# Unsupervised Learning

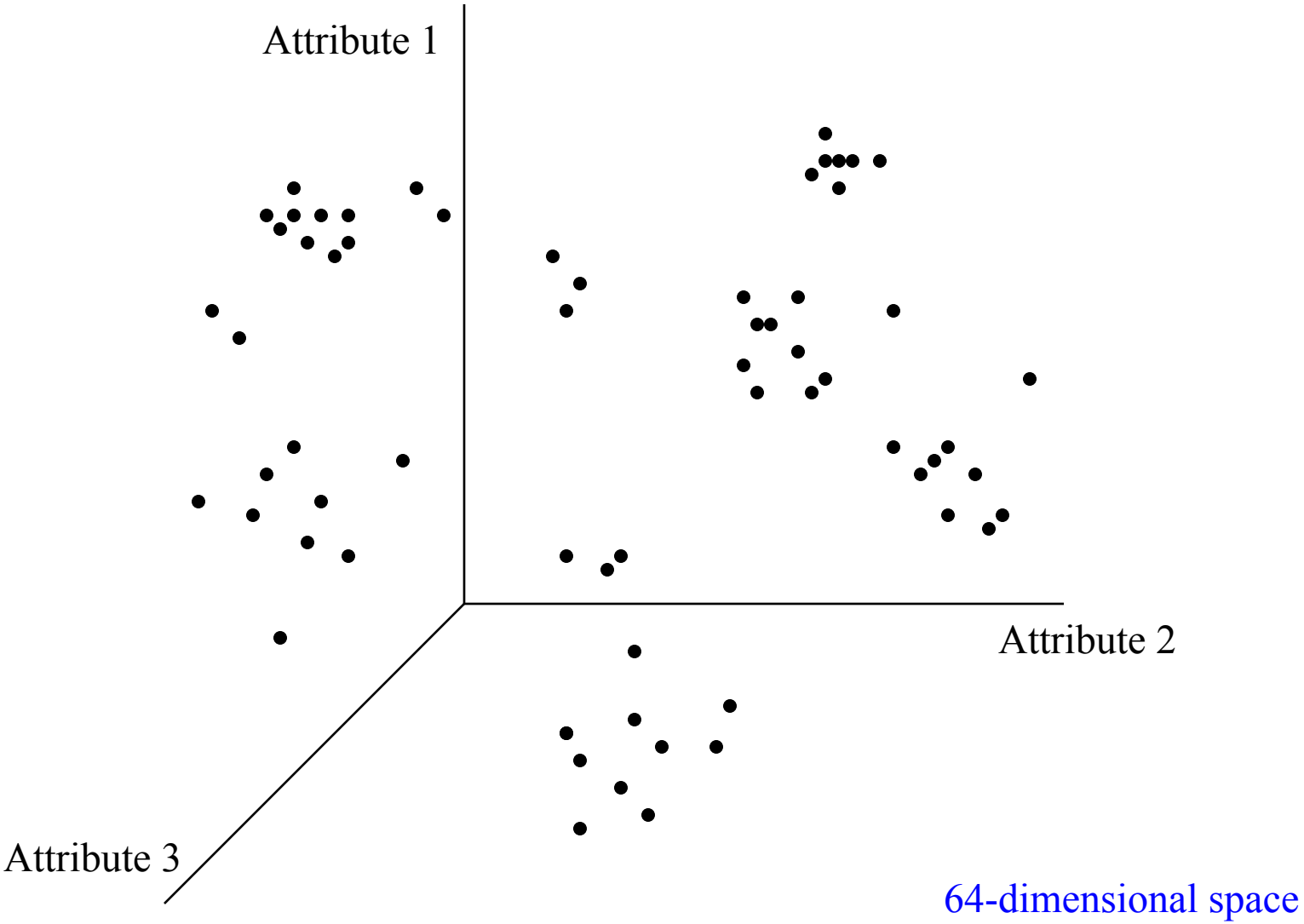
Reading:

Chapter 8 from *Introduction to Data Mining* by Tan, Steinbach, and Kumar, pp. 489-518, 532-544, 548-552

(http://www-users.cs.umn.edu/%7Ekumar/dmbook/ch8.pdf)

# Supervised learning vs. unsupervised learning

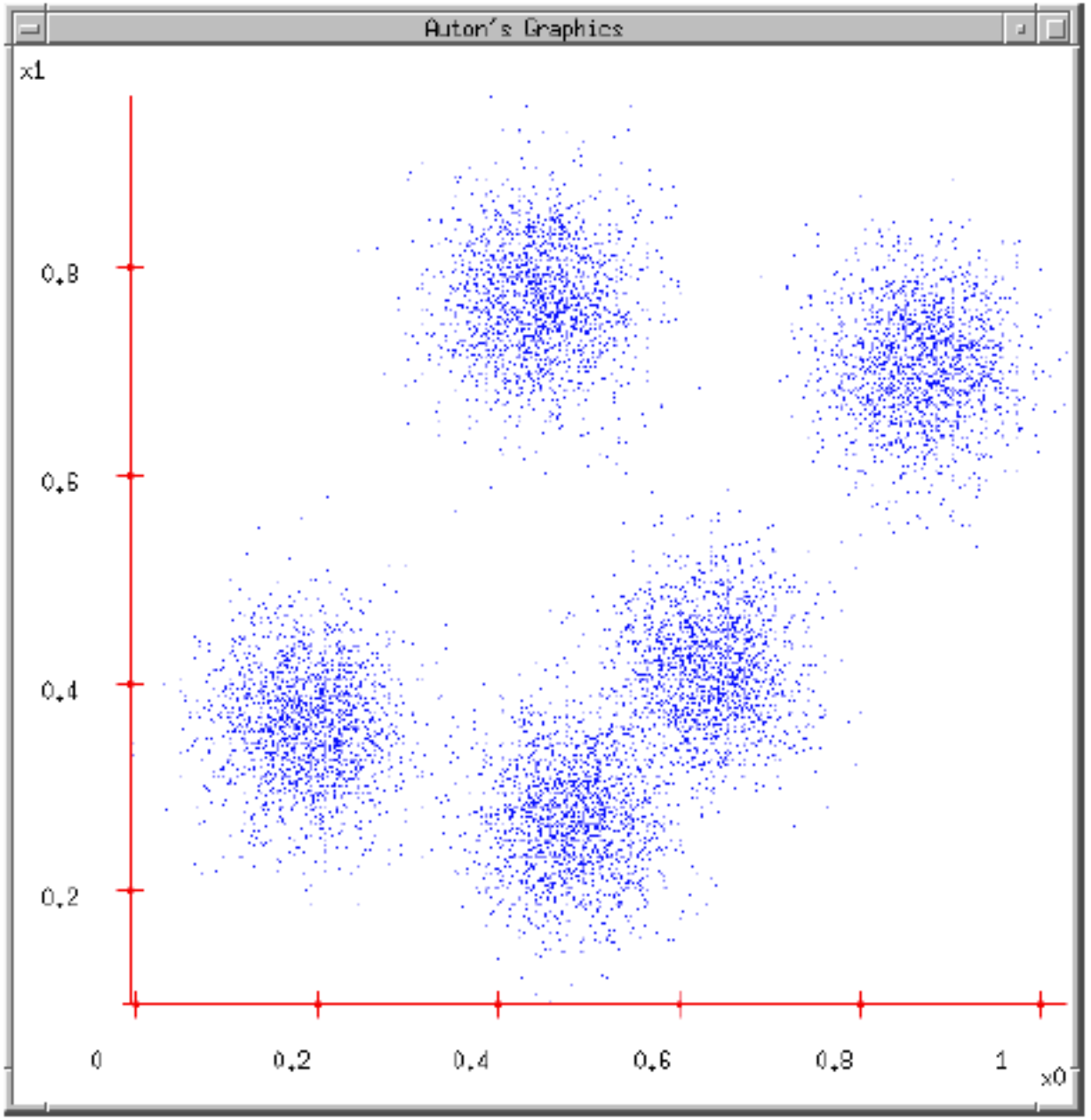
# Unsupervised Classification of Optdigits




64-dimensional space

# K-means

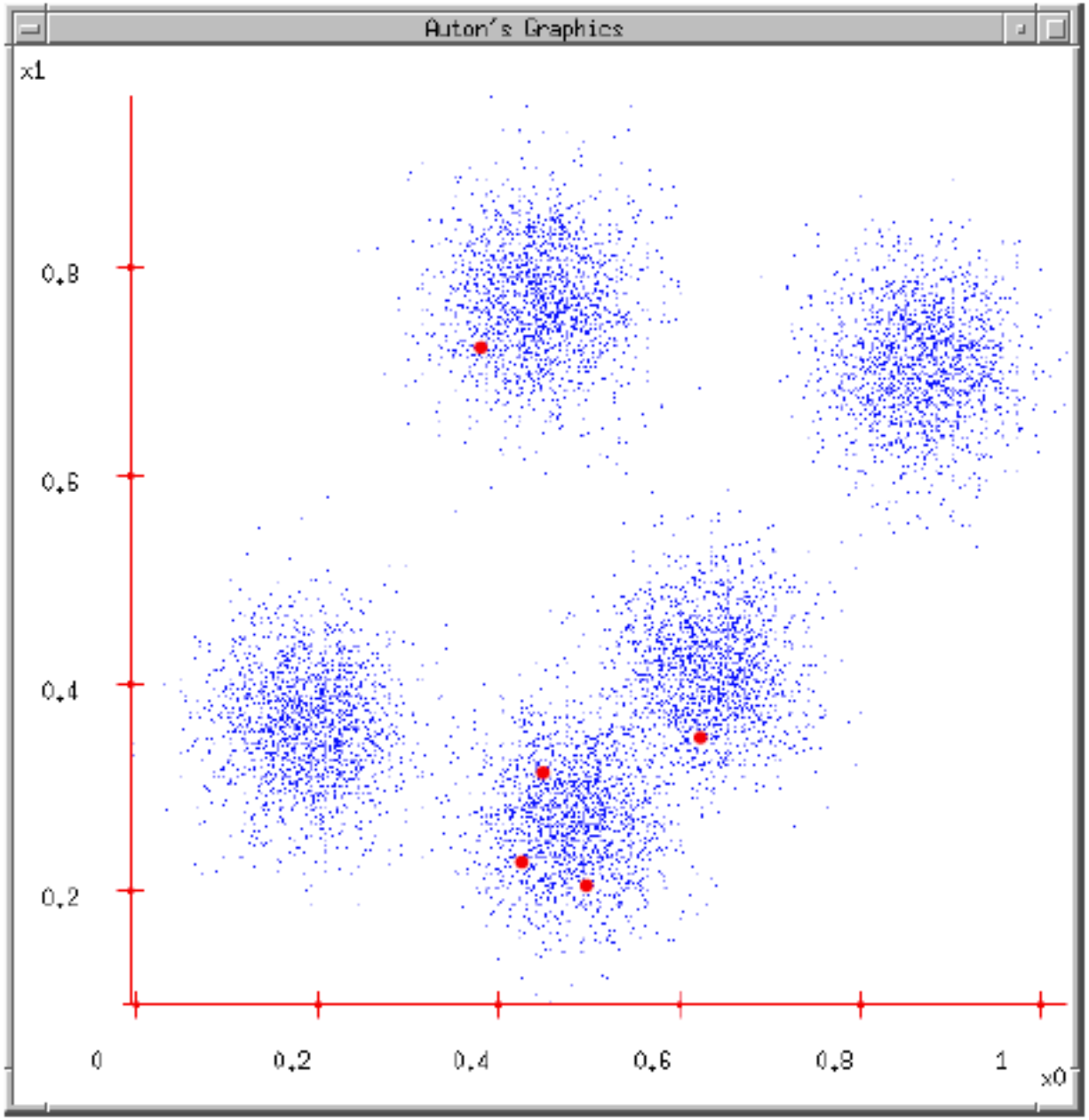
1. Ask user how many clusters they'd like. *(e.g. k=5)*




Adapted from Andrew Moore, http://www.cs.cmu.edu/~awm/tutorials

# K-means

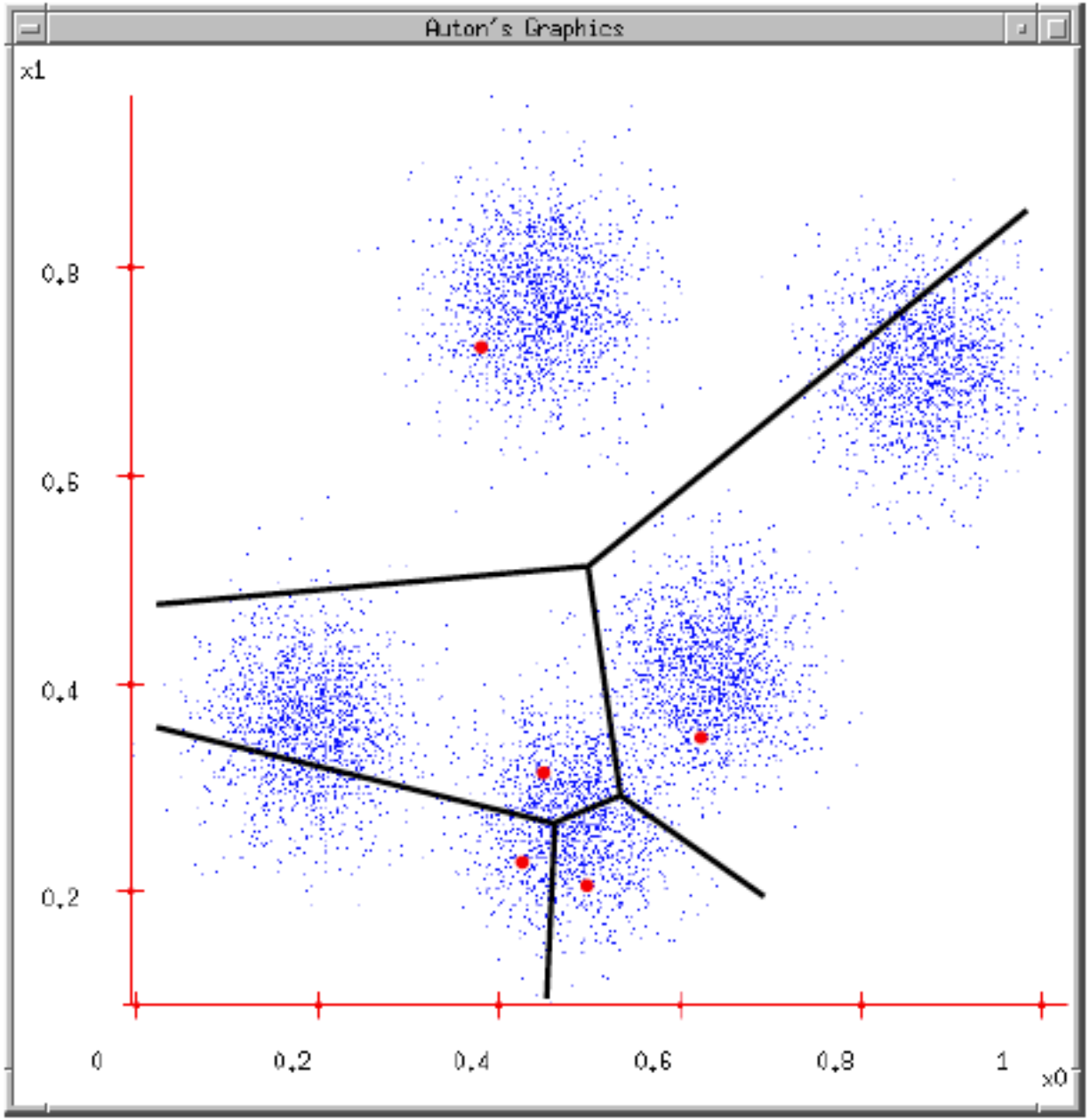
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations




Adapted from Andrew Moore, http://www.cs.cmu.edu/~awm/tutorials

# K-means

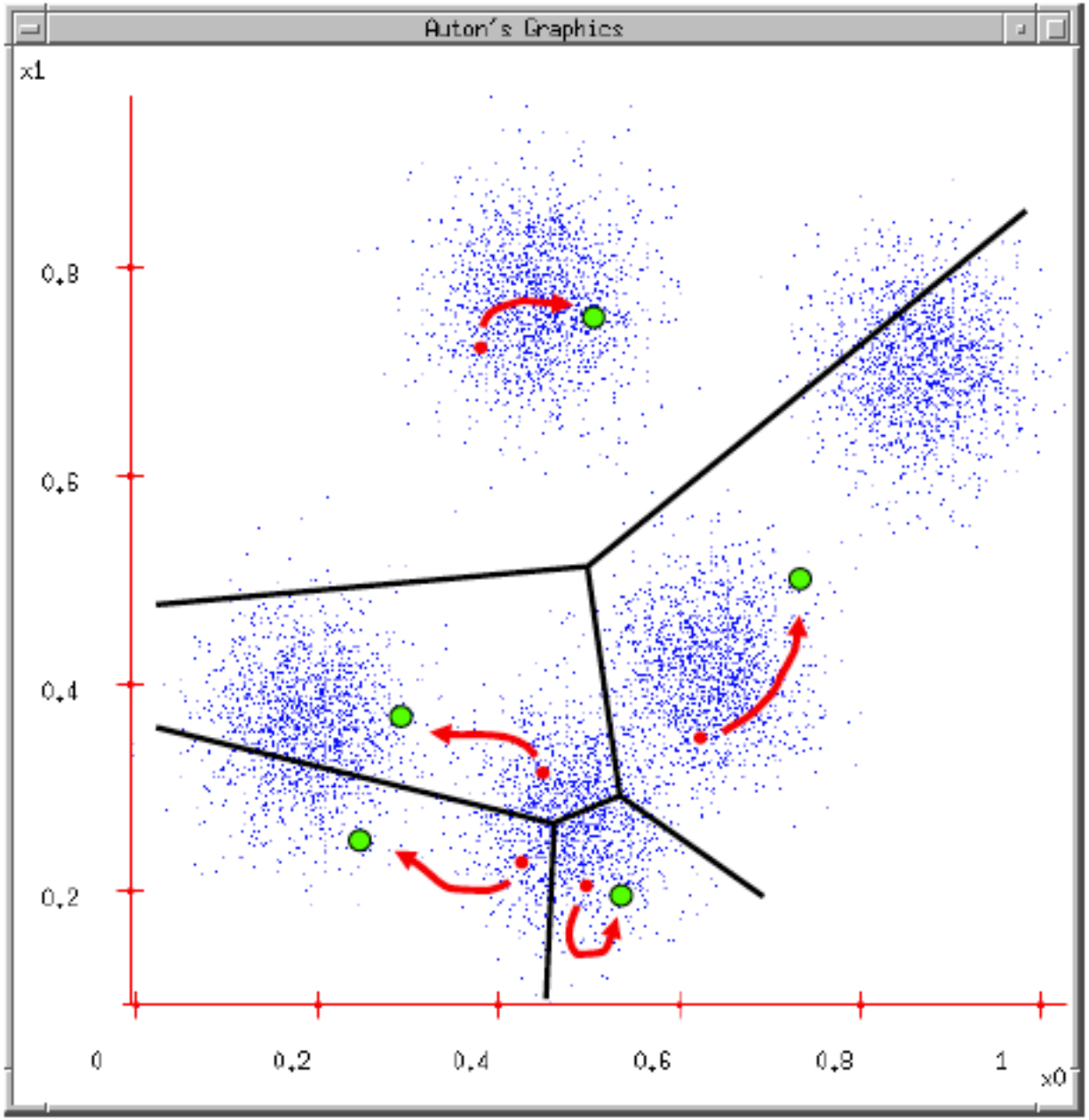
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)




Adapted from Andrew Moore, http://www.cs.cmu.edu/~awm/tutorials

# K-means

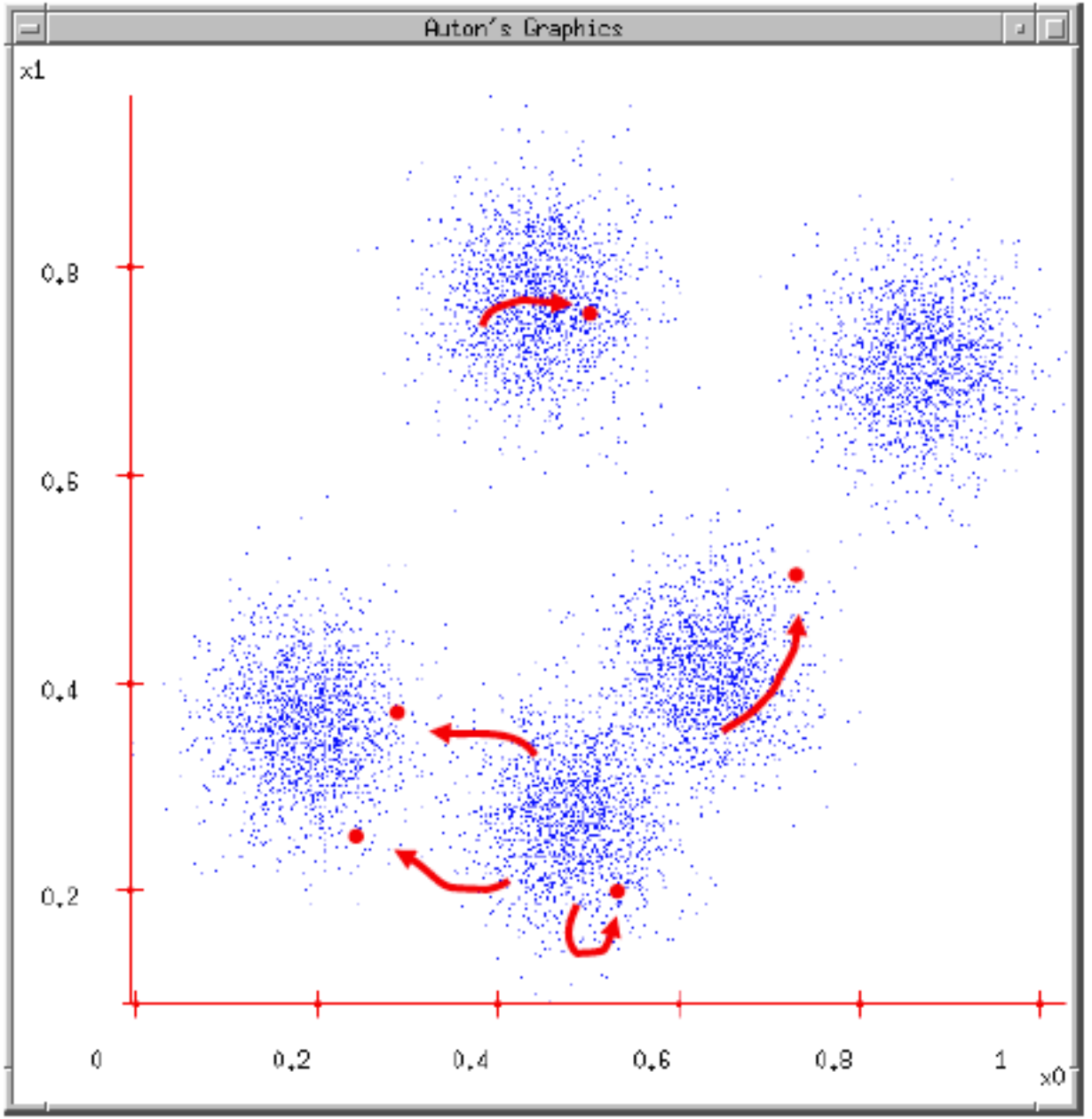
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns




Adapted from Andrew Moore, http://www.cs.cmu.edu/~awm/tutorials

# 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.

# 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.

**Typically, use mean of points in cluster as centroid**

# 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.

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

# Demo

http://www.rob.cs.tu-bs.de/content/04-teaching/06-interactive/Kmeans/Kmeans.html

# Stopping/convergence criteria

1. No change of centroids (or minimum change)

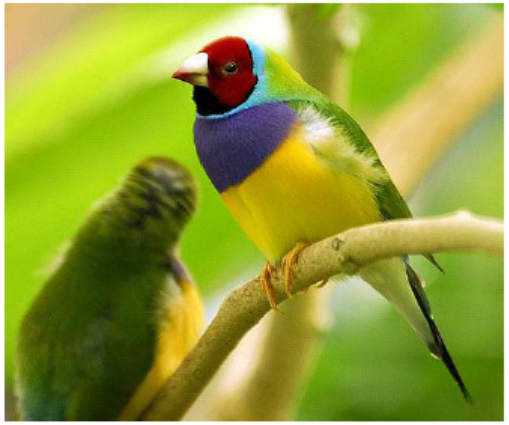
2. No (or minimum) decrease in the **sum squared error** (SSE),

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

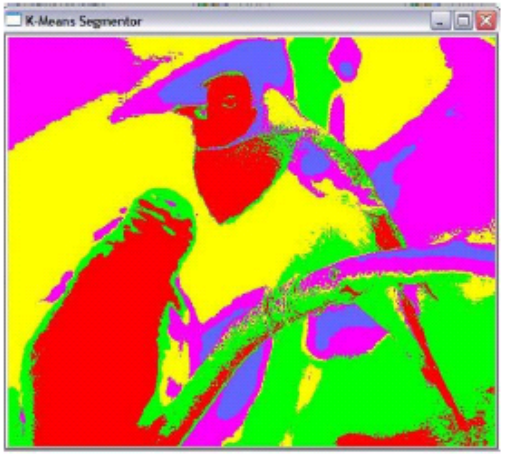
where $C_i$ is the $i$th cluster, $\mathbf{m}_i$ is the centroid of cluster $C_i$ (the mean vector of all the data points in $C_i$), and $d(\mathbf{x}, \mathbf{m}_i)$ is the distance between data point $\mathbf{x}$ and centroid $\mathbf{m}_i$.

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

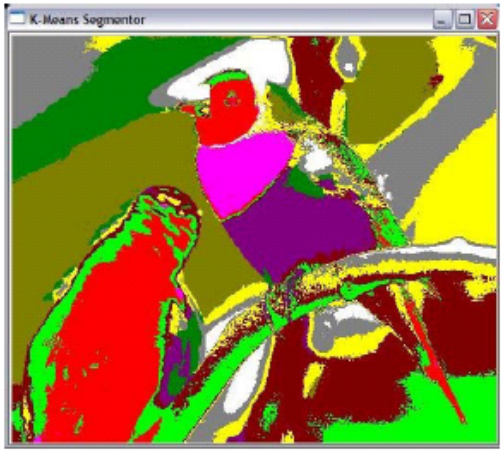
From http://vitroz.com/Documents/Image%20Segmentation.pdf



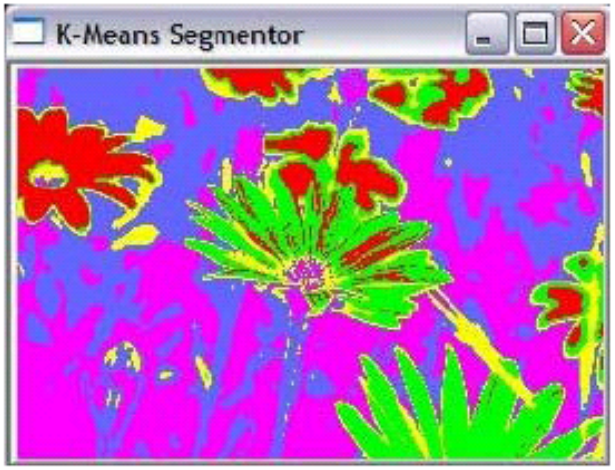
*K*=5, RGB space



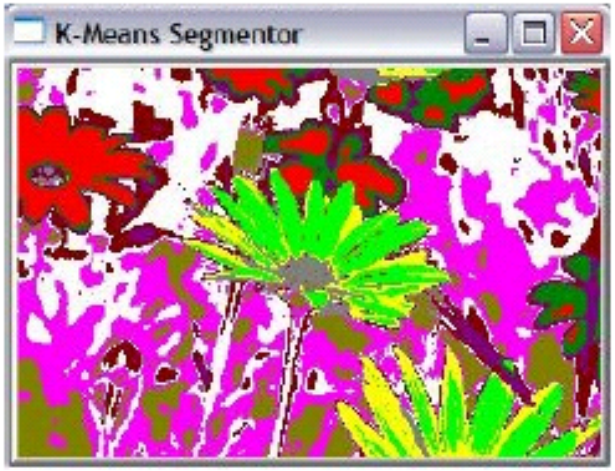

*K*=10, RGB space



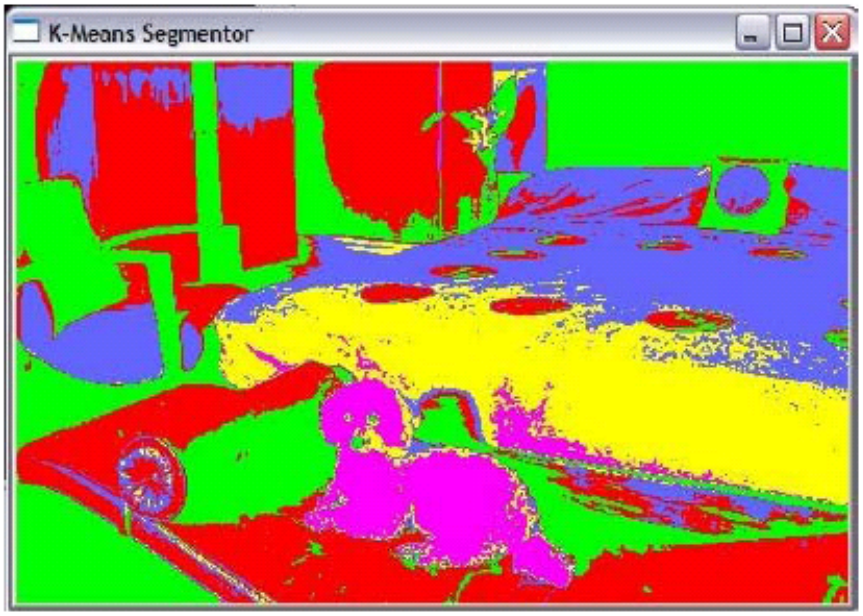
*K*=5, RGB space



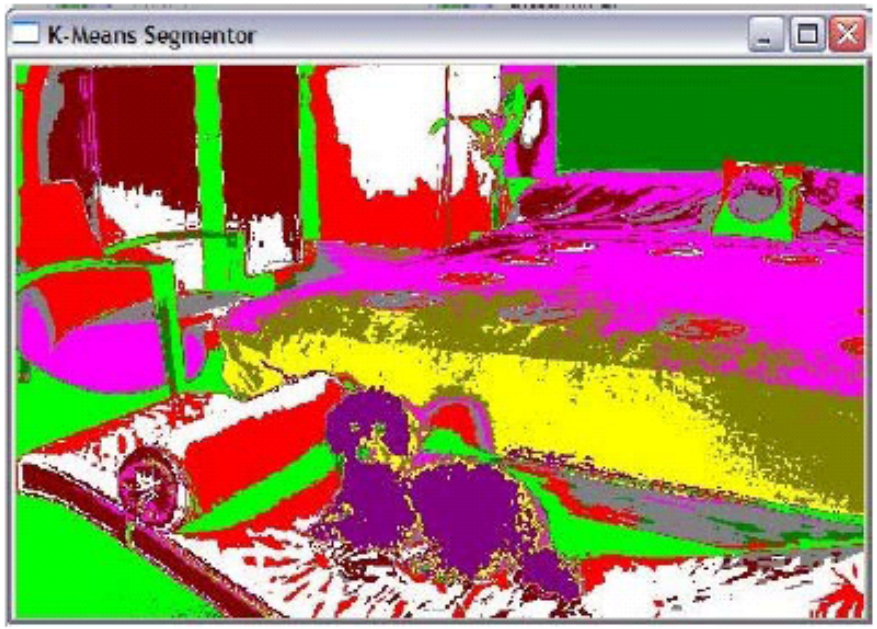

*K*=10, RGB space



*K*=5, RGB space



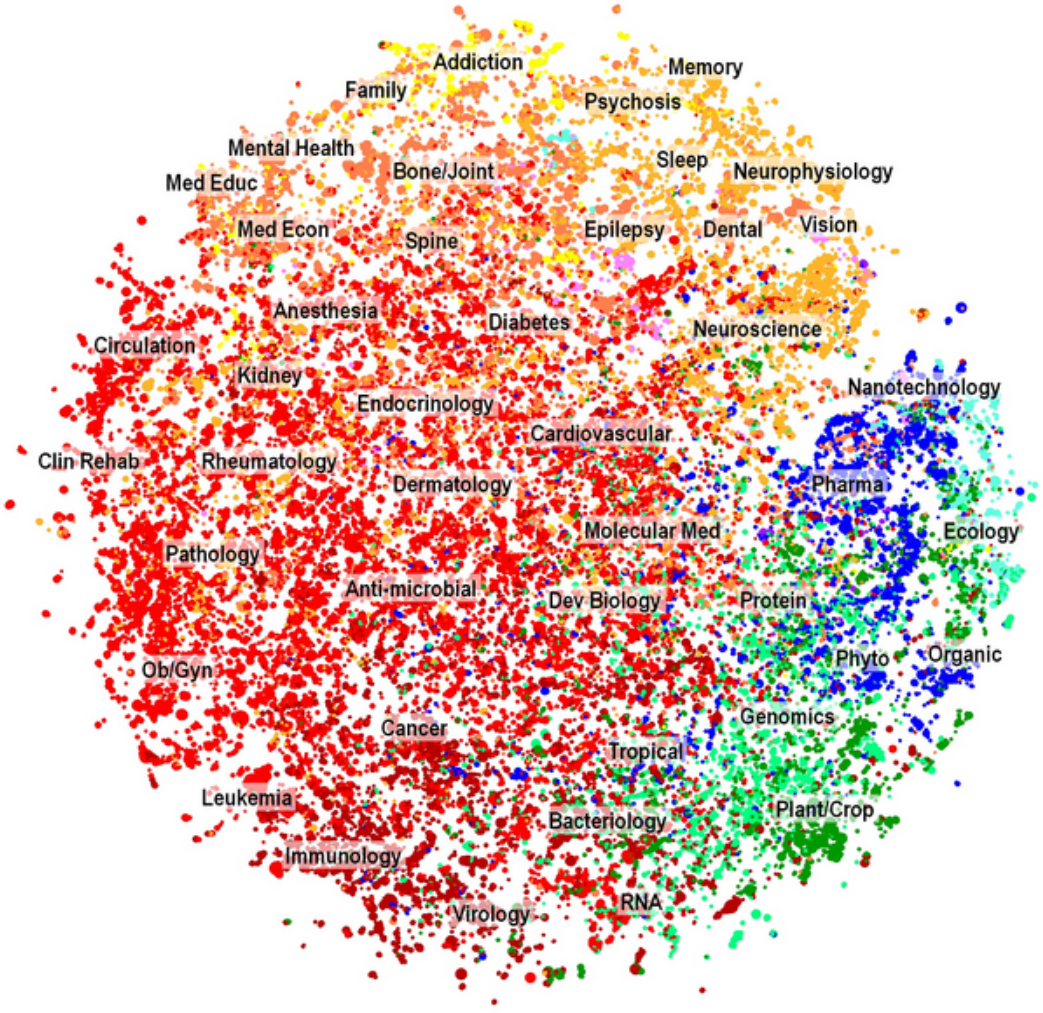

*K*=10, RGB space



# 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

# *K-means* clustering example

Consider the following five two-dimensional data points,

$$\mathbf{x}_1 = (1,1),\ \mathbf{x}_2 = (0,3),\ \mathbf{x}_3 = (2,2),\ \mathbf{x}_4 = (4,3),\ \mathbf{x}_5 = (4,4)$$

and the following two initial cluster centers,

$$\mathbf{m}_1 = (0,0),\ \mathbf{m}_2 = (1,4)$$

Simulate the K-means algorithm by hand on this data and these initial centers, using Euclidean distance.

Give the SSE of the clustering.

# 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.,

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

**Calculate *Sum Squared Separation* for previous example.**

# 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 exercise

Suppose there are 3 classes: 1, 2, 3

Class 1: 6 instances
Class 2: 6 instances
Class 3: 9 instances

| Cluster 1 | Cluster2 | Cluster3 |
|---|---|---|
| 1 2 1 3 1 1 3 | 2 3 3 3 2 3 3 | 1 1 3 2 2 3 2 |

Useful values:

$\log_2 1/7 = -2.8$
$\log_2 2/7 = -1.8$
$\log_2 3/7 = -1.2$
$\log_2 4/7 = -.8$
$\log_2 5/7 = -.5$

# In-Class Exercises

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

# Weaknesses of K-means

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

# Weaknesses of K-means

- The algorithm is only applicable if the mean is defined.
  - For categorical data, *k*-mode - the centroid is represented by most frequent values.

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

# Weaknesses of K-means

- The algorithm is only applicable if the mean is defined.
  - For categorical data, *k*-mode - the centroid is represented by most frequent values.

- The user needs to specify *K*.

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

# Weaknesses of K-means

- The algorithm is only applicable if the mean is defined.
  - For categorical data, *k*-mode - the centroid is represented by 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

# Weaknesses of K-means

- The algorithm is only applicable if the mean is defined.
  - For categorical data, *k*-mode - the centroid is represented by most frequent values.

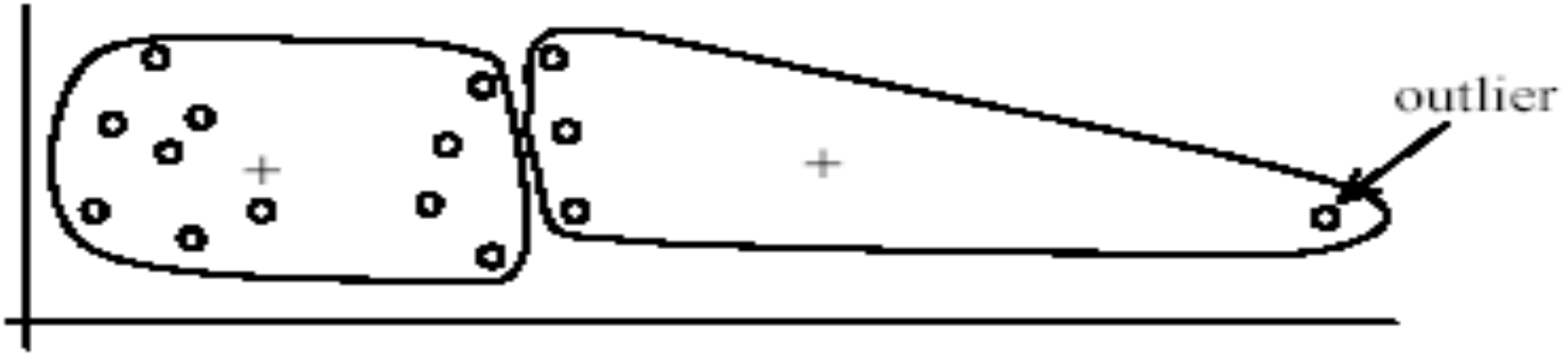
- The user needs to specify *K*.

- 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.

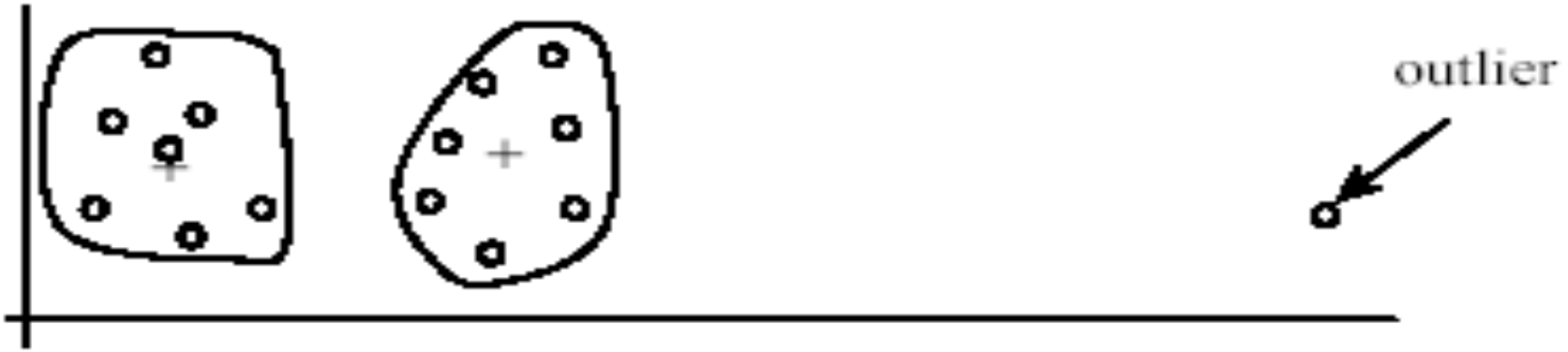
- K-means is sensitive to initial random centroids

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

# Weaknesses of 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

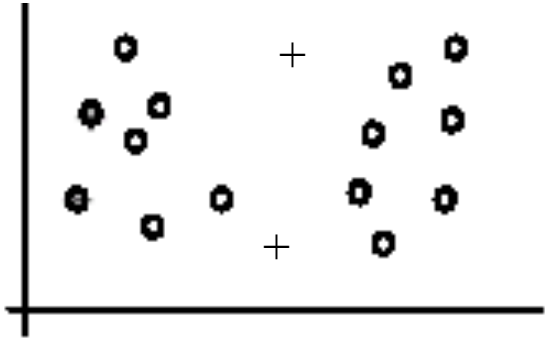
- One method is to remove some data points in the clustering process that are much further away from the centroids than other data points.
  - Expensive
  - Not always a good idea!

- 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

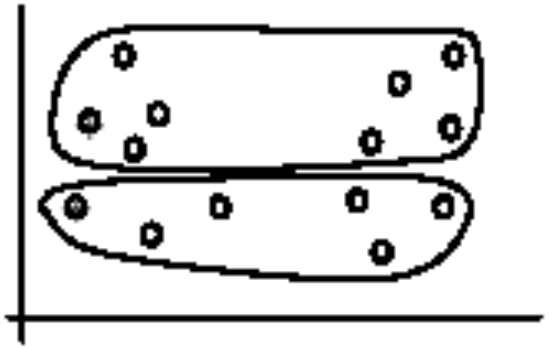
Adapted from Bing Liu, UIC
http://www.cs.uic.edu/~liub/teach/cs583-fall-05/CS583-unsupervised-learning.ppt

# Weaknesses of K-means (cont …)

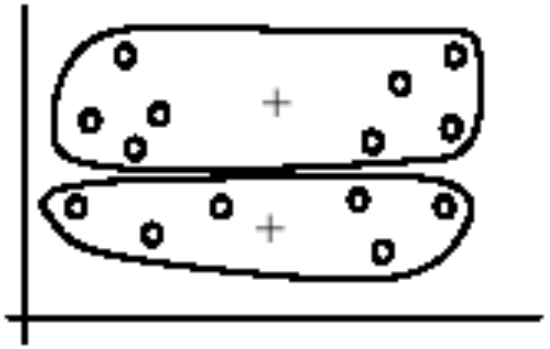
- The algorithm is sensitive to initial seeds.



(A). Random selection of seeds (centroids)



(B). Iteration 1

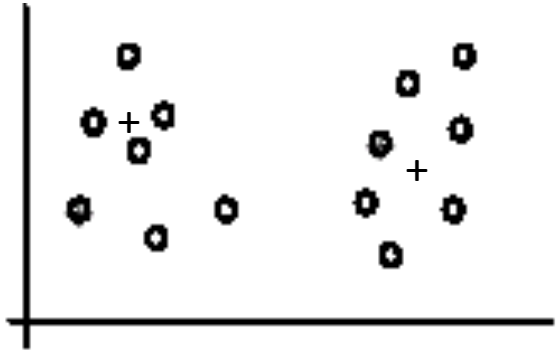


(C). Iteration 2

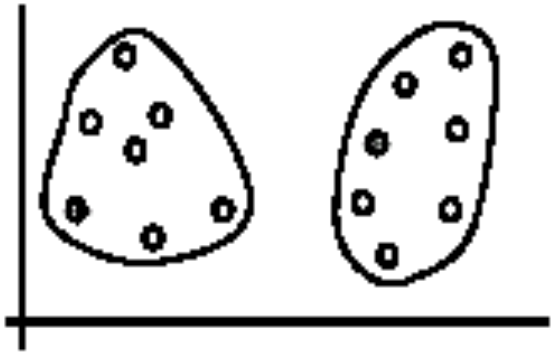
Adapted from Bing Liu, UIC
http://www.cs.uic.edu/~liub/teach/cs583-fall-05/CS583-unsupervised-learning.ppt

# Weaknesses of K-means (cont …)

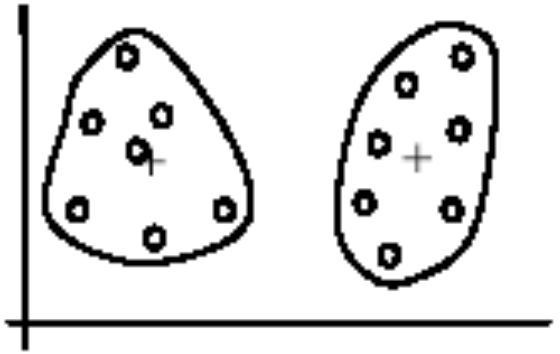
- If we use different seeds: good results



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



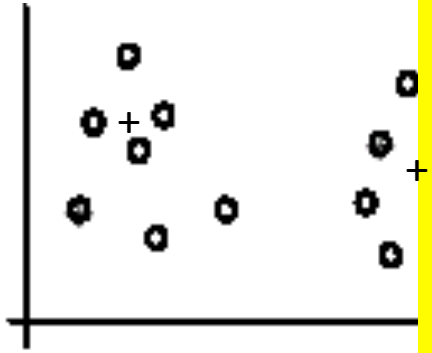
(B). Iteration 1



(C). Iteration 2

# Weaknesses of K-means (cont …)

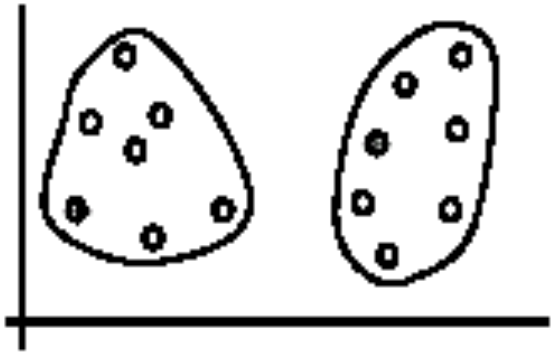
- If we use different seeds: good r



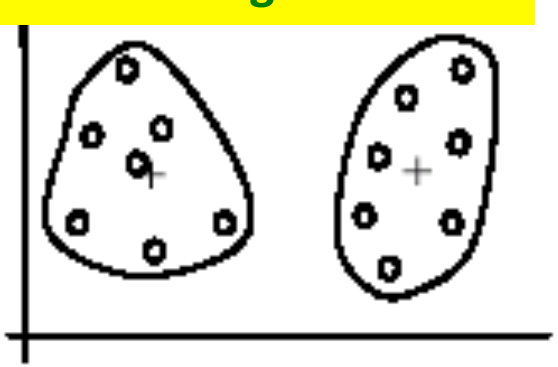
(A). Random selection of $k$ s

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

**See assigned reading for methods to help choose good seeds**



(B). Iteration 1

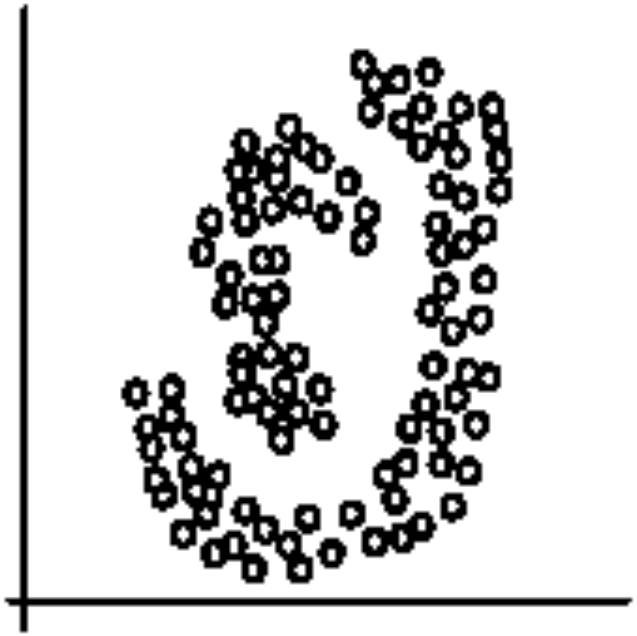


(C). Iteration 2

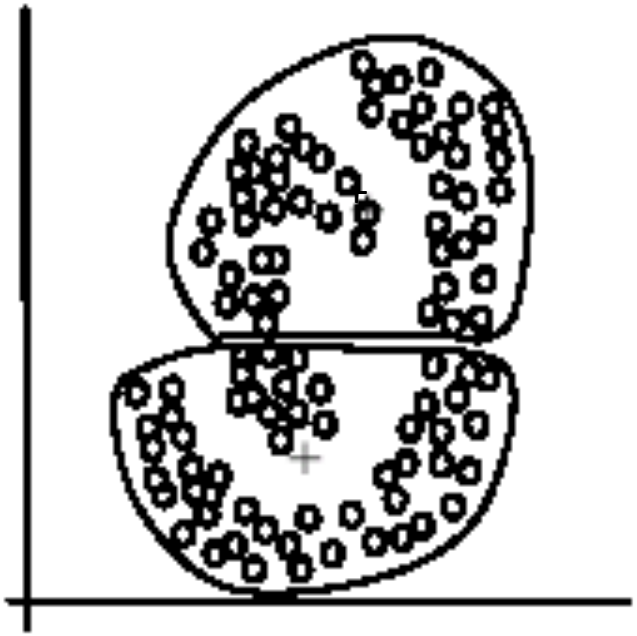
Adapted from Bing Liu, UIC
http://www.cs.uic.edu/~liub/teach/cs583-fall-05/CS583-unsupervised-learning.ppt

# Weaknesses of 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



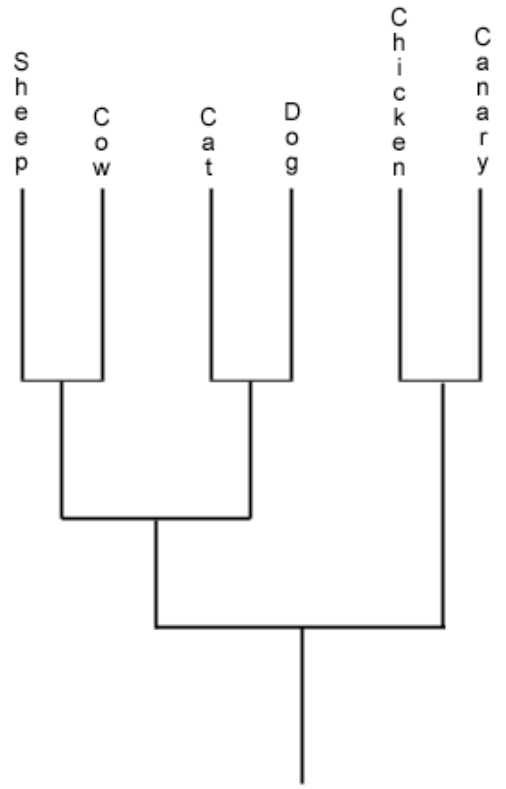
(B): *k*-means clusters

# 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* ?

# Hierarchical clustering




http://painconsortium.nih.gov/symptomresearch/chapter_23/sec27/cahs27pg1.htm

# Discriminative vs. Generative Models

- **Discriminative:**
  - E.g., drawing a hyperplane to separate data instances
- **Generative**
  - Create probabilistic models of data instances (e.g., K-means centroids and standard deviations)
  - For new instance **d**, find maximum likelihood model for that instance:

$$\underset{h \in models}{\operatorname{argmax}} P(\mathbf{d} \mid h)$$

  - **Example:** K-means**,** Gaussian mixture models