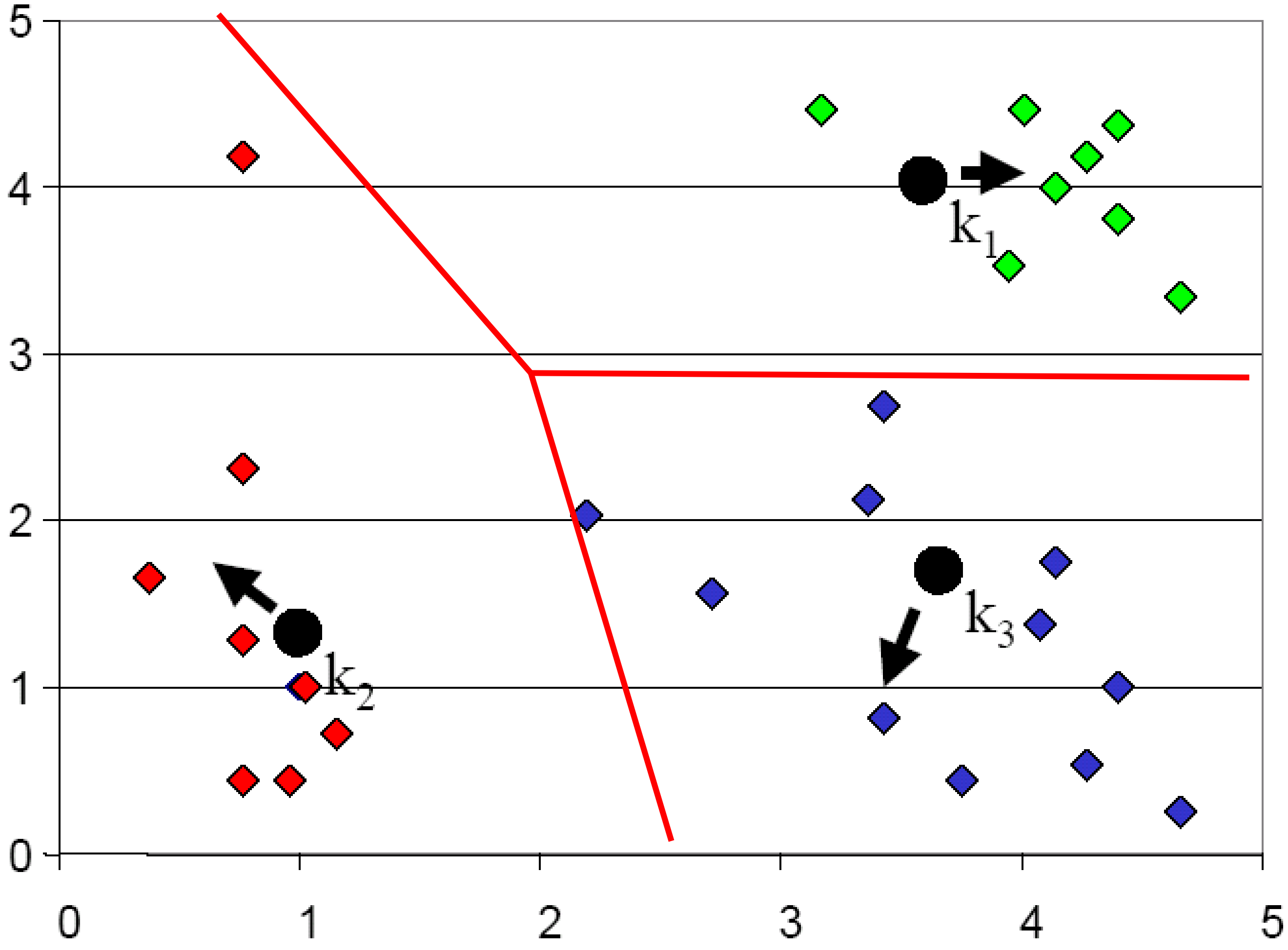


K-means Clustering: Step 3



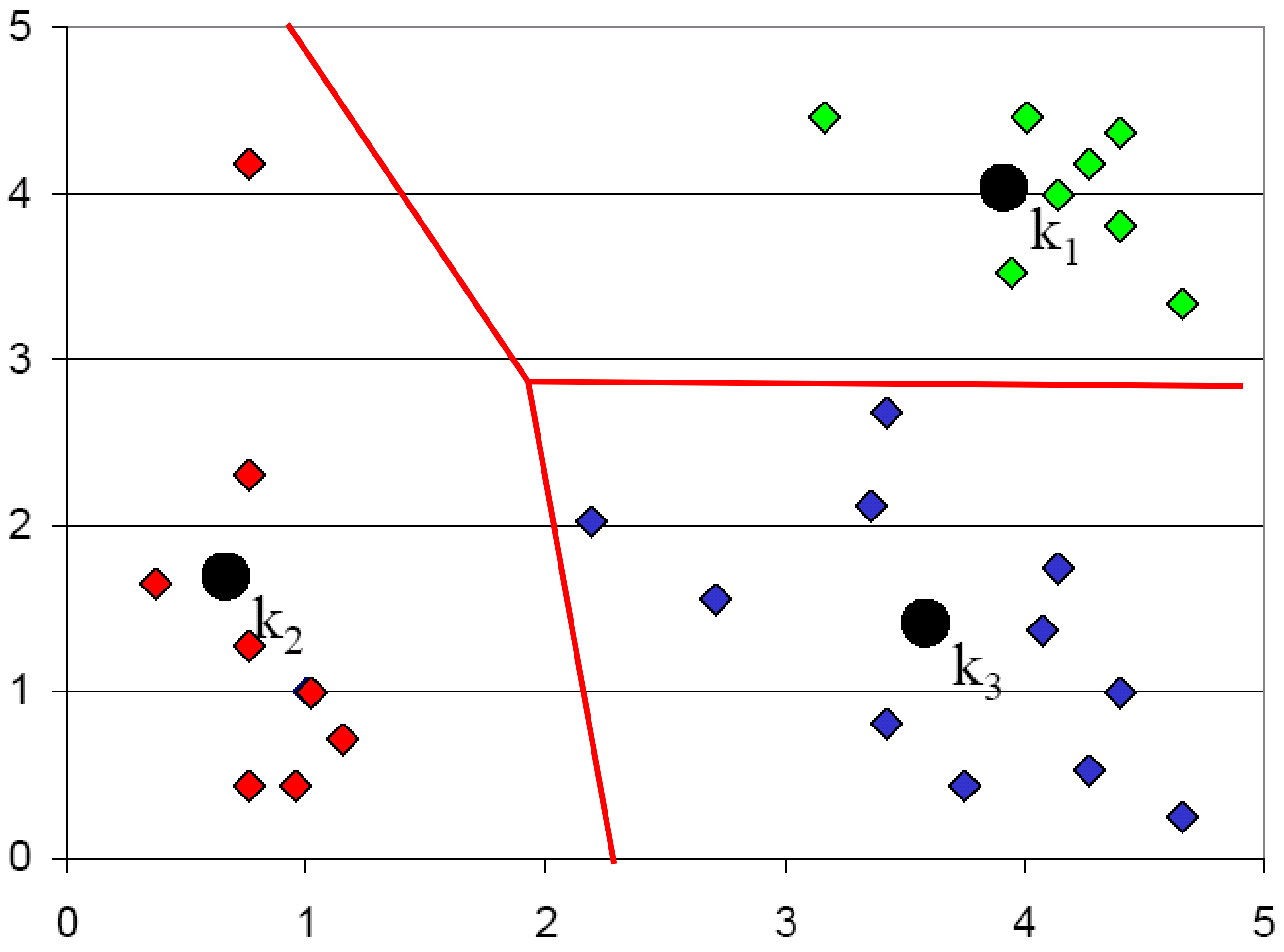
Slide credit: CMU MLD Eric Xing

K-means Clustering: Step 4



Slide credit: CMU MLD Eric Xing

K-means Clustering: Step 5



Slide credit: CMU MLD Eric Xing

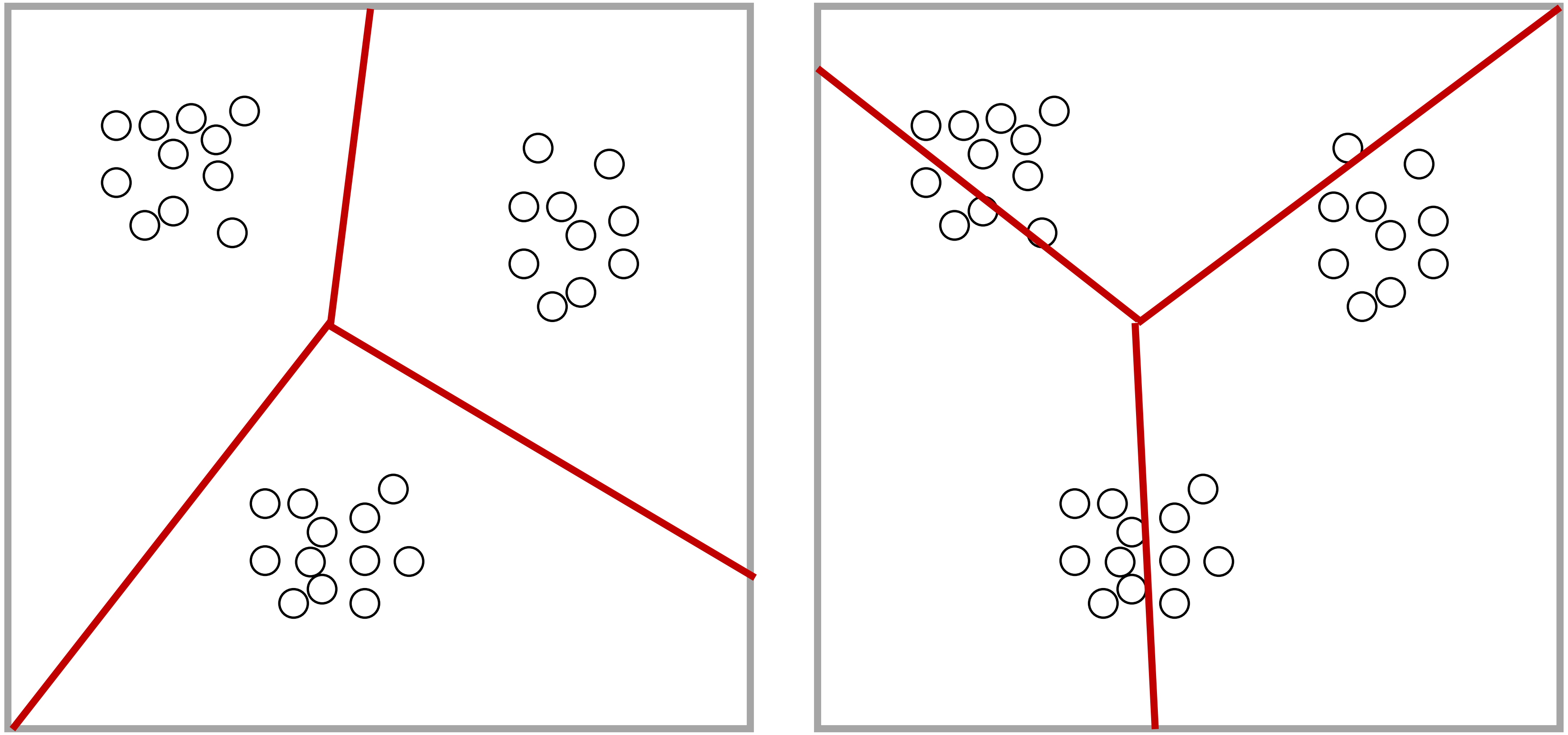
K-means Optimization

Optimization recipe

Slide credit: CMU MLD Matt Gormley

K-means Optimization

Question: Which of these partitions is "better"?



Slide credit: CMU MLD Matt Gormley

K-means Optimization

K-means Optimization

Computational complexity

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

K-means Optimization

Alternating minimization

Slide credit: CMU MLD Matt Gormley

K-means Optimization

Alternating minimization

Alternating minimization

Where have we seen this before?

Recommender Systems

$$\min_{\boldsymbol{U},\boldsymbol{V}} J(\boldsymbol{U},\boldsymbol{V}) \qquad J(\boldsymbol{U},\boldsymbol{V}) = \sum_{i,j\in\mathcal{S}} \left(R_{ij} - \boldsymbol{u}^{(i)^T}\boldsymbol{v}^{(j)}\right)^2$$

① $\min_{U} J(U, \underline{\underline{V}})$ $\qquad U^{t+1} \leftarrow U^{t} - \alpha \nabla_U J$

② $\min_{V} J(\underline{\underline{U}}, V)$ $\qquad V \leftarrow V - \alpha \nabla_V J$

Alternating minimization

Two different approaches

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

Alternating minimization

Two different approaches

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

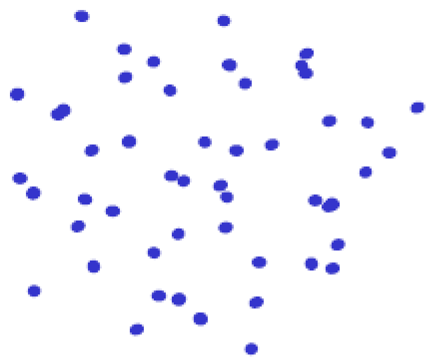
Computational Complexity

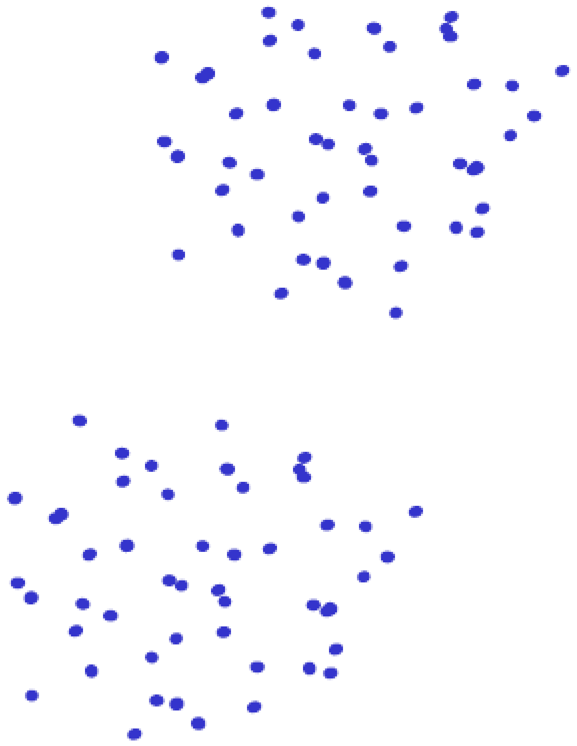
- At each iteration,
  - Computing cluster centers: Each object gets added once to some cluster: O(N)
  - Computing distance between each of the N objects and the K cluster centers is O(KN)

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

Issues: Seed Choice

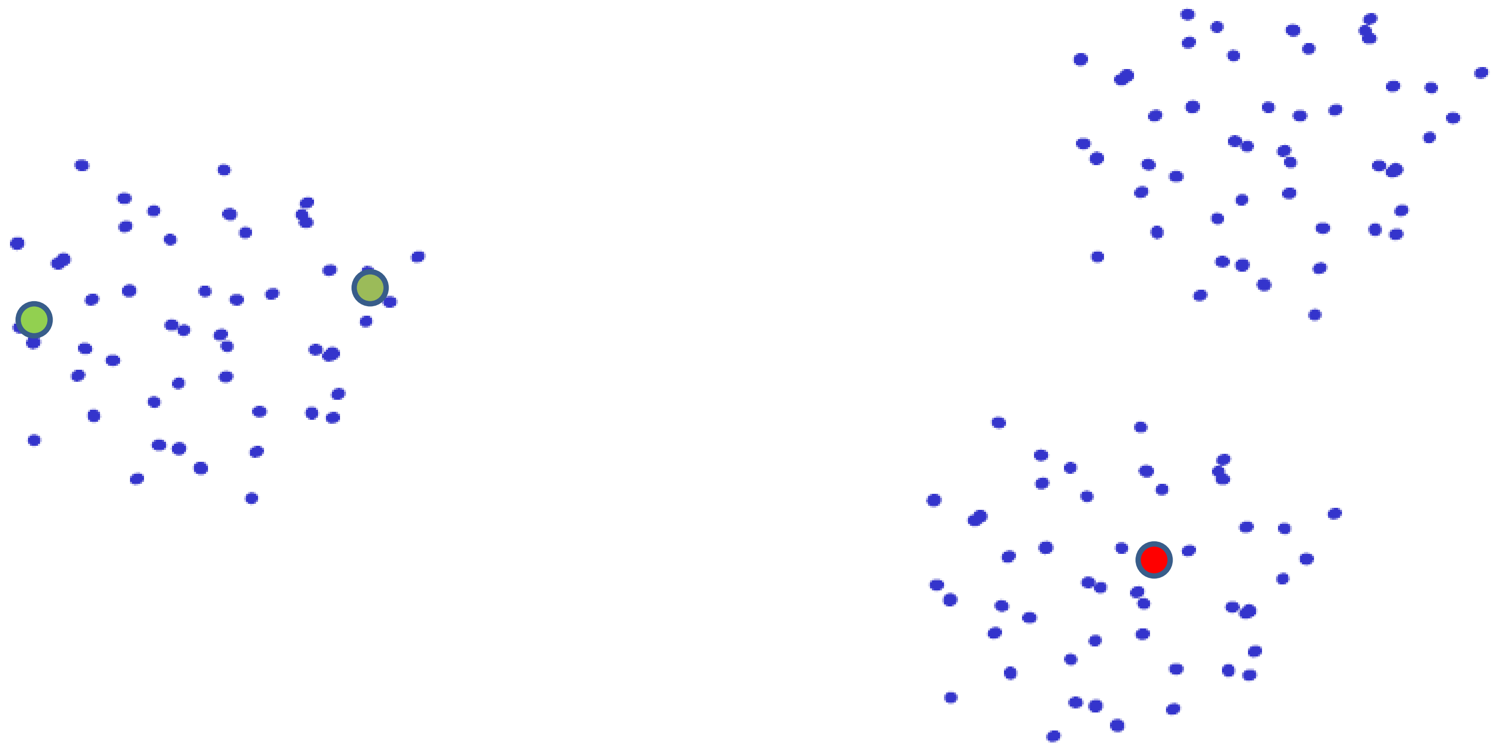
- Results are quite sensitive to seed selection.





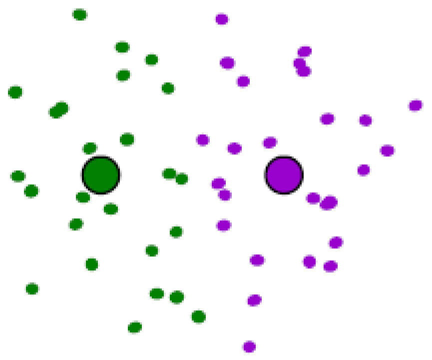
Issues: Seed Choice

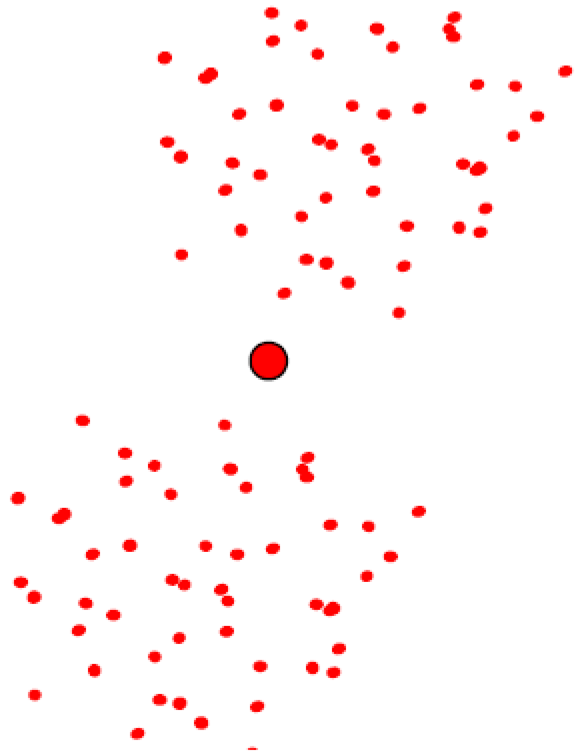
- Results are quite sensitive to seed selection.



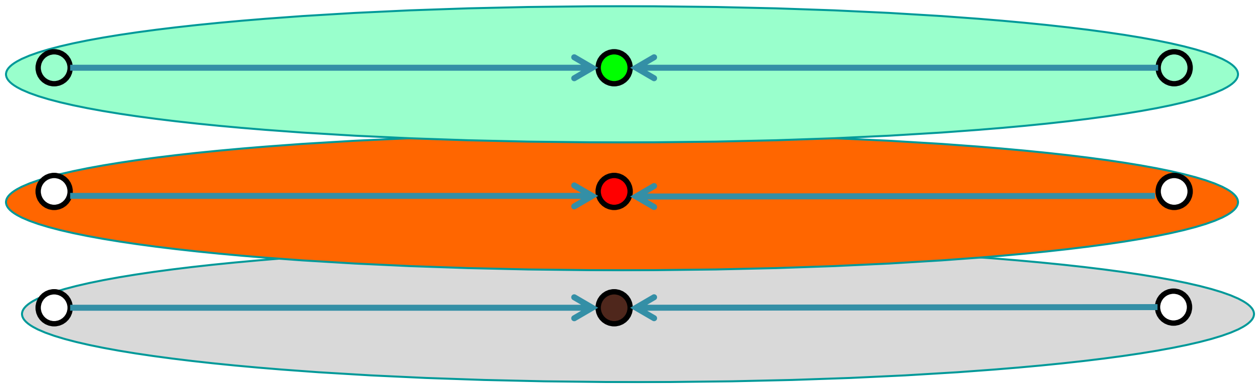
Issues: Seed Choice

- Results are quite sensitive to seed selection.





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

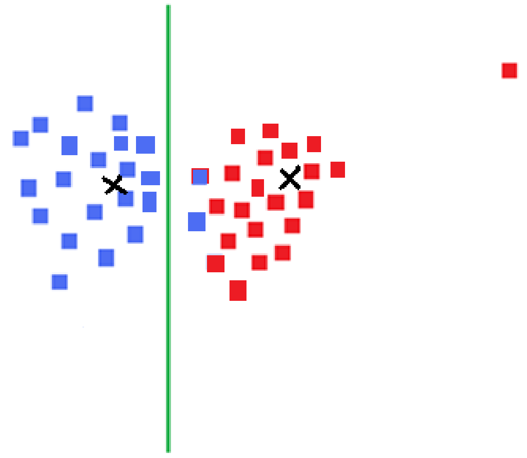
    key idea: choose centers that are far apart

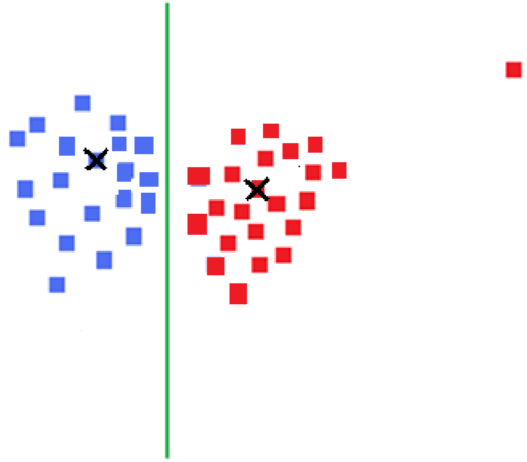
    (probability of picking a point as cluster center $\propto$ distance from nearest center picked so far)

Other Issues

- Shape of clusters
  - Assumes isotropic, equal variance, convex clusters

- Sensitive to Outliers
  - use K-medoids





Slide credit: CMU MLD Aarti Singh

K-medoids

Use actual training point as cluster center (medoid) rather than mean

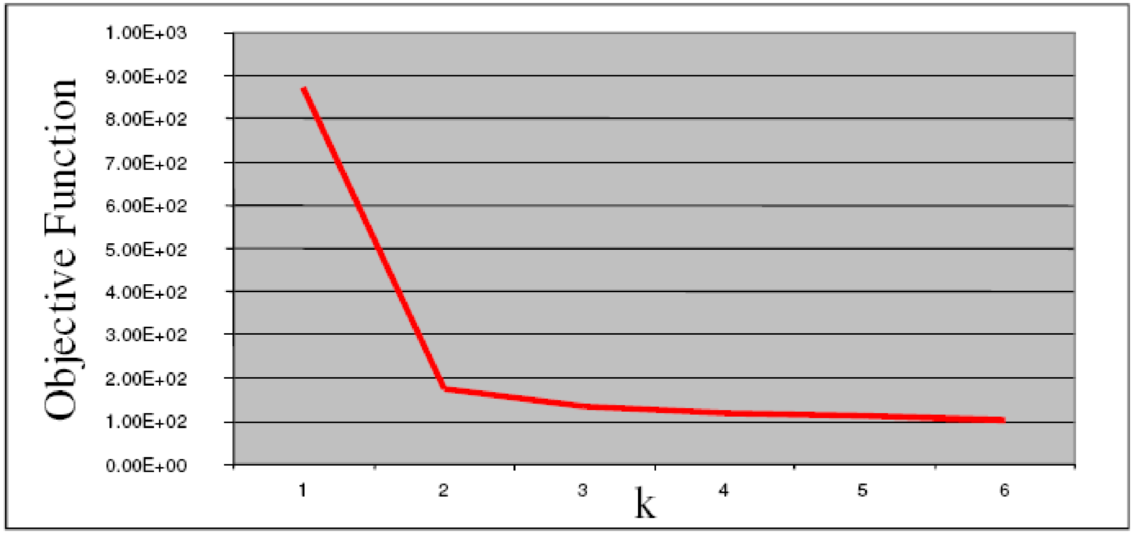
- More robust to outliers pulling the mean away
- Better interpretability, can “visualize” the medoid
- More work to compute medoid than mean

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

K-means algorithm

- Optimize potential function:

$$\min_{\mu} \min_{C} F(\mu, C) = \min_{\mu} \min_{C} \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

- **K-means algorithm:** (coordinate descent on F)

(1) Fix μ, optimize C — **Expected** cluster assignment

(2) Fix C, optimize μ — **Maximum** likelihood for center

Next lecture, we will see a generalization of this approach:

EM algorithm

Slide credit: CMU MLD Aarti Singh