Unsupervised Learning

Reading:

Chapter 8 from *Introduction to Data Mining* by Tan, Steinbach, and Kumar, pp. 487-515, 532-541, 546-552

(http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf)

Unsupervised learning
= No labels on training examples!

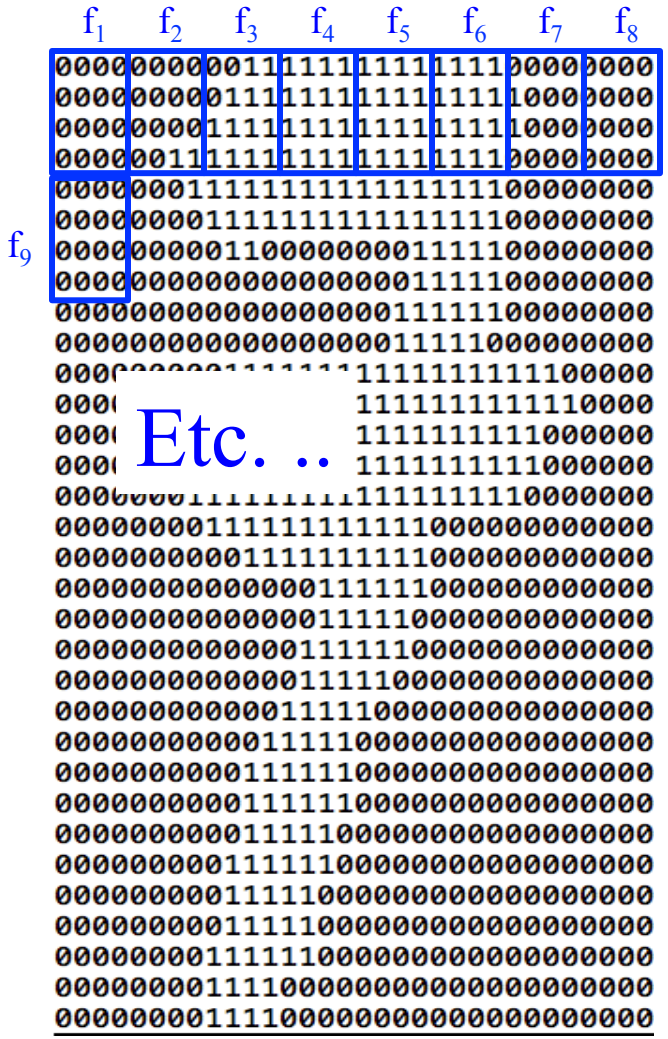
Main approach: Clustering

Examples of possible applications?

Partitional vs. Hierarchical Clustering

Example: Optdigits data set

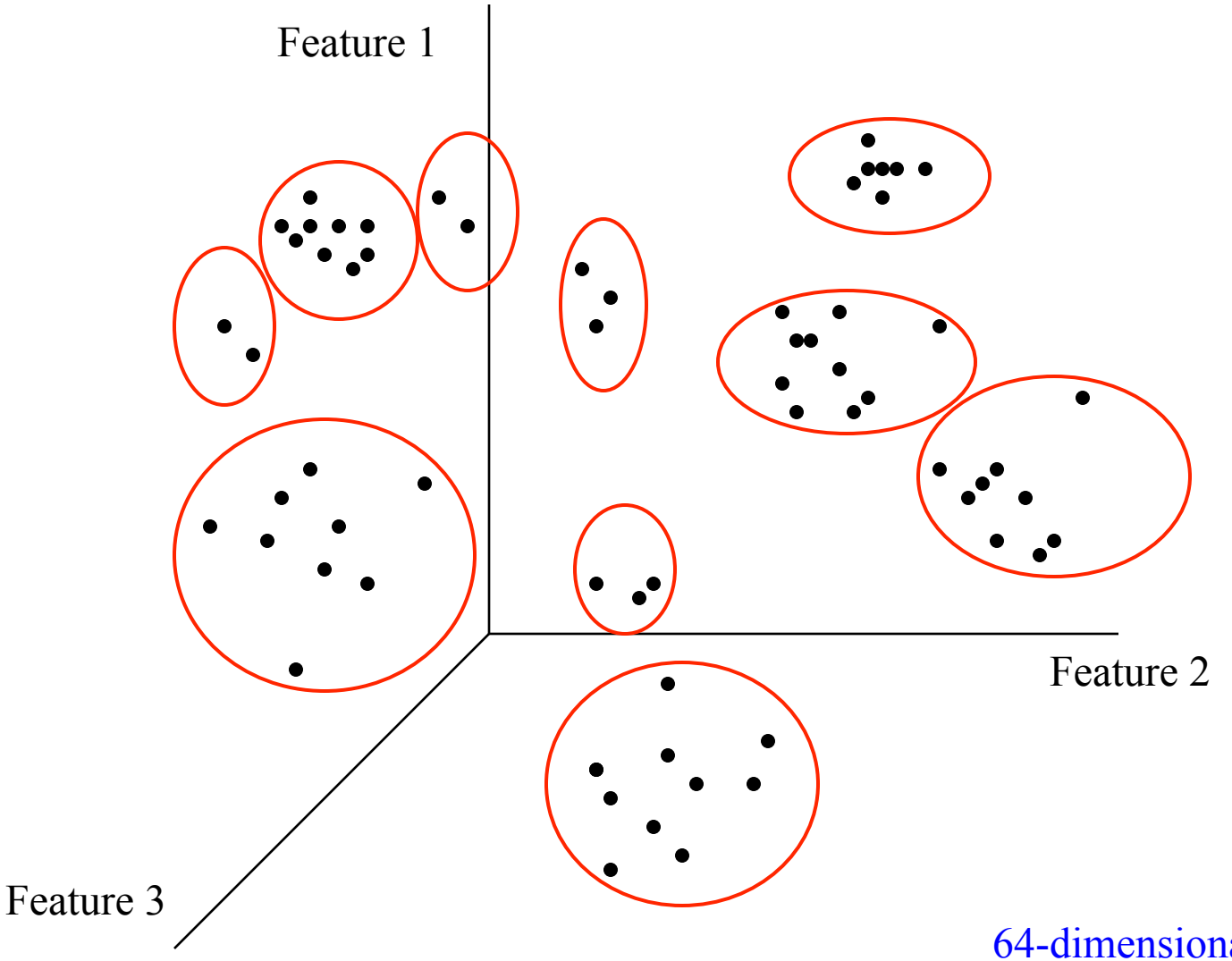
Optdigits features



$\mathbf{x} = (f_1, f_2, ..., f_{64})$

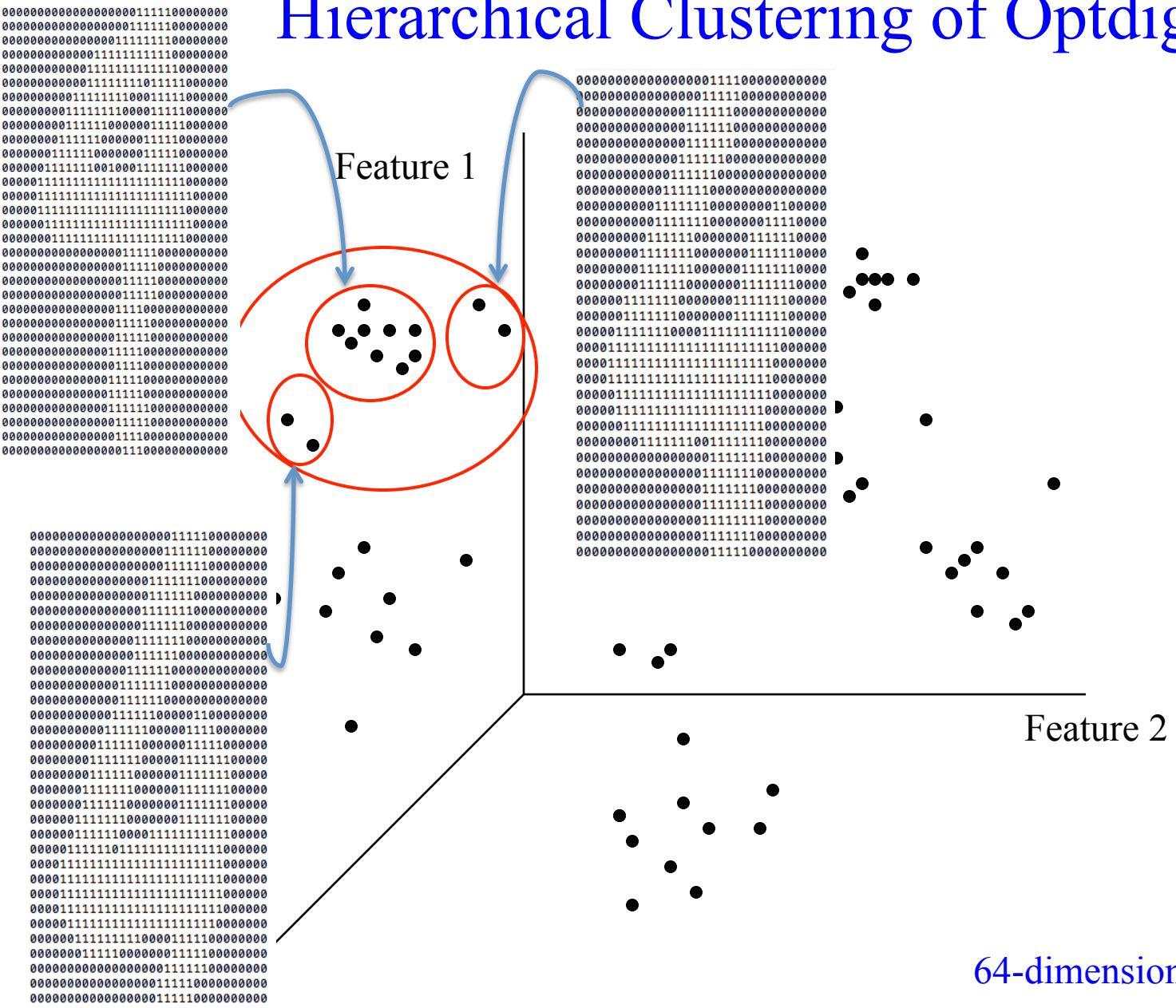
$= (0, 2, 13, 16, 16, 16, 2, 0, 0, ...)$

Partitional Clustering of Optdigits



64-dimensional space

Hierarchical Clustering of Optdigits



64-dimensional space

Issues for clustering algorithms

- How to measure distance between pairs of instances?
- How many clusters to create?
- Should clusters be hierarchical? (I.e., clusters of clusters)
- Should clustering be "soft"? (I.e., an instance can belong to different clusters, with "weighted belonging")

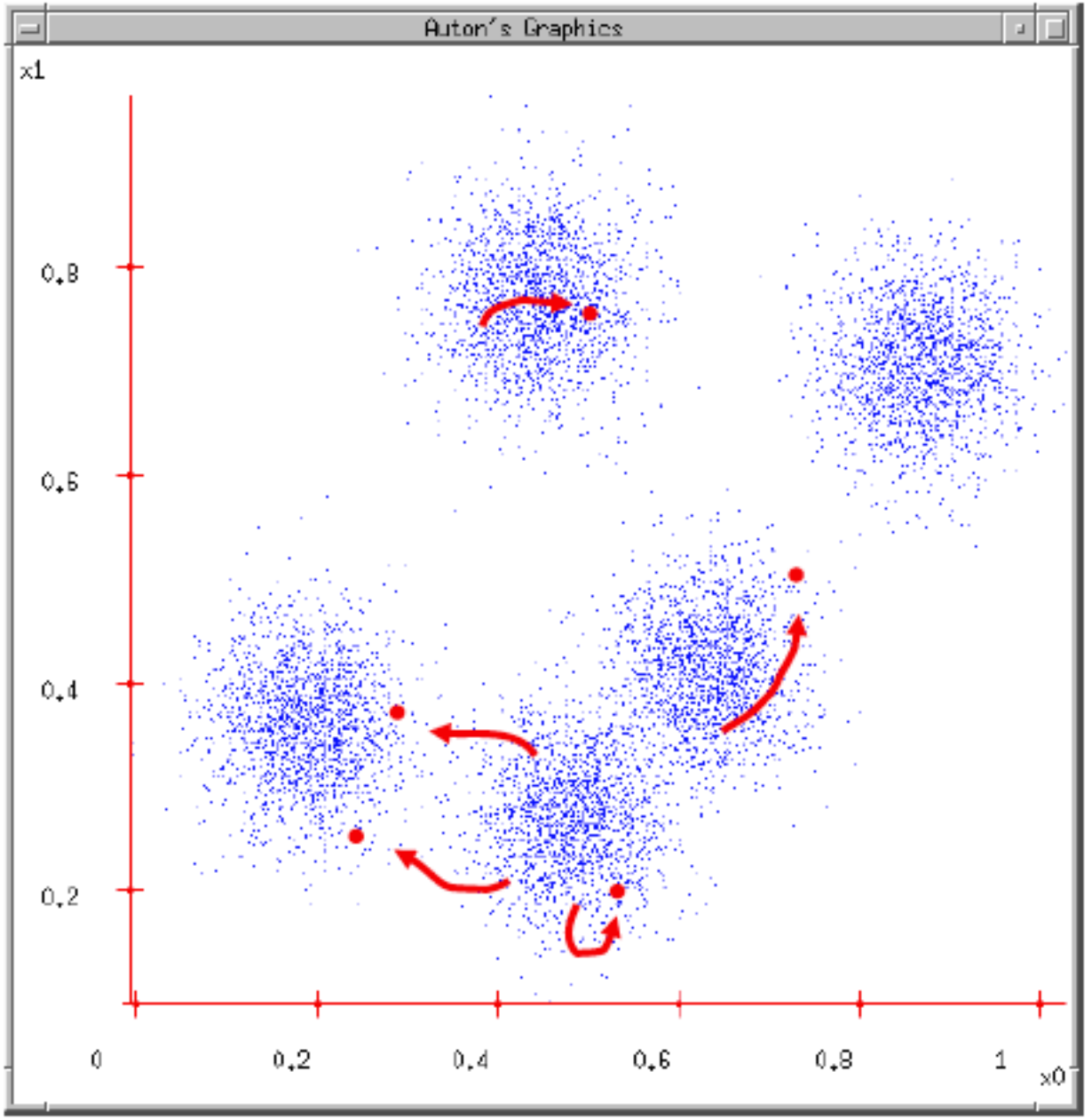
E.g., Consider the problem of "community detection" in social networks

Most commonly used (and simplest) clustering algorithm:

K-Means Clustering

K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns...
5. ...and jumps there
6. ...Repeat until terminated!



K-means clustering algorithm

Algorithm 8.1 Basic K-means algorithm.

1: Select K points as initial centroids.
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: **until** Centroids do not change.

Distance metric: Chosen by user.

For numerical attributes, often use L_2 (Euclidean) distance.

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Stopping/convergence criteria

1. No change of centroids (or minimum change)

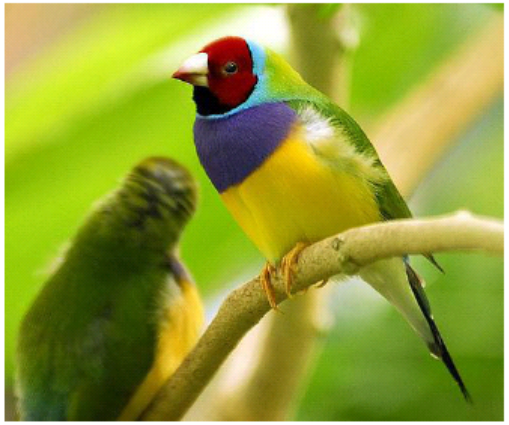
2. No (or minimum) decrease in the **sum squared error** (SSE),

$$SSE = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} d(\mathbf{x}, \mathbf{m}_i)^2$$

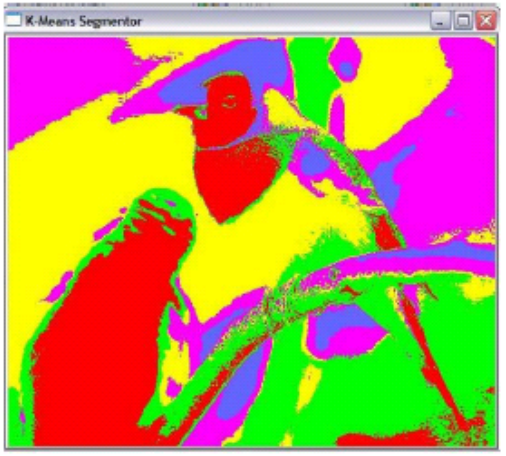
where C_i is the ith cluster, $\mathbf{m}_i$ is the centroid of cluster C_i (the mean vector of all the data points in C_i), and $d(\mathbf{x}, \mathbf{m}_i)$ is the distance between data point $\mathbf{x}$ and centroid $\mathbf{m}_i$.

Example: Image segmentation by *K*-means clustering by color

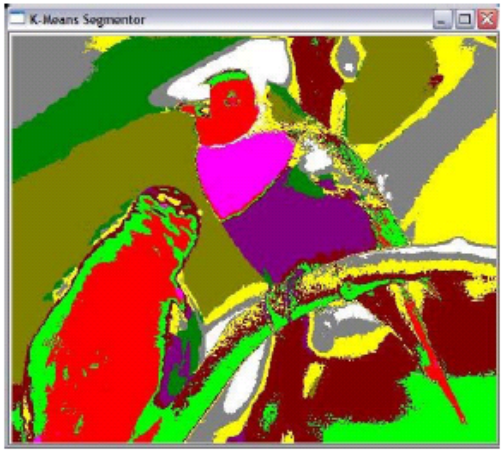
From http://vitroz.com/Documents/Image%20Segmentation.pdf



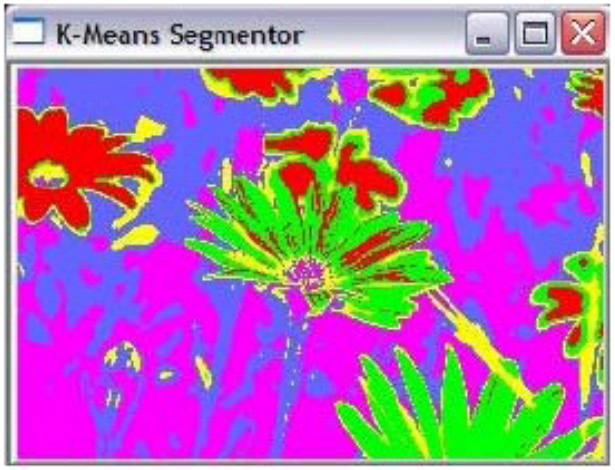
K=5, RGB space



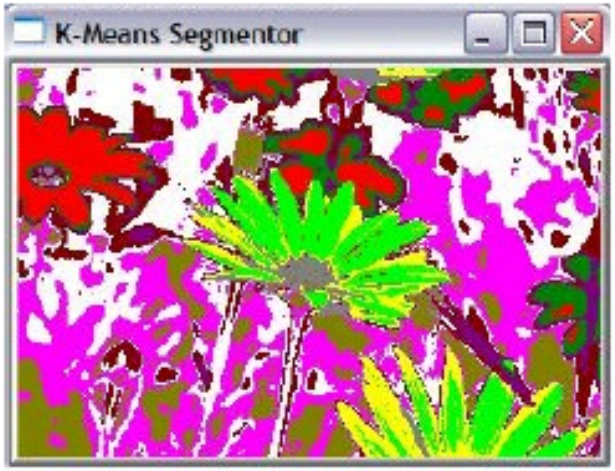
K=10, RGB space



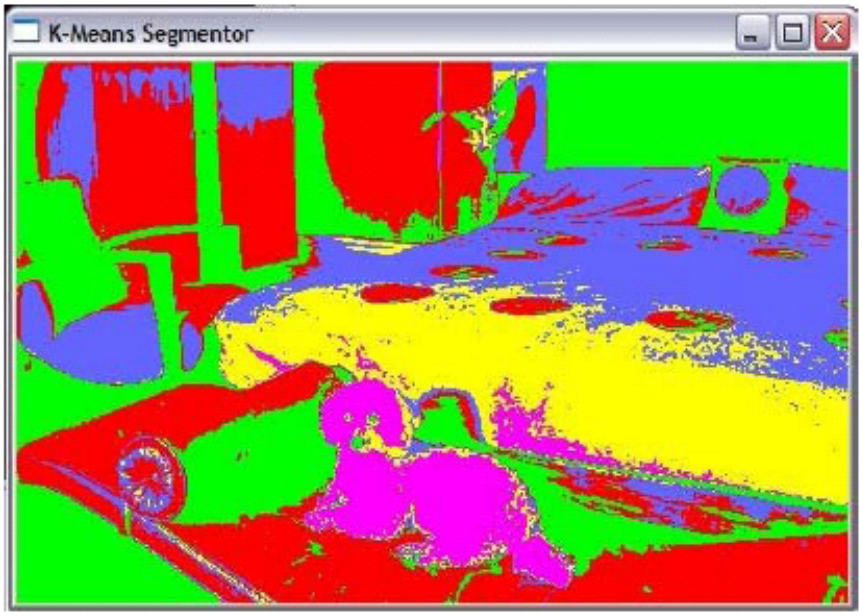
K=5, RGB space



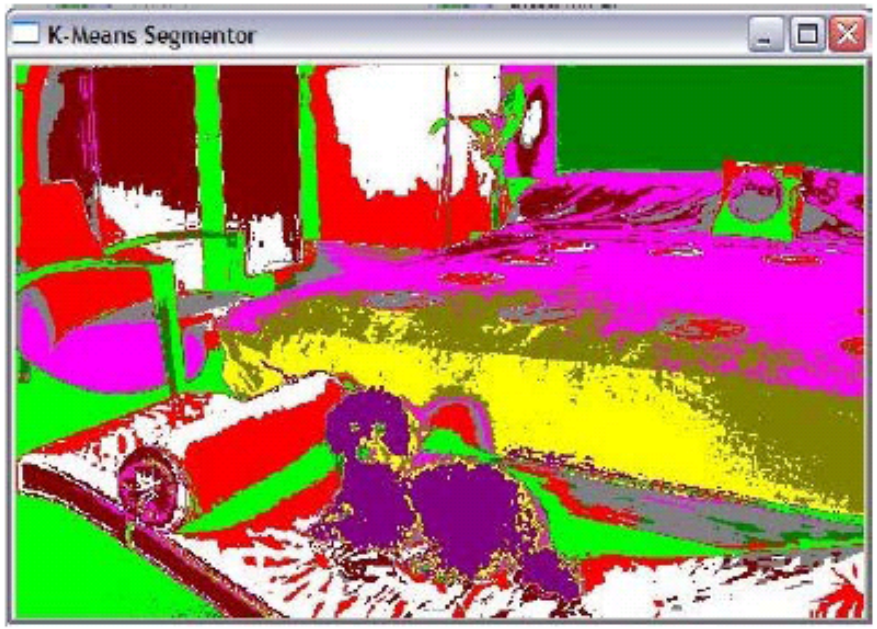
K=10, RGB space



K=5, RGB space



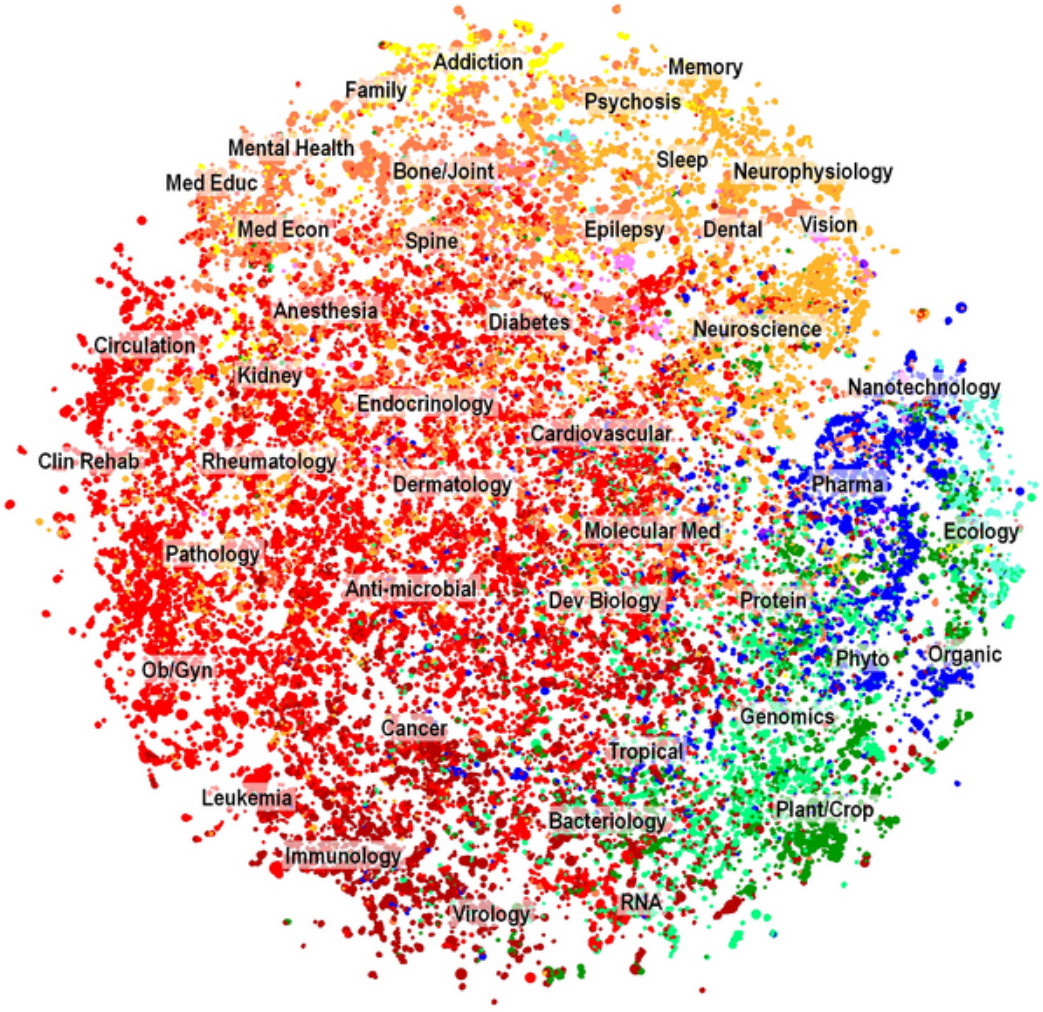
K=10, RGB space



Clustering text documents

- A text document is represented as a feature vector of word frequencies
- Distance between two documents is the cosine of the angle between their corresponding feature vectors.

Figure 4. Two-dimensional map of the PMRA cluster solution, representing nearly 29,000 clusters and over two million articles.



Boyack KW, Newman D, Duhon RJ, Klavans R, et al. (2011) Clustering More than Two Million Biomedical Publications: Comparing the Accuracies of Nine Text-Based Similarity Approaches. PLoS ONE 6(3): e18029. doi:10.1371/journal.pone.0018029
http://www.plosone.org/article/info:doi/10.1371/journal.pone.0018029

In-Class Exercise 1 (a)-(b)

How to evaluate clusters produced by *K-means*?

- Unsupervised evaluation

- Supervised evaluation

Unsupervised Cluster Evaluation

We don't know the classes of the data instances

We want to minimize internal coherence of each cluster – i.e., minimize SSE.

We want to maximize pairwise separation of each cluster – i.e.,

$$Sum\ Squared\ Separation\ (clustering) = \sum_{\text{all distinct pairs of clusters } i,\ j\ (i \neq j)} \mathbf{d}(m_i, m_j)^2$$

Supervised Cluster Evaluation

Suppose we know the classes of the data instances

Entropy of a cluster: The degree to which a cluster consists of objects of a single class.

$$entropy(C_i) = -\sum_{j=1}^{|Classes|} p_{i,j} \log_2 p_{i,j}$$

where

$p_{i,j}$ = probability that a member of cluster i belongs to class j

$= \frac{m_{i,j}}{m_i}$, where $m_{i,j}$ is the number of instances in cluster i with class j

and m_i is the number of instances in cluster i

Mean entropy of a clustering: Average entropy over all clusters in the clustering

$$mean\ entropy(Clustering) = \sum_{1}^{K} \frac{m_i}{m}\ entropy(C_i)$$

where m_i is the number of instances in cluster i

and m is the total number of instances in the clustering.

We want to minimize mean entropy

Entropy Example

Suppose there are 3 classes: 1, 2, 3

Cluster 1	Cluster2	Cluster3
1 2 1 3 1 1 3	2 3 3 3 2 3	1 1 3 2 2 3 2

$$entropy(C_1) = -\left(\frac{4}{7}\log_2\frac{4}{7} + \frac{1}{7}\log_2\frac{1}{7} + \frac{2}{7}\log_2\frac{2}{7}\right) = 1.37$$

$$entropy(C_2) = -\left(0 + \frac{2}{6}\log_2\frac{2}{6} + \frac{4}{6}\log_2\frac{4}{6}\right) = 0.91$$

$$entropy(C_3) = -\left(\frac{2}{7}\log_2\frac{2}{7} + \frac{3}{7}\log_2\frac{3}{7} + \frac{2}{7}\log_2\frac{2}{7}\right) = 1.54$$

$$mean\ entropy(Clustering) = \frac{7}{20}(1.37) + \frac{6}{20}(0.91) + \frac{7}{20}(1.54)$$

In-Class Exercises (c)-(d)

K-Means Variants

- *K*-means often leads to local optimum (with respect to SSE).

- The reading details several variants of K-means, or post-processing techniques, that can improve on the local optimum

Adapted from Bing Liu, UIC
http://www.cs.uic.edu/~liub/teach/cs583-fall-05/CS583-unsupervised-learning.ppt

Issues for *K*-means

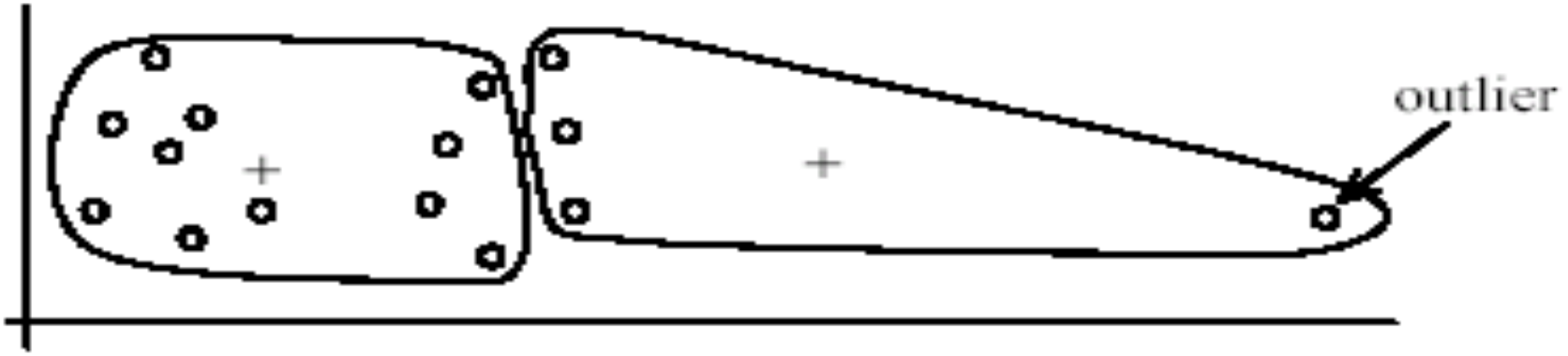
- The algorithm is only applicable if the mean is defined.
  - For categorical data, use *K*-modes: The centroid is represented by the most frequent values.

- The user needs to specify *K*.

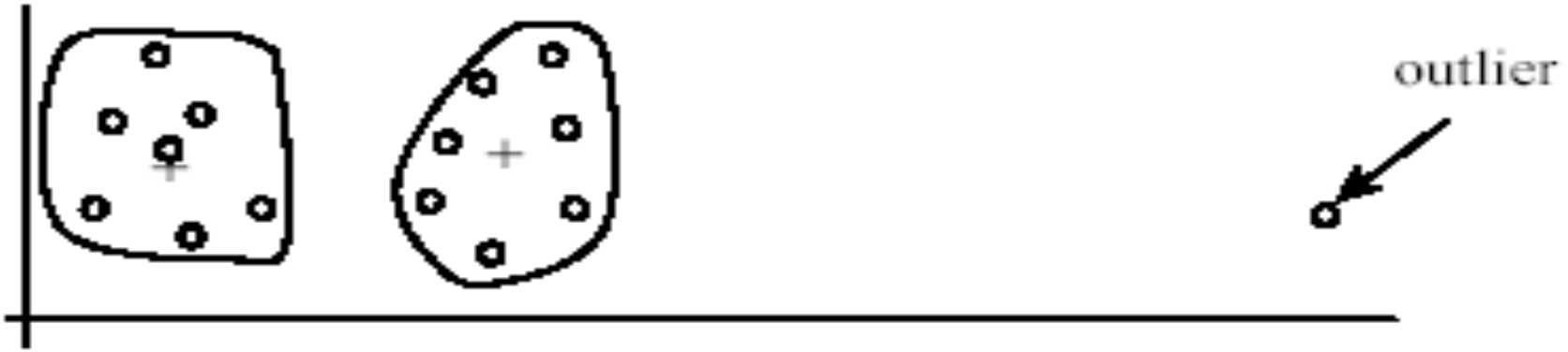
- The algorithm is sensitive to **outliers**
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

Adapted from Bing Liu, UIC
http://www.cs.uic.edu/~liub/teach/cs583-fall-05/CS583-unsupervised-learning.ppt

Issues for *K*-means: Problems with outliers



(A): Undesirable clusters



(B): Ideal clusters

Adapted from Bing Liu, UIC
http://www.cs.uic.edu/~liub/teach/cs583-fall-05/CS583-unsupervised-learning.ppt

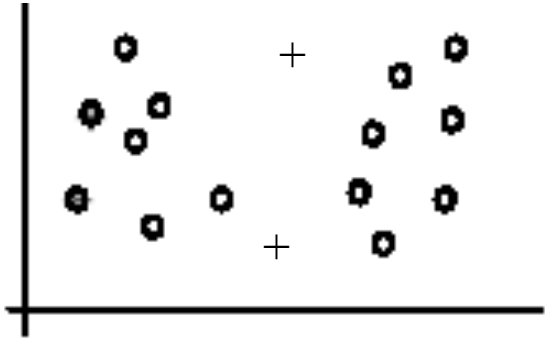
Dealing with outliers

- One method is to remove some data points in the clustering process that are much further away from the centroids than other data points.
  - Expensive
  - Not always a good idea!

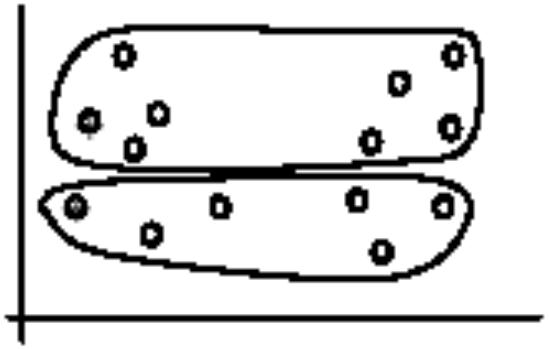
- Another method is to perform random sampling. Since in sampling we only choose a small subset of the data points, the chance of selecting an outlier is very small.
  - Assign the rest of the data points to the clusters by distance or similarity comparison, or classification

Issues for *K*-means (cont …)

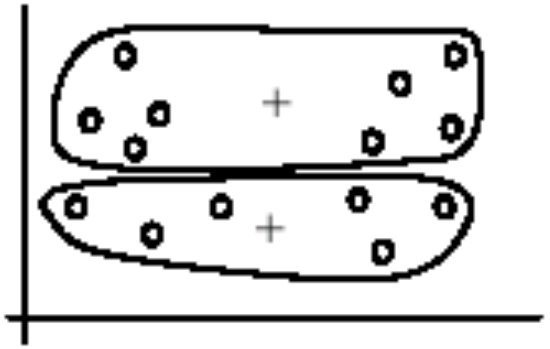
- The algorithm is sensitive to initial seeds.



(A). Random selection of seeds (centroids)



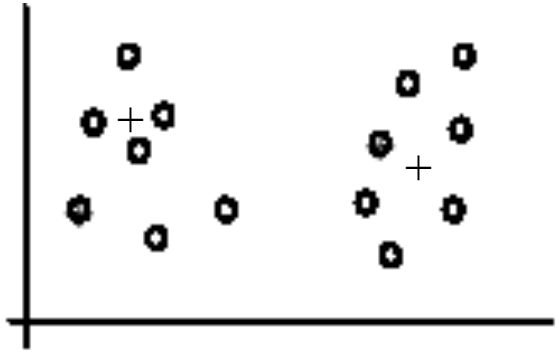
(B). Iteration 1



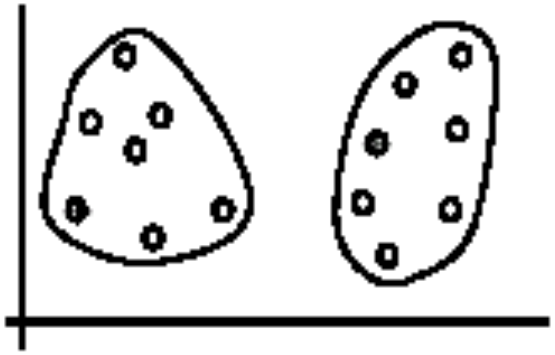
(C). Iteration 2

Issues for *K*-means (cont …)

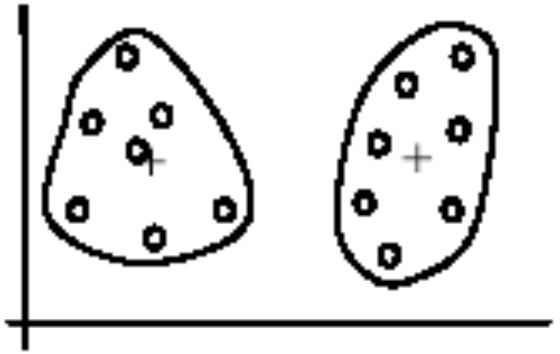
- If we use different seeds: good results



(A). Random selection of *k* seeds (centroids)



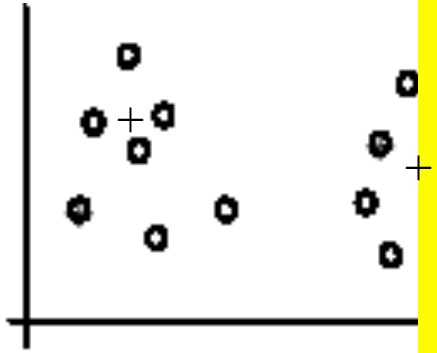
(B). Iteration 1



(C). Iteration 2

Issues for *K*-means (cont …)

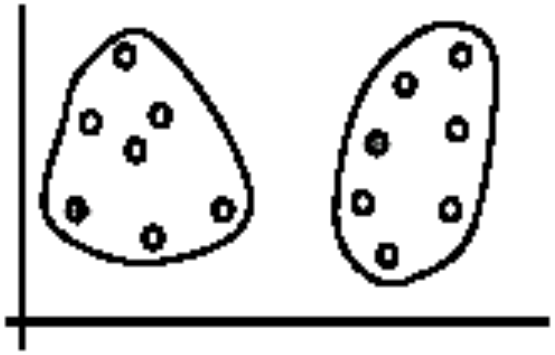
- If we use different seeds: good re



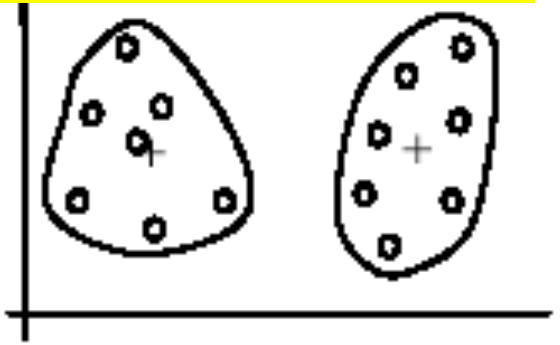
(A). Random selection of *k* s

Often can improve *K*-means results by doing several random restarts.

See assigned reading for methods to help choose good seeds



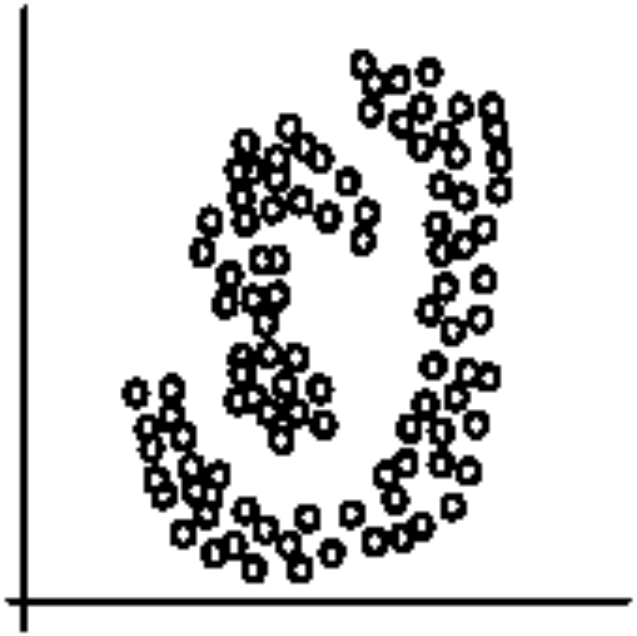
(B). Iteration 1



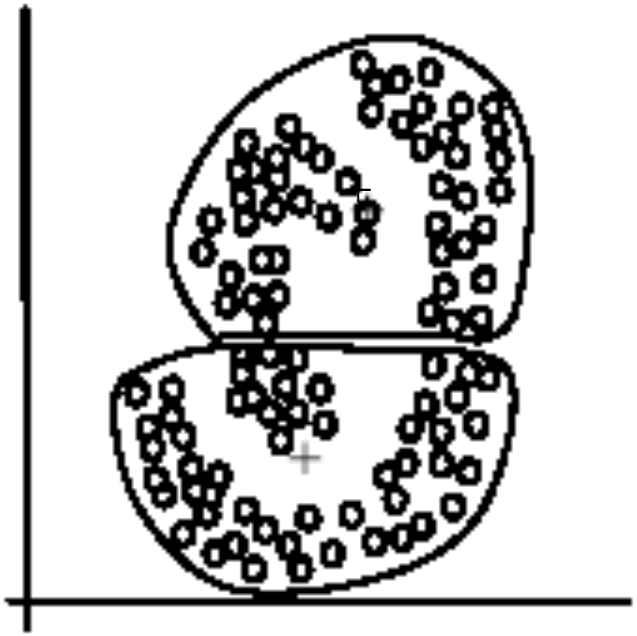
(C). Iteration 2

Issues for *K*-means (cont …)

- The *K-means* algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).



(A): Two natural clusters



(B): *k*-means clusters

Other Issues

- What if a cluster is empty?
  - Choose a replacement centroid
    - At random, or
    - From cluster that has highest SSE

- How to choose *K* ?

K-means as an Optimization Problem

- Optimization problem: minimize total SSE.
- Assume, for simplicity, that data is one-dimensional: i.e., $dist(x,y) = (x - y)^2$
- We want to minimize SSE, where

$$SSE = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} dist(\mathbf{x}, \mathbf{m}_i)^2$$

$$= (x - m_i)^2 \quad (\mathbf{x} = x \text{ is one-dimensional})$$

$$\frac{\partial}{\partial m_k} SSE = \frac{\partial}{\partial m_k} \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} (x - m_i)^2$$

$$= \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} \frac{\partial}{\partial m_k} (x - m_i)^2$$

$$= \sum_{\mathbf{x} \in C_k} 2(x - m_k) = \sum_{\mathbf{x} \in C_k} 2x - 2m_k = -2\left(\left| C_k \right| m_k + \sum_{\mathbf{x} \in C_k} x \right) = 0$$

$$\Rightarrow m_k = \frac{1}{\left| C_k \right|} \sum_{\mathbf{x} \in C_k} x$$

This justifies use of m_k (i.e., mean) as centroid of cluster k.

Topics for next time

- How to determine the *K* in *K*-means?
- Hierarchical clustering
- Soft clustering with Gaussian mixture models
- Expectation-Maximization method
- Review for quiz Tuesday