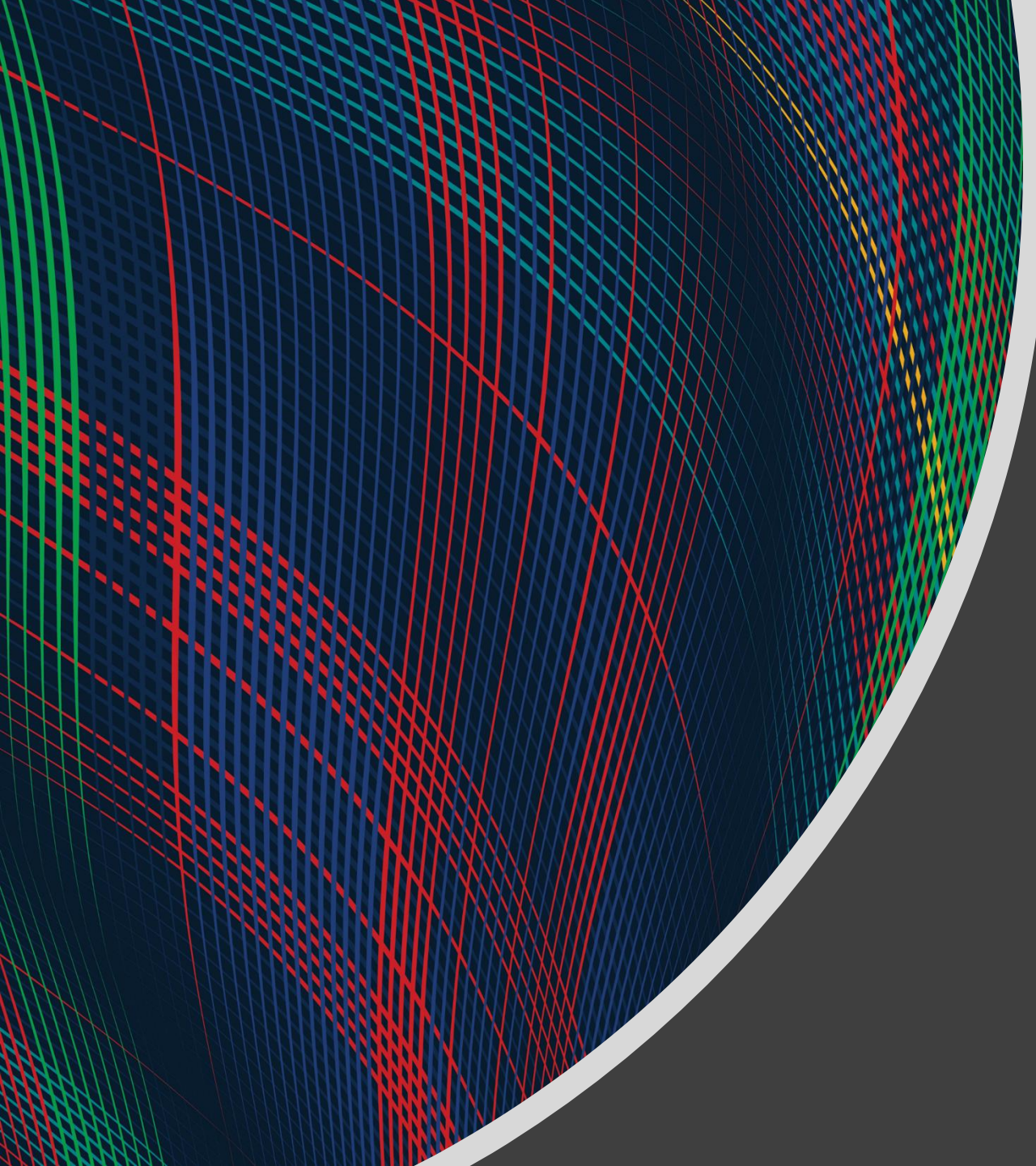


An abstract graphic on the left side of the slide, featuring a sphere-like shape composed of a dense grid of intersecting red, green, and blue lines. The lines are curved and follow the contours of the sphere, creating a complex, woven pattern. The sphere is set against a dark gray background.

10-315
Introduction to ML

Clustering:
K-means

Instructor: Pat Virtue

Learning Paradigms

Paradigm	Data
Supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N \quad \mathbf{x} \sim p^*(\cdot) \text{ and } y = c^*(\cdot)$
$\hookrightarrow$ Regression	$y^{(i)} \in \mathbb{R}$
$\hookrightarrow$ Classification	$y^{(i)} \in \{1, \dots, K\}$
$\hookrightarrow$ 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	$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), (\mathbf{x}^{(3)}, y^{(3)}), \dots\}$
Active Learning	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ and can query $y^{(i)} = c^*(\cdot)$ at a cost
Imitation Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}), (s^{(2)}, a^{(2)}), \dots\}$
Reinforcement Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}, r^{(1)}), (s^{(2)}, a^{(2)}, r^{(2)}), \dots\}$

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:

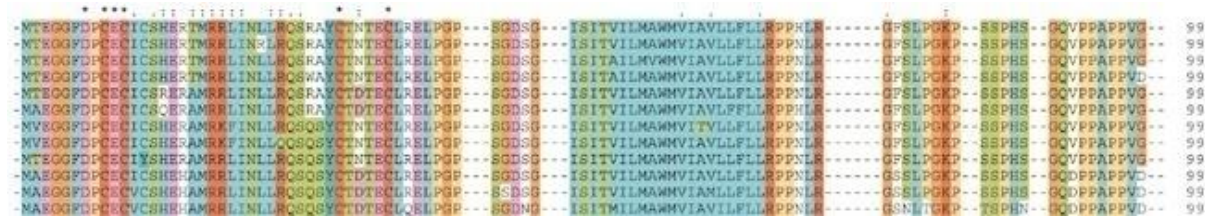
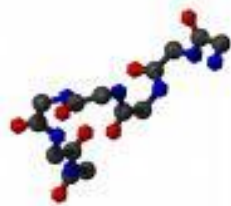
- Automatically organizing data.
- Understanding hidden structure in data.
- Preprocessing for further analysis.
 - Representing high-dimensional data in a low-dimensional space (e.g., for visualization purposes).

Applications (Clustering comes up everywhere...)

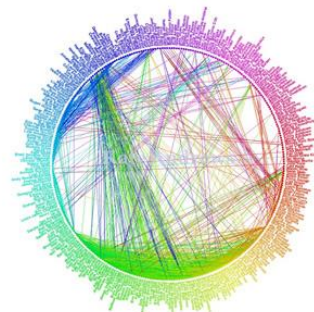
Cluster news articles or web pages or search results by topic.



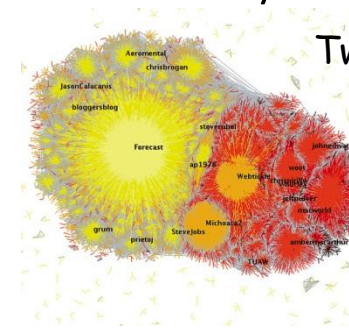
- Cluster protein sequences by function or genes according to expression profile.



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



Facebook network



Twitter Network

Applications (Clustering comes up everywhere...)

Cluster customers according to purchase history.



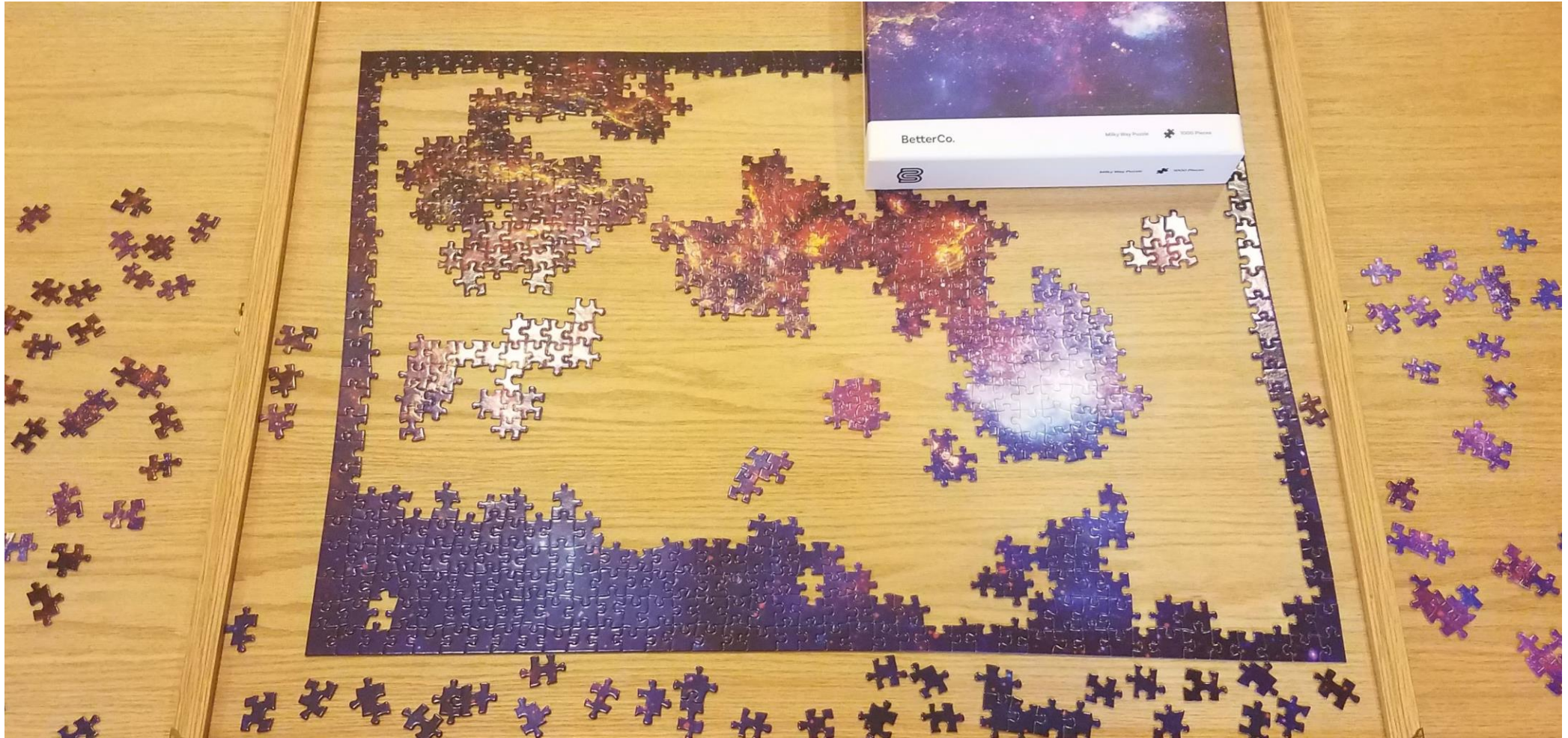
- Cluster galaxies or nearby stars (e.g. Sloan Digital Sky Survey)



- And many many more applications....

Clustering Applications

Jigsaw puzzles!

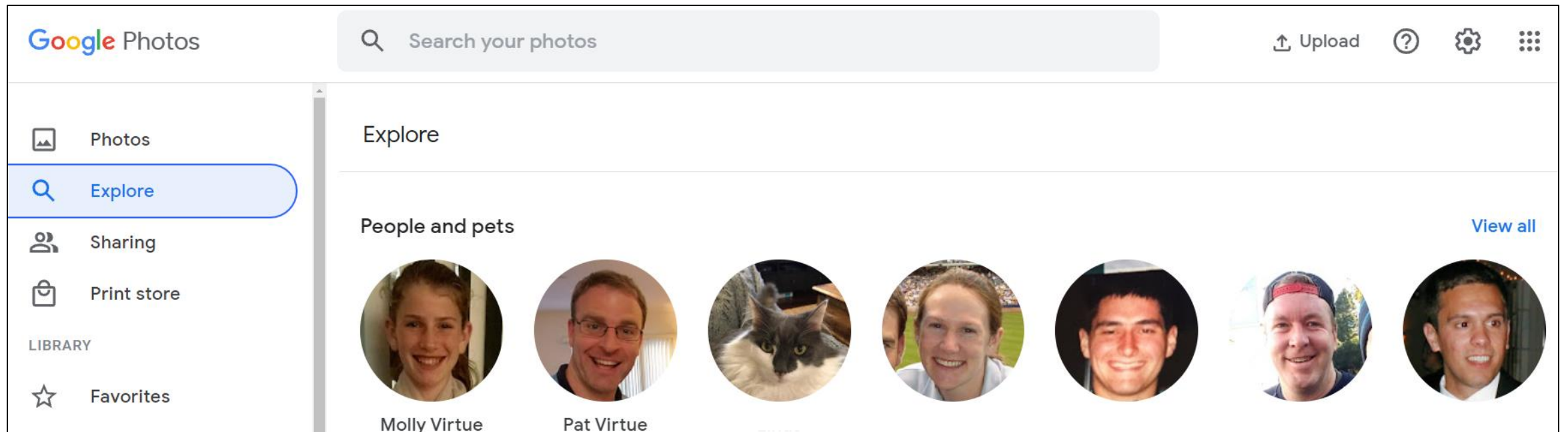


Semi-supervised Learning

Supervised: $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$

Unsupervised: $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$

Semi-supervised: $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N_1} \cup \{\mathbf{x}^{(j)}, y^{(j)}\}_{j=1}^{N_2}$

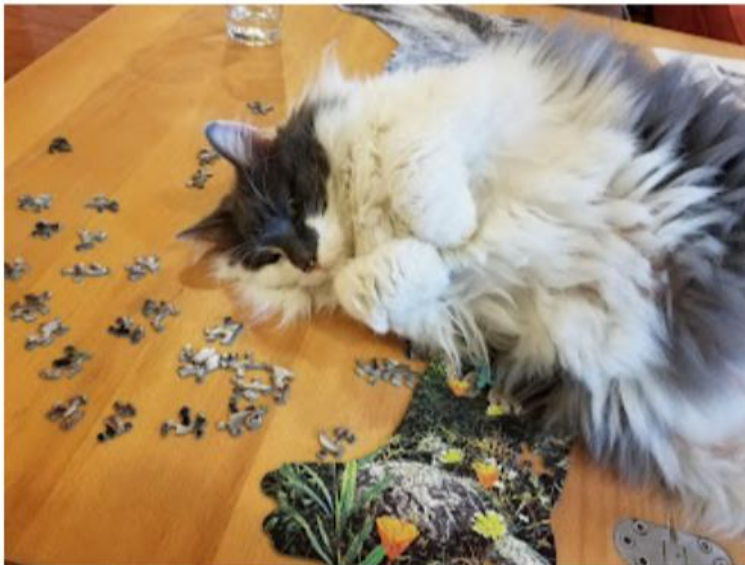
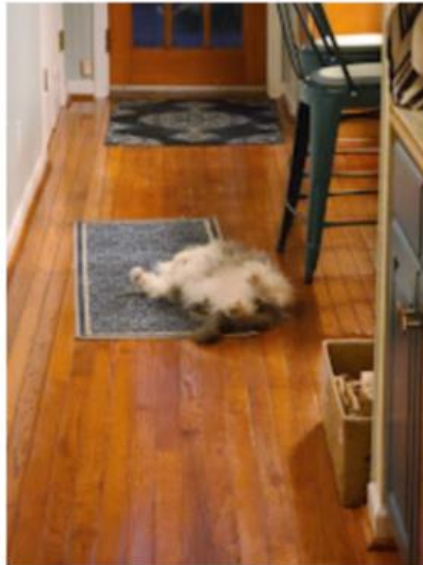


Semi-supervised Learning

Supervised: $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$

Unsupervised: $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$

Semi-supervised: $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N_1} \cup \{\mathbf{x}^{(j)}, y^{(j)}\}_{j=1}^{N_2}$

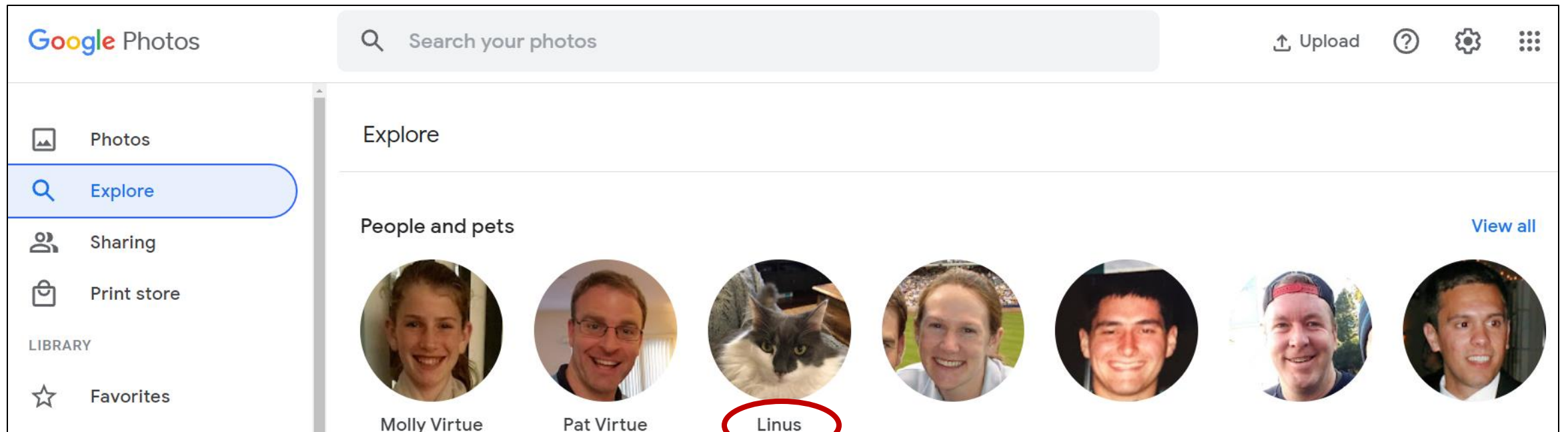


Semi-supervised Learning

Supervised: $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$

Unsupervised: $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$

Semi-supervised: $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N_1} \cup \{\mathbf{x}^{(j)}, y^{(j)}\}_{j=1}^{N_2}$

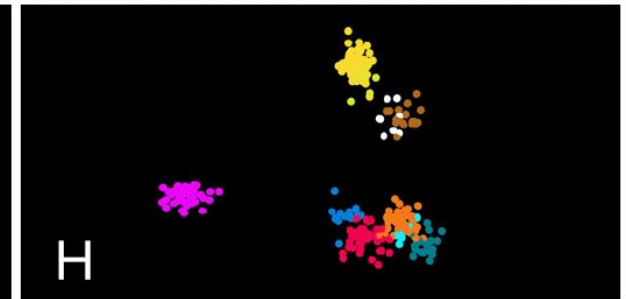
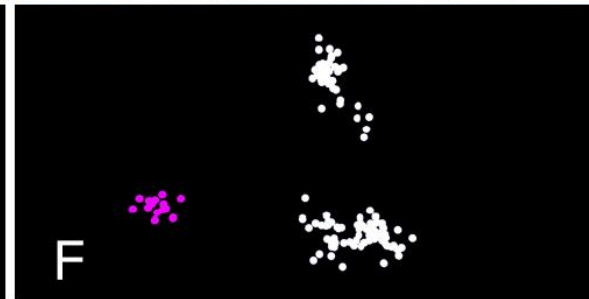


Semi-supervised Learning

Supervised: $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$

Unsupervised: $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$

Semi-supervised: $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N_1} \cup \{\mathbf{x}^{(j)}, y^{(j)}\}_{j=1}^{N_2}$



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

K-Means

Algorithm

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

Initialize – the K cluster centers (randomly if necessary)

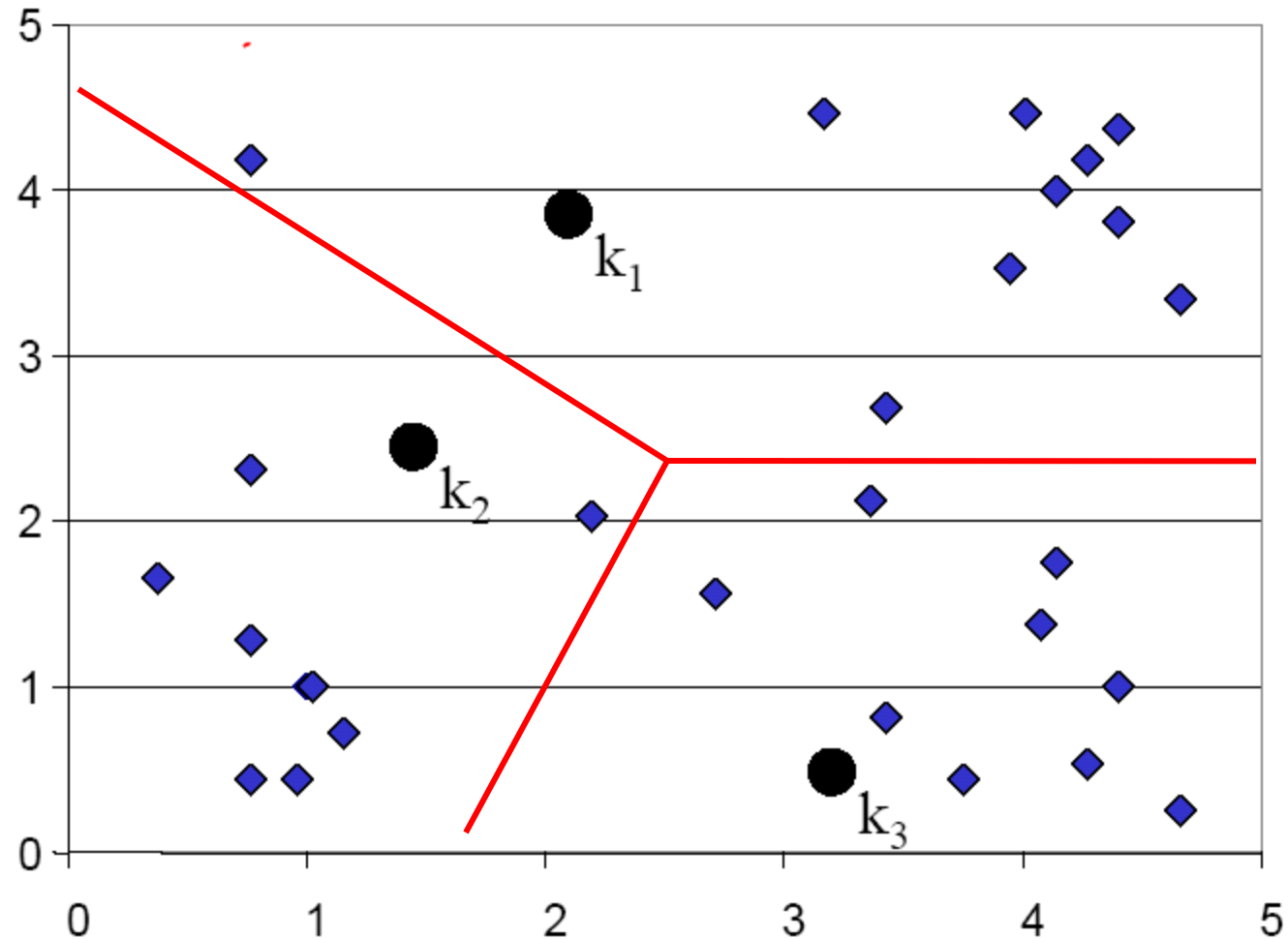
Iterate –

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.

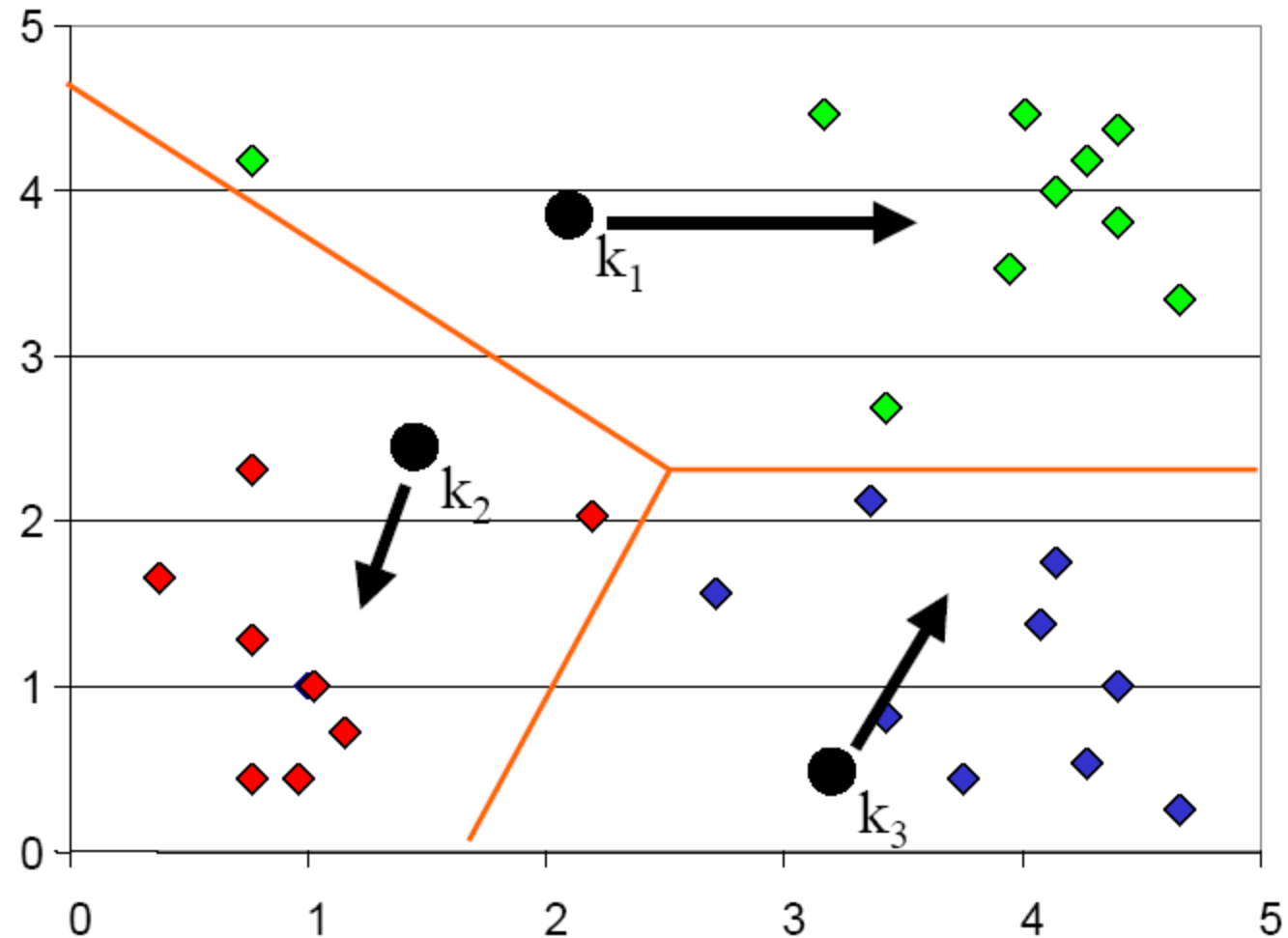
Termination –

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

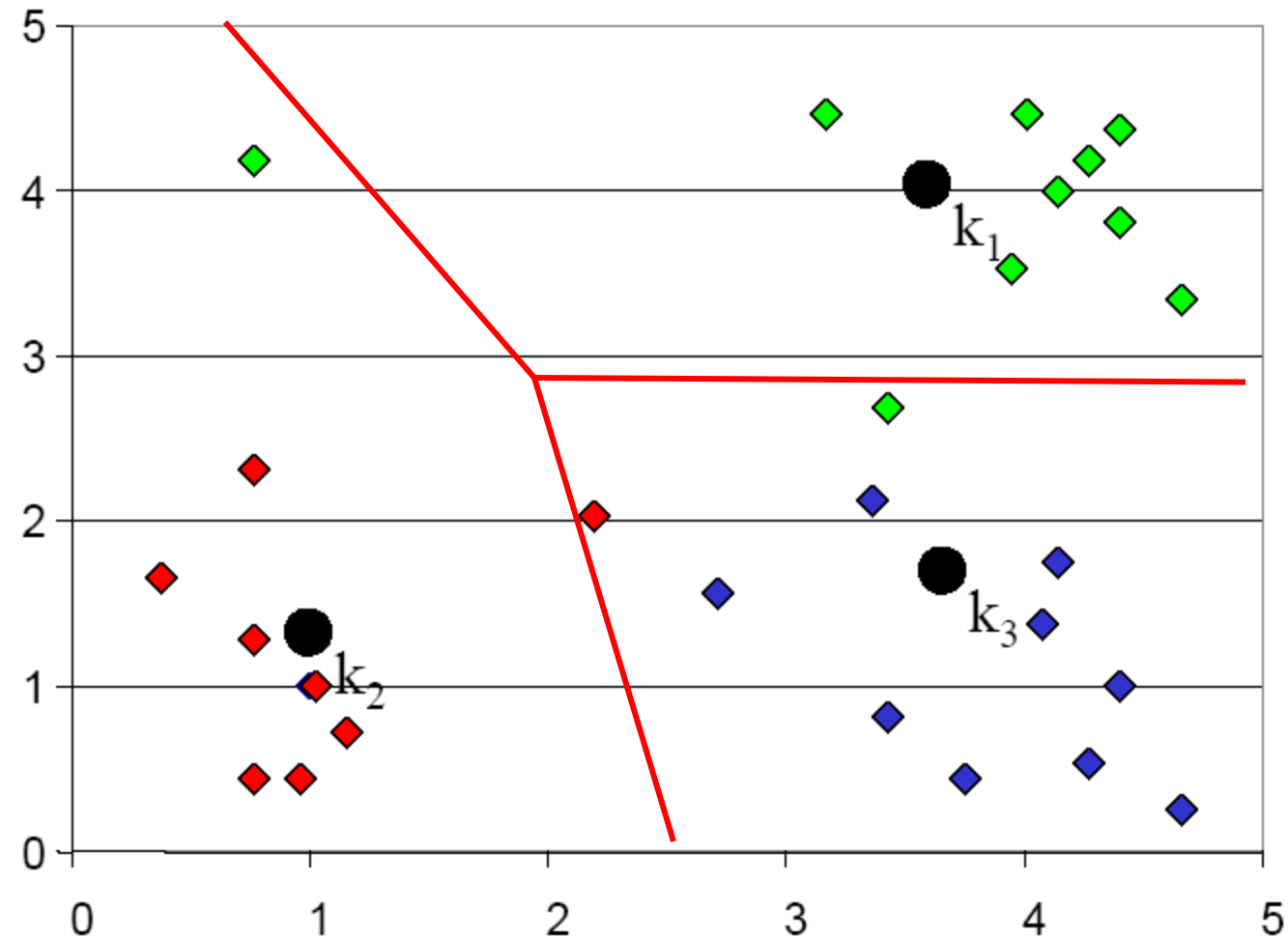
K-means Clustering: Assign points



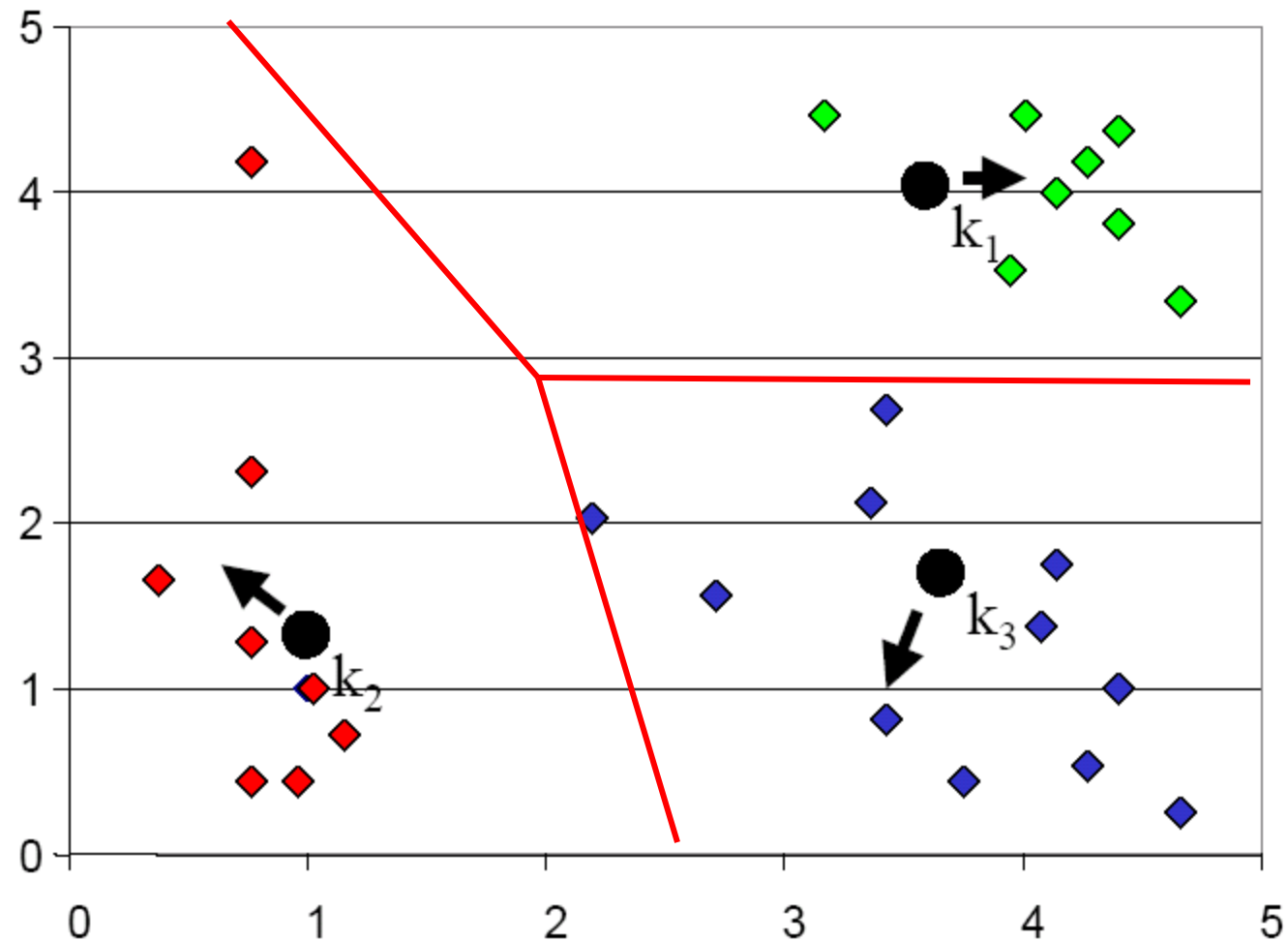
K-means Clustering: Update centers



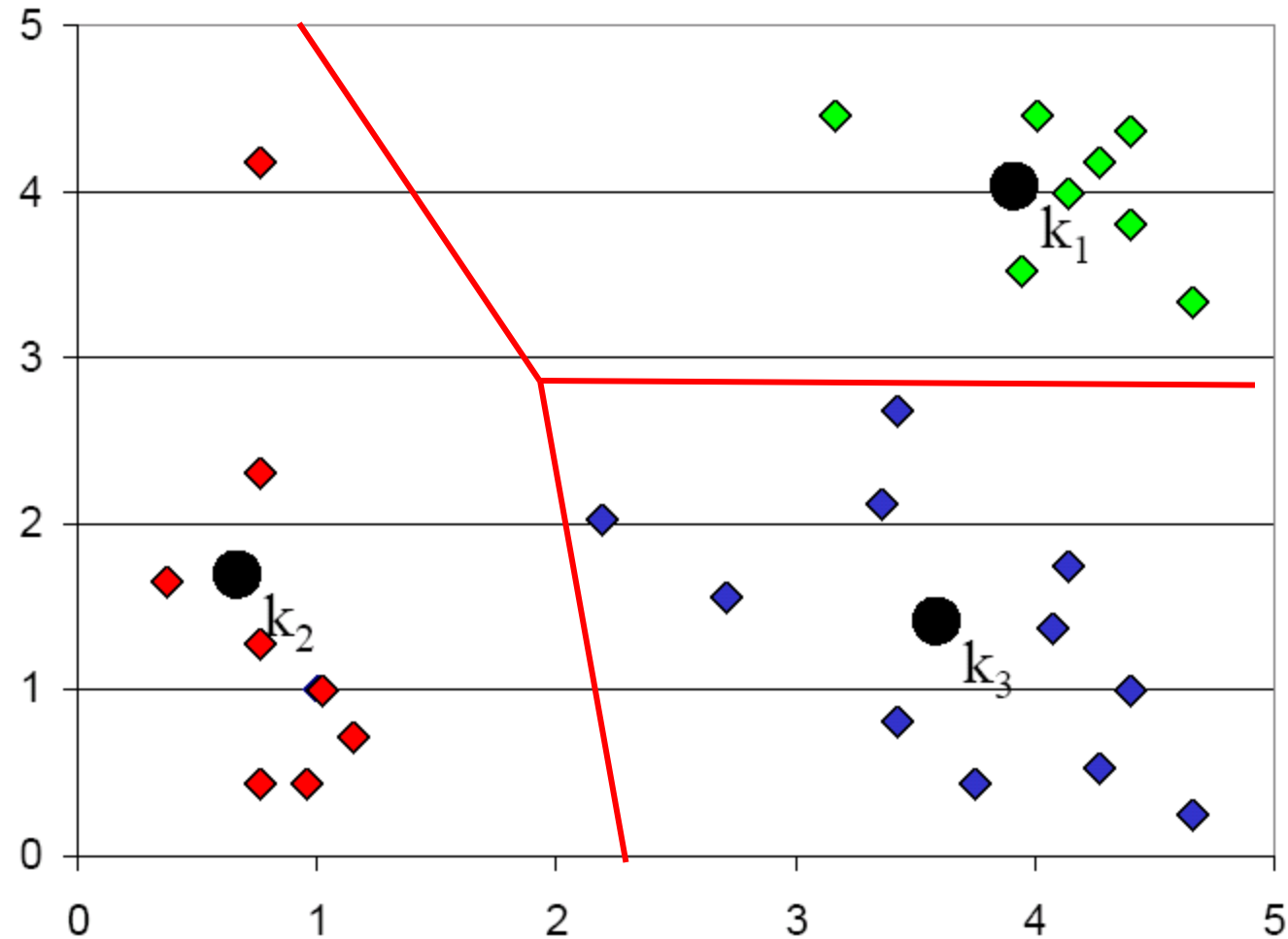
K-means Clustering: Assign points



K-means Clustering: Update centers



K-means Clustering: Assign points



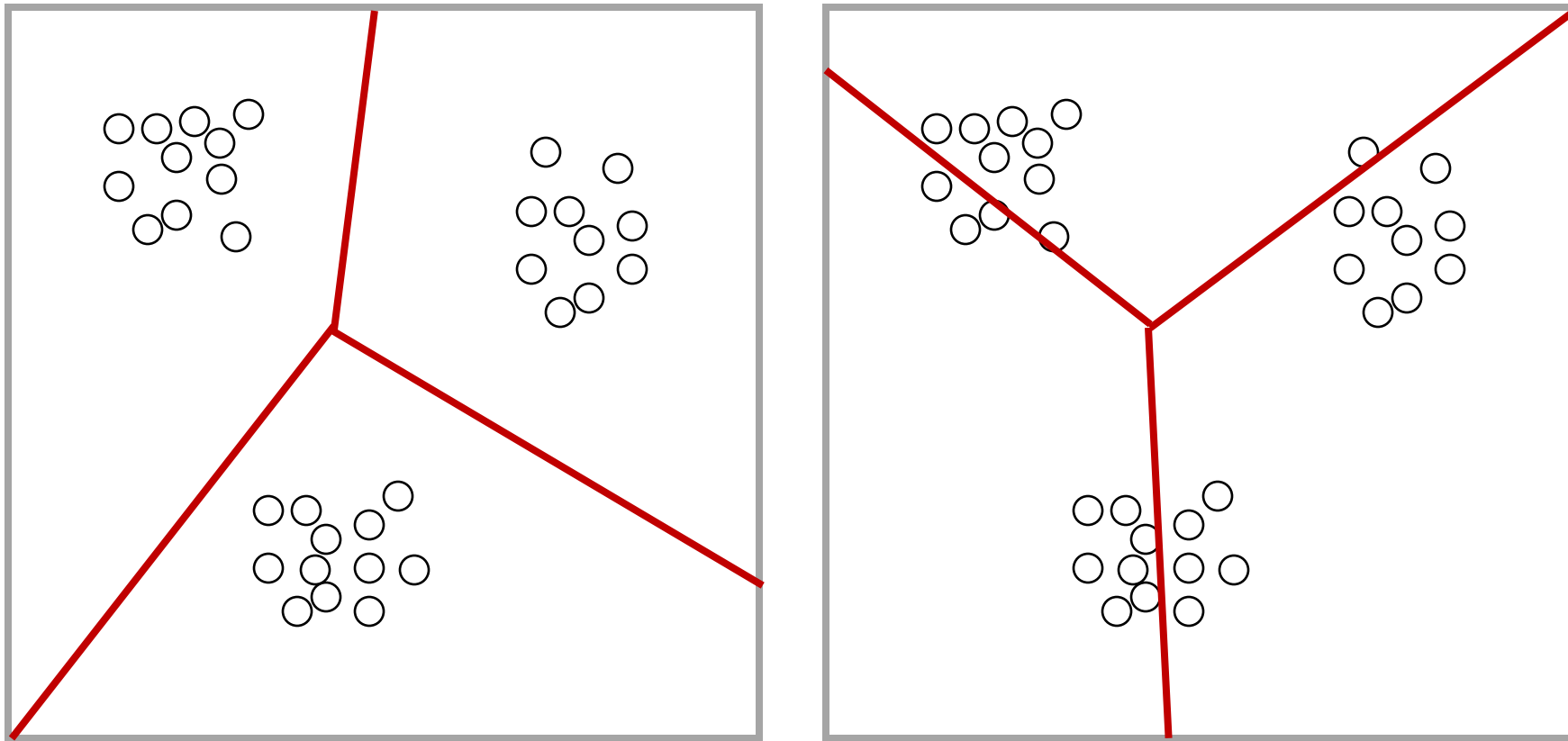
K-means Optimization

Optimization recipe

1. Formulate objective
2. Minimize objective

K-means Optimization

Question: Which of these partitions is “better”?



K-means Optimization

Input: $K, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}, \mathbf{x}^{(i)} \in \mathbb{R}^M$

Num clusters, unlabeled data

Output: $z^{(1)}, \dots, z^{(N)}, z^{(i)} \in \{1 \dots K\}$

Cluster assignments per point

Output: $\mu_1, \dots, \mu_K, \mu_k \in \mathbb{R}^M$

Cluster centers

K-means Optimization

Computational complexity

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

K-means Optimization

Alternating minimization

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

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

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

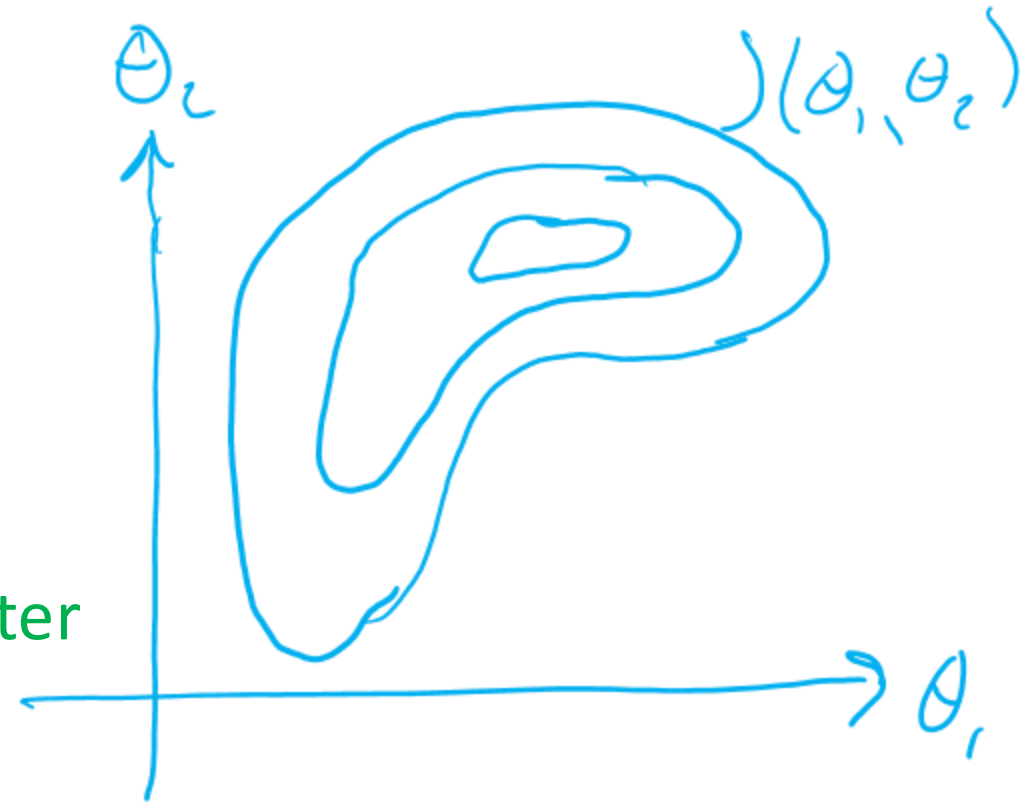
a. $\theta_1 \leftarrow \theta_1 - \eta \partial J / \partial \theta_1$

b. $\theta_2 \leftarrow \theta_2 - \eta \partial J / \partial \theta_2$

2. Find minimum for one parameter

a. $\theta_1 \leftarrow \underset{\theta_1}{\operatorname{argmin}} J(\theta_1, \theta_2)$

b. $\theta_2 \leftarrow \underset{\theta_2}{\operatorname{argmin}} J(\theta_1, \theta_2)$



Alternating minimization

Block coordinate descent

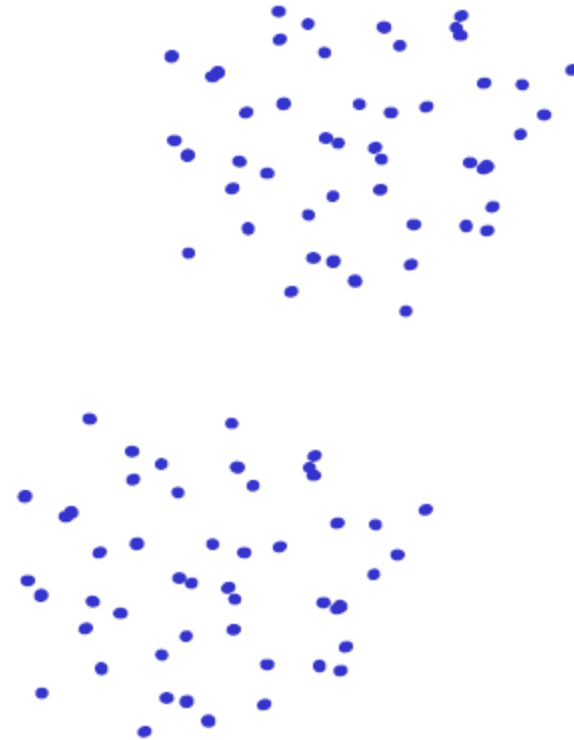
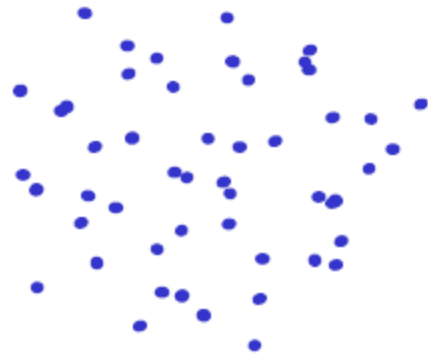
Two different approaches

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

1. Step based on gradient for one set of parameters (step size η)
 - a. $\alpha \leftarrow \alpha - \eta \nabla_{\alpha} J$
 - b. $\beta \leftarrow \beta - \eta \nabla_{\beta} J$
2. Find minimum for one set of parameter (no hyperparameters!)
 - a. $\alpha \leftarrow \operatorname{argmin}_{\alpha} J(\alpha, \beta)$
 - b. $\beta \leftarrow \operatorname{argmin}_{\beta} J(\alpha, \beta)$

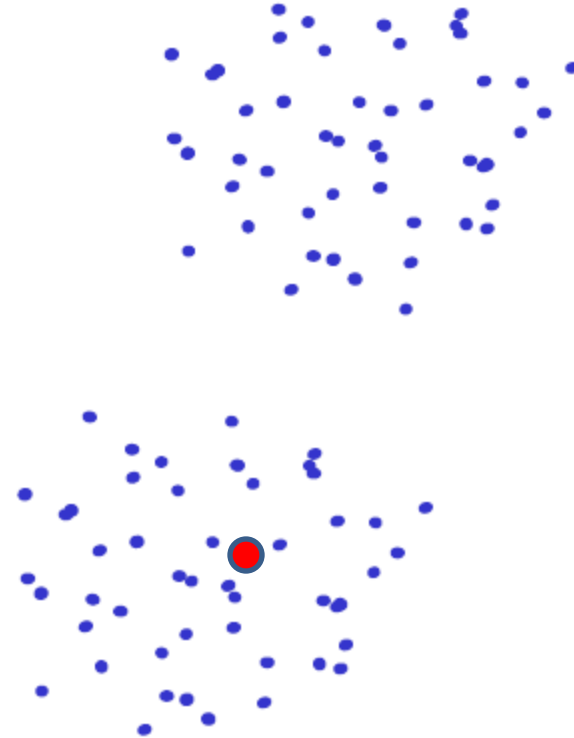
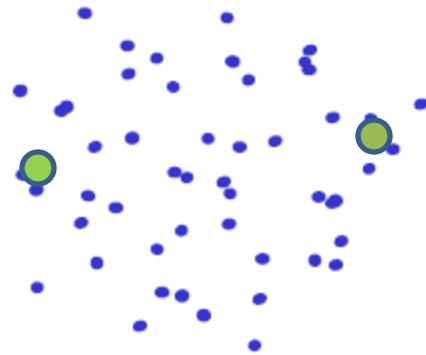
Issues: Seed Choice

- Results are quite sensitive to seed selection.



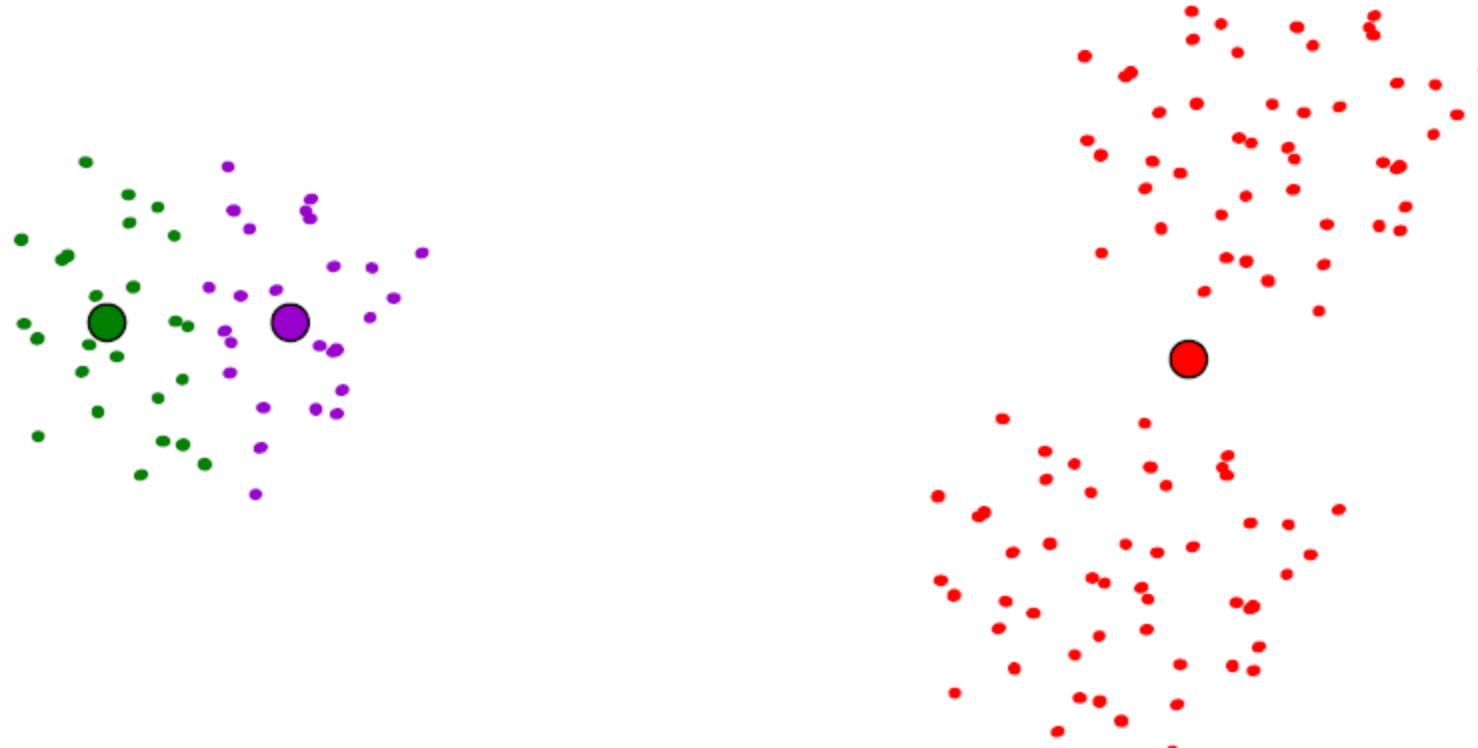
Issues: Seed Choice

- Results are quite sensitive to seed selection.

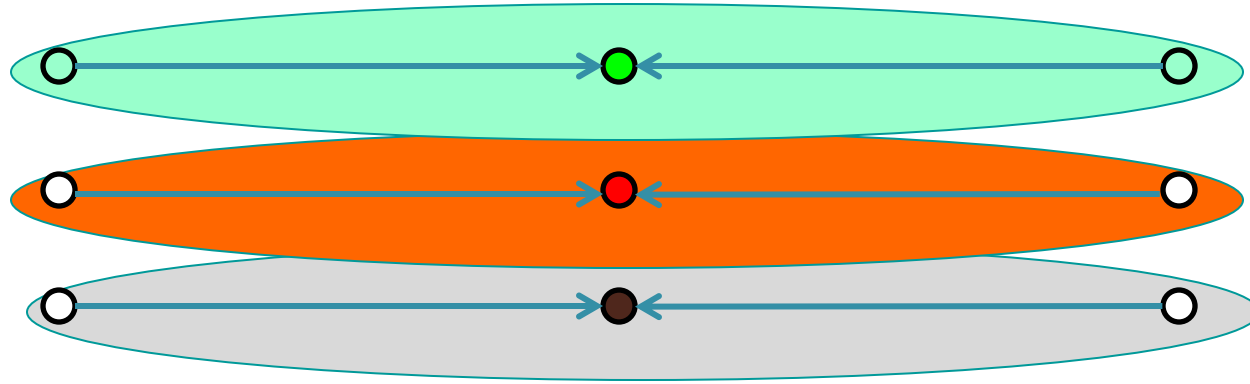


Issues: Seed Choice

- 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)

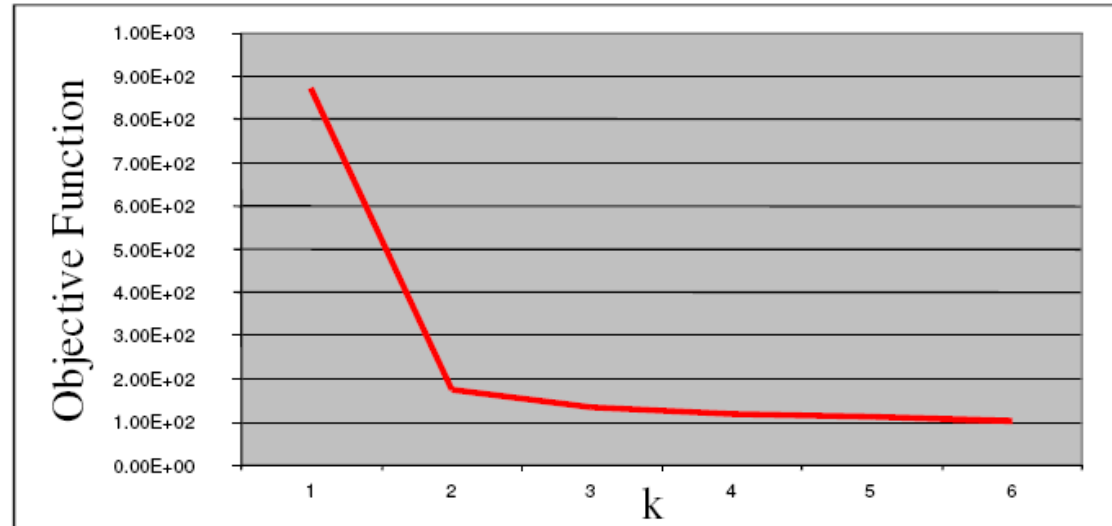
Other Issues

- Number of clusters K

- Objective function

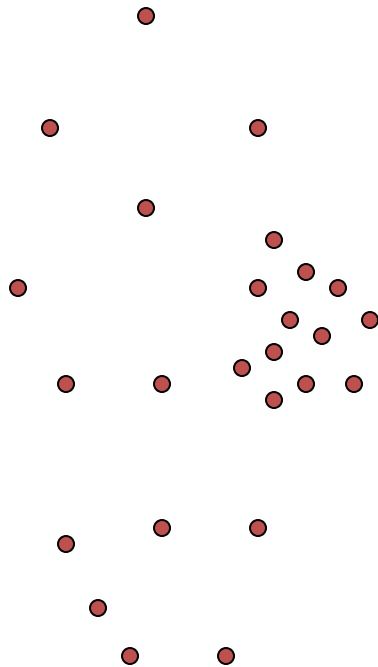
$$\sum_{j=1}^m \|\mu_{C(j)} - x_j\|^2$$

- Look for “Knee” in objective function



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

(One) bad case for K-means



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

Additional Slides

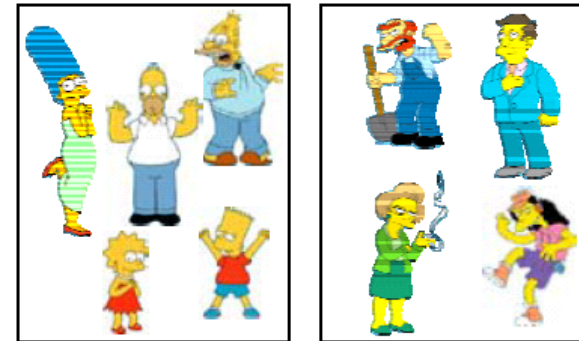
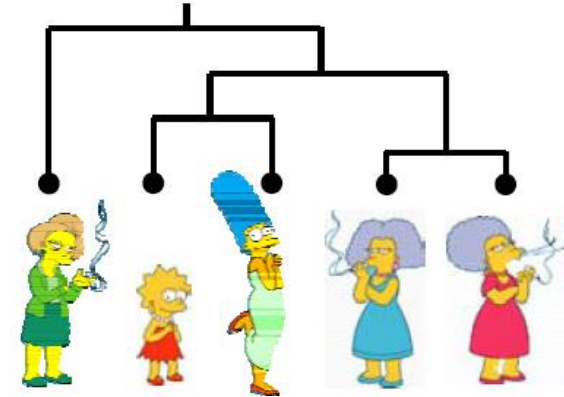
Clustering Algorithms

Hierarchical algorithms

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

Partition algorithms

- K means clustering
- Mixture-Model based clustering



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

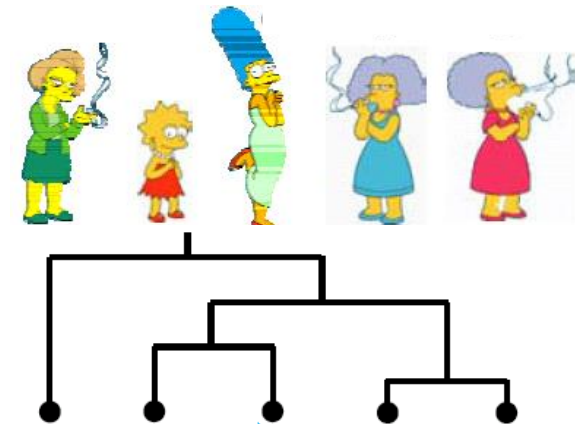


Top-Down divisive

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

