Machine Learning

Lecture 19b: Unsupervised Clustering (II): K-means

Dr. Beilun Wang

Southeast University
School of Computer Science
and Engineering

Course Content Plan

- □ Regression (supervised)

  Y is a continuous

- □ Classification (supervised)

  Y is a discrete

- □ Unsupervised models

  NO Y

  - □ Dimension Reduction (PCA)
  - □ Clustering (K-means, GMM/EM, Hierarchical)

- □ Learning theory

  About f()

- □ Graphical models

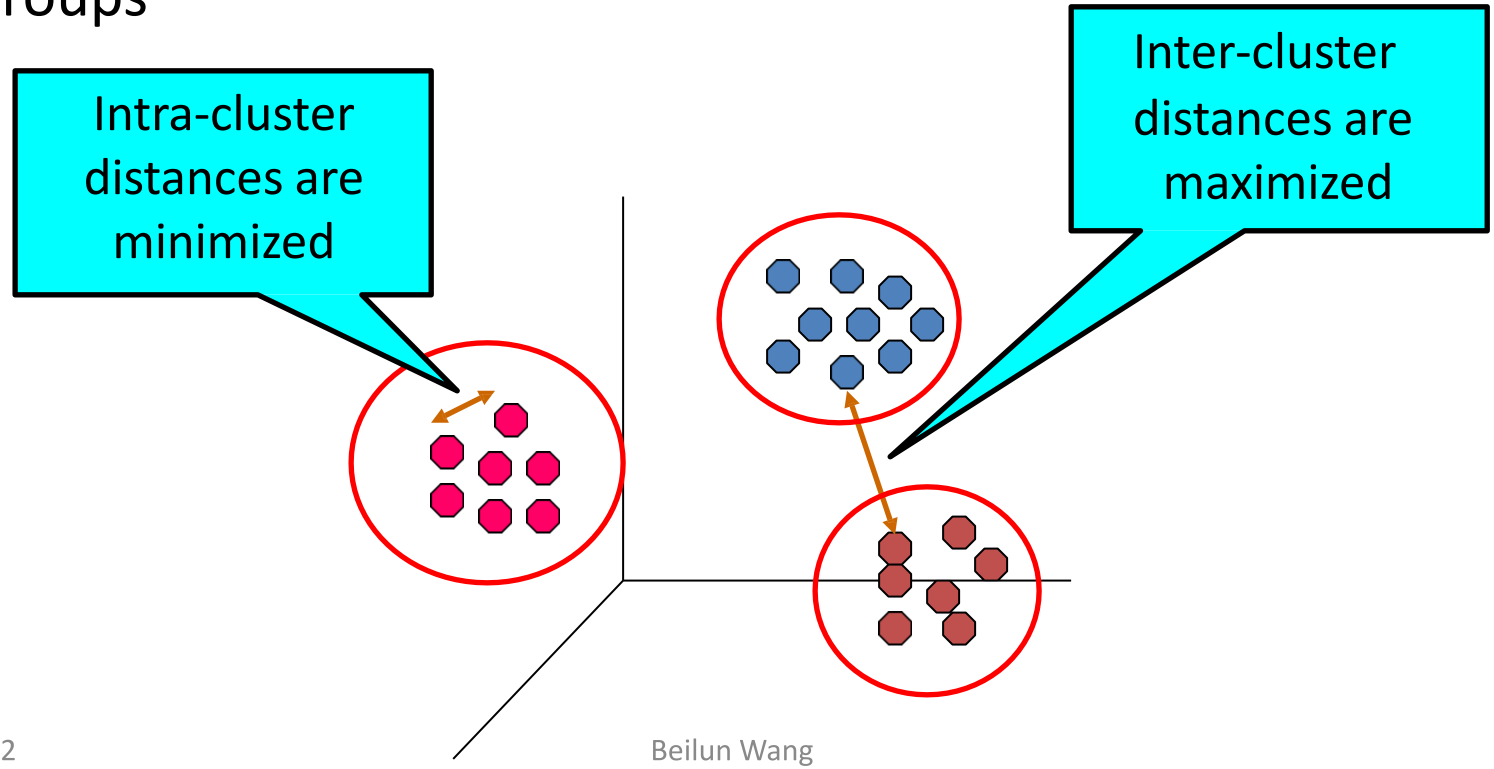
  About interactions among X1,… Xp

- □ Reinforcement Learning

  Learn program to Interact with its environment

What is clustering?

- Find groups (clusters) of data points such that data points in a group will be similar (or related) to one another and different from (or unrelated to) the data points in other groups



Intra-cluster distances are minimized

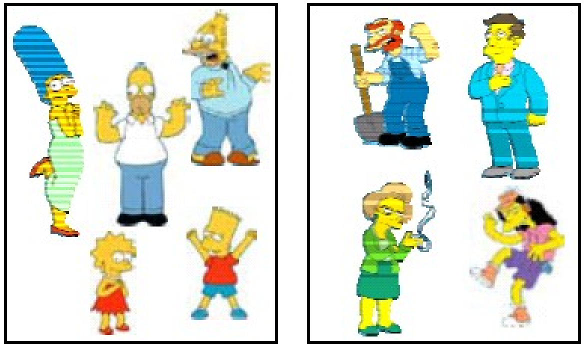
Inter-cluster distances are maximized

Today

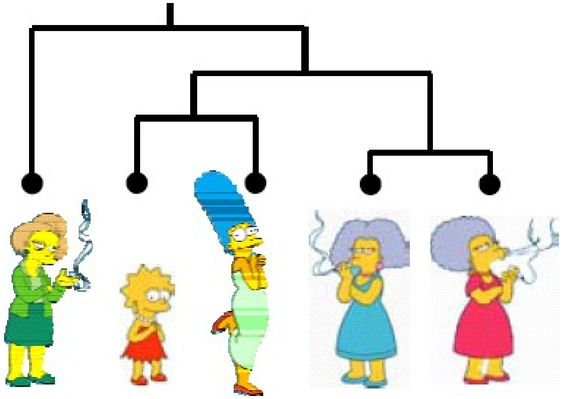
- Definition of "groupness"
- Definition of "similarity/distance"
- Clustering Algorithms
  - Hierarchical algorithms
  - Partitional algorithms
- Formal foundation and convergence
- How many clusters?

Clustering Algorithms

- Partitional algorithms
  - Usually start with a random (partial) partitioning
  - Refine it iteratively
    - K means clustering
    - Mixture-Model based clustering

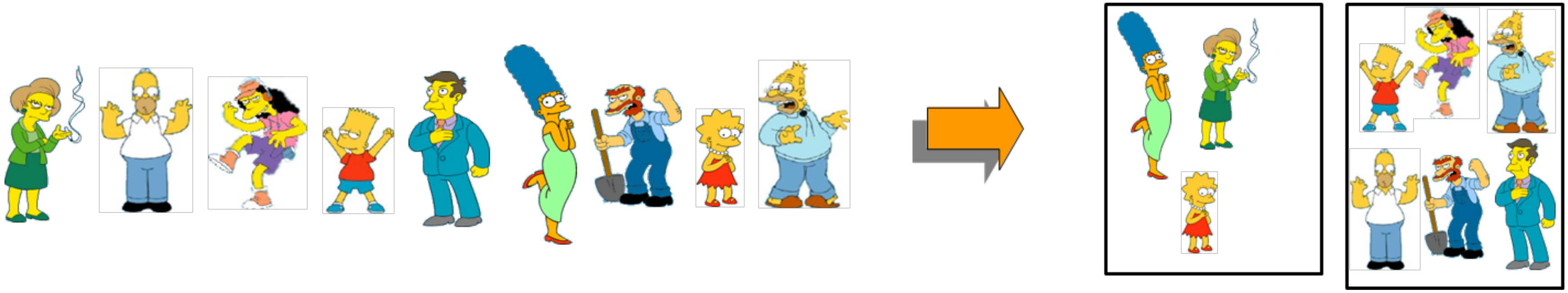


- Hierarchical algorithms
  - Bottom-up, agglomerative
  - Top-down, divisive

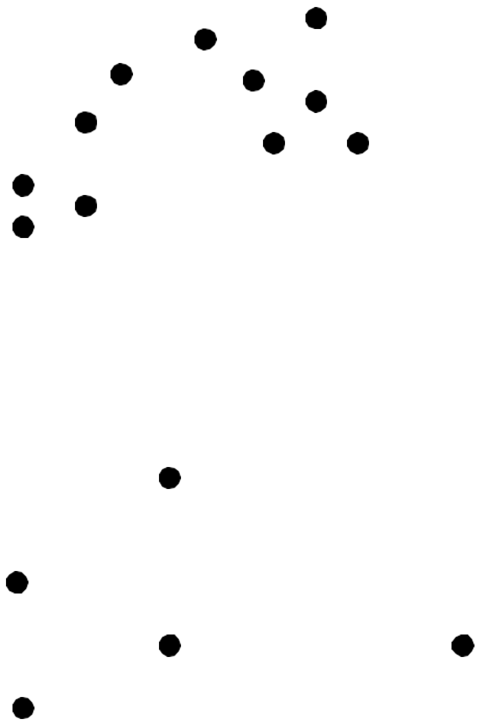


Partitional Clustering

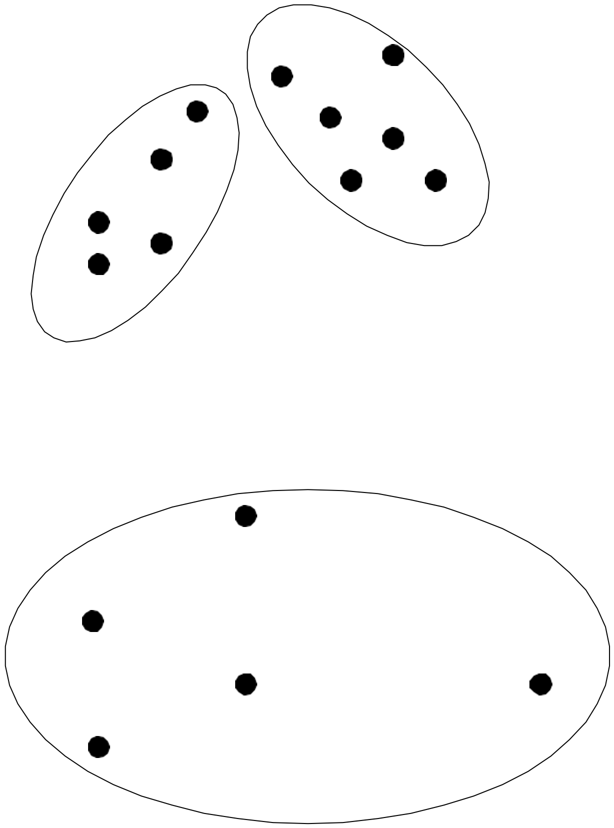
- Nonhierarchical
- Construct a partition of n objects into a set of K clusters
- User has to specify the desired number of clusters K.



Partitional Clustering (e.g. K=3)

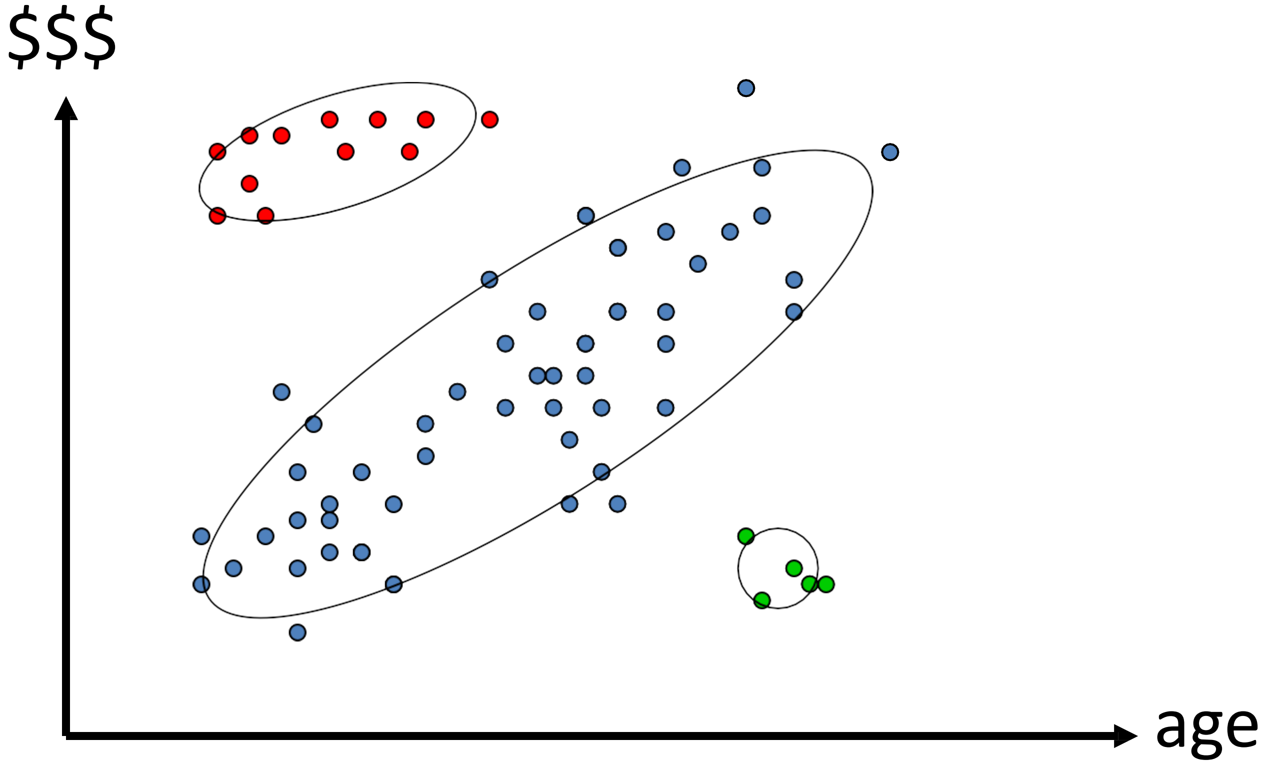


Original points



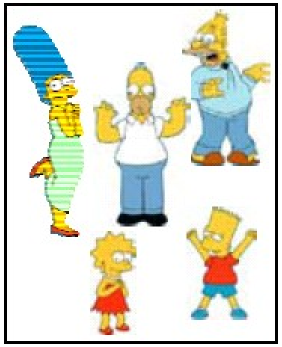
Partitional clustering

Partitional Clustering (e.g. K=3)



Clustering Algorithms

- Partitional algorithms
  - Usually start with a random (partial) partitioning
  - Refine it iteratively
    - K means clustering
    - Mixture-Model based clustering



Partitioning Algorithms

- Given: a set of objects and the number *K*

- Find: a partition of *K* clusters that optimizes a chosen partitioning criterion
  - Globally optimal: exhaustively enumerate all partitions
  - Effective heuristic methods: K-means and K-medoids algorithms

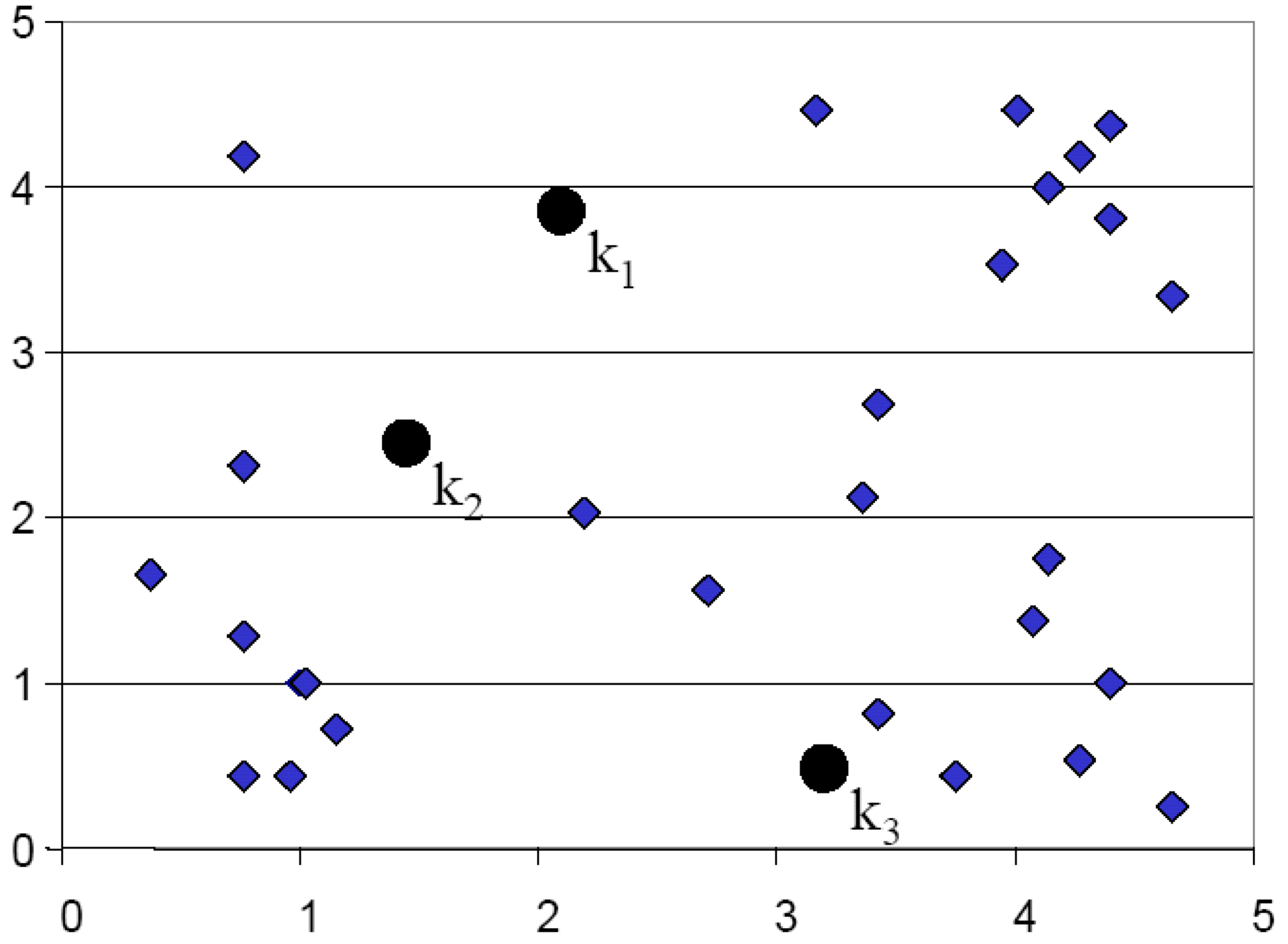
K-Means Algorithm

- Decide on a value for k.
- Initialize the k cluster centers randomly if necessary.
- Decide the class memberships of the N objects by assigning them to the nearest cluster centroids (aka the center of gravity or mean)

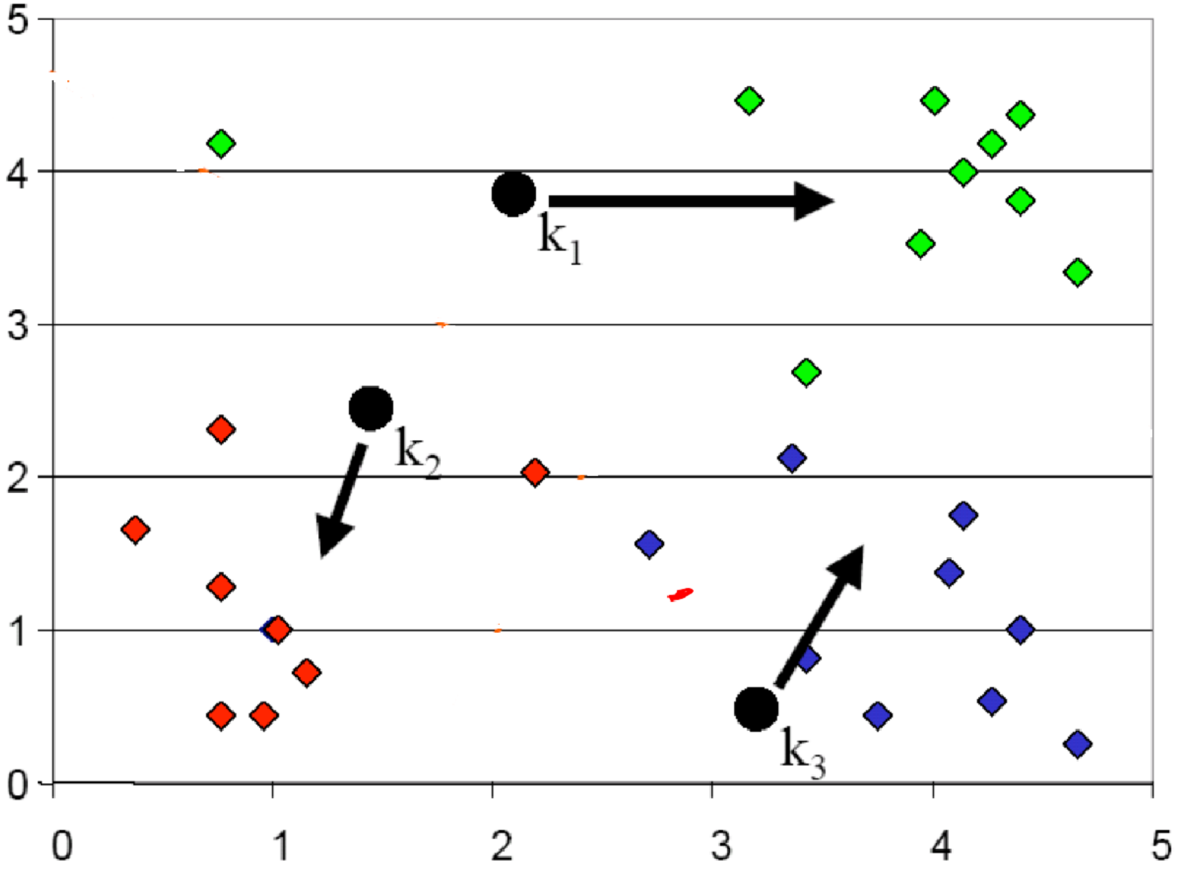
$$\vec{\mu}_k = \frac{1}{\mathcal{C}_k} \sum_{i \in \mathcal{C}_k} \vec{x}_i$$

- Re-estimate the k cluster centers, by assuming the memberships found above are correct.
- If none of the N objects changed membership in the last iteration, exit. Otherwise go to 3.

K-means Clustering: Step 1
Random guess of cluster centers

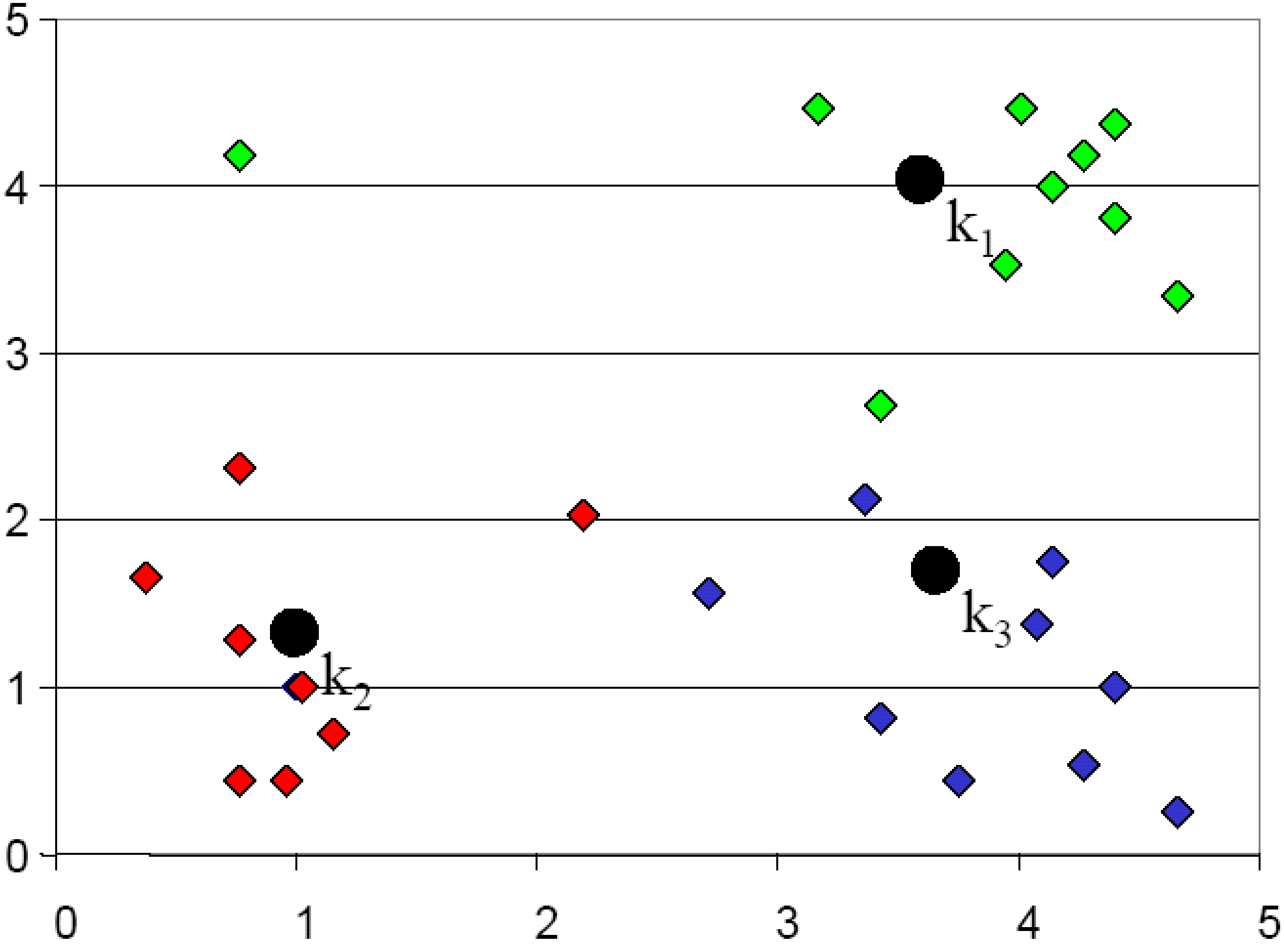


K-means Clustering: Step 2
Determine the membership of each data points

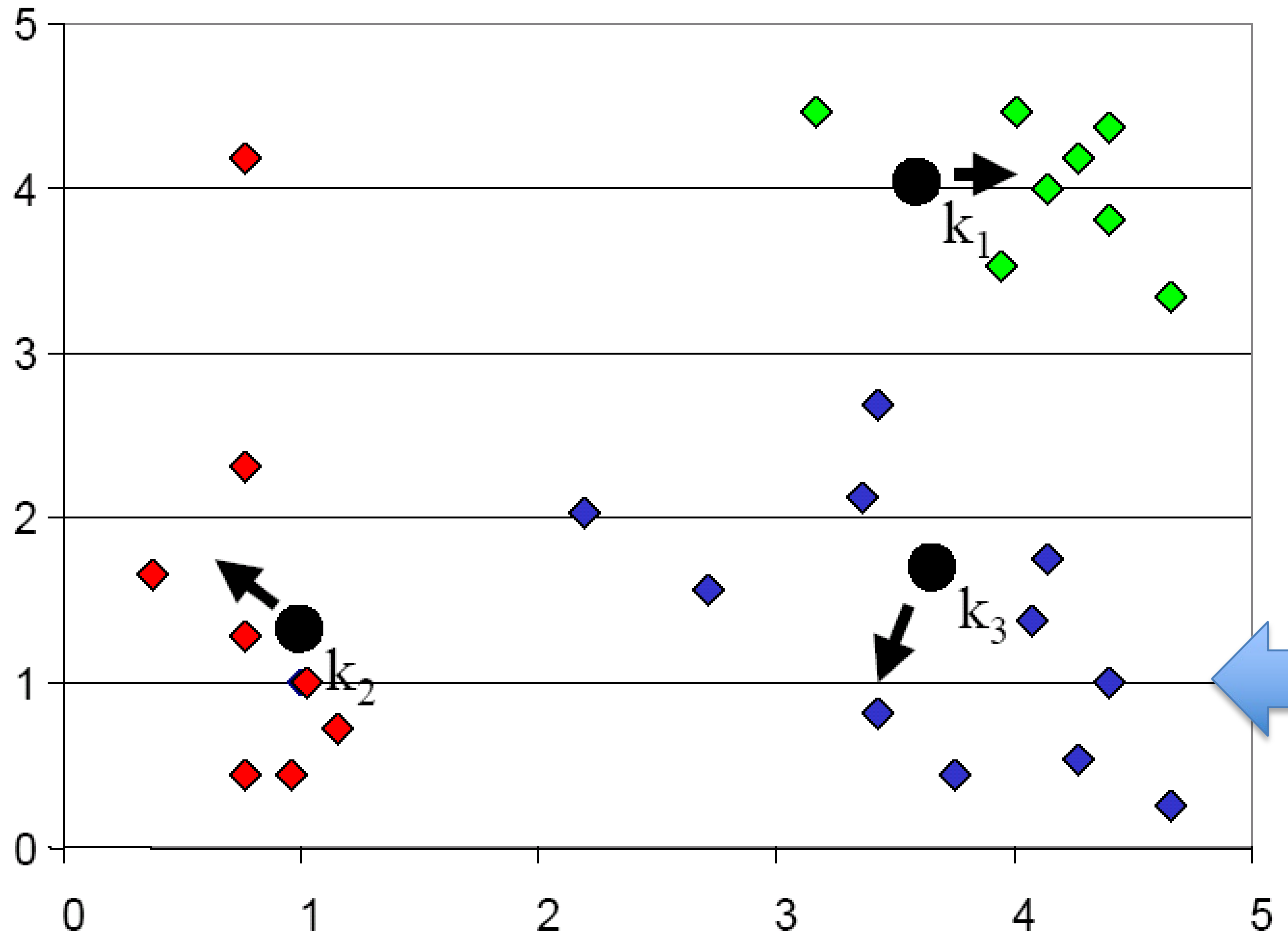


$$\vec{\mu}_k = \frac{1}{\mathcal{C}_k} \sum_{i \in \mathcal{C}_k} \vec{x}_i$$

K-means Clustering: Step 3
Adjust the cluster centers



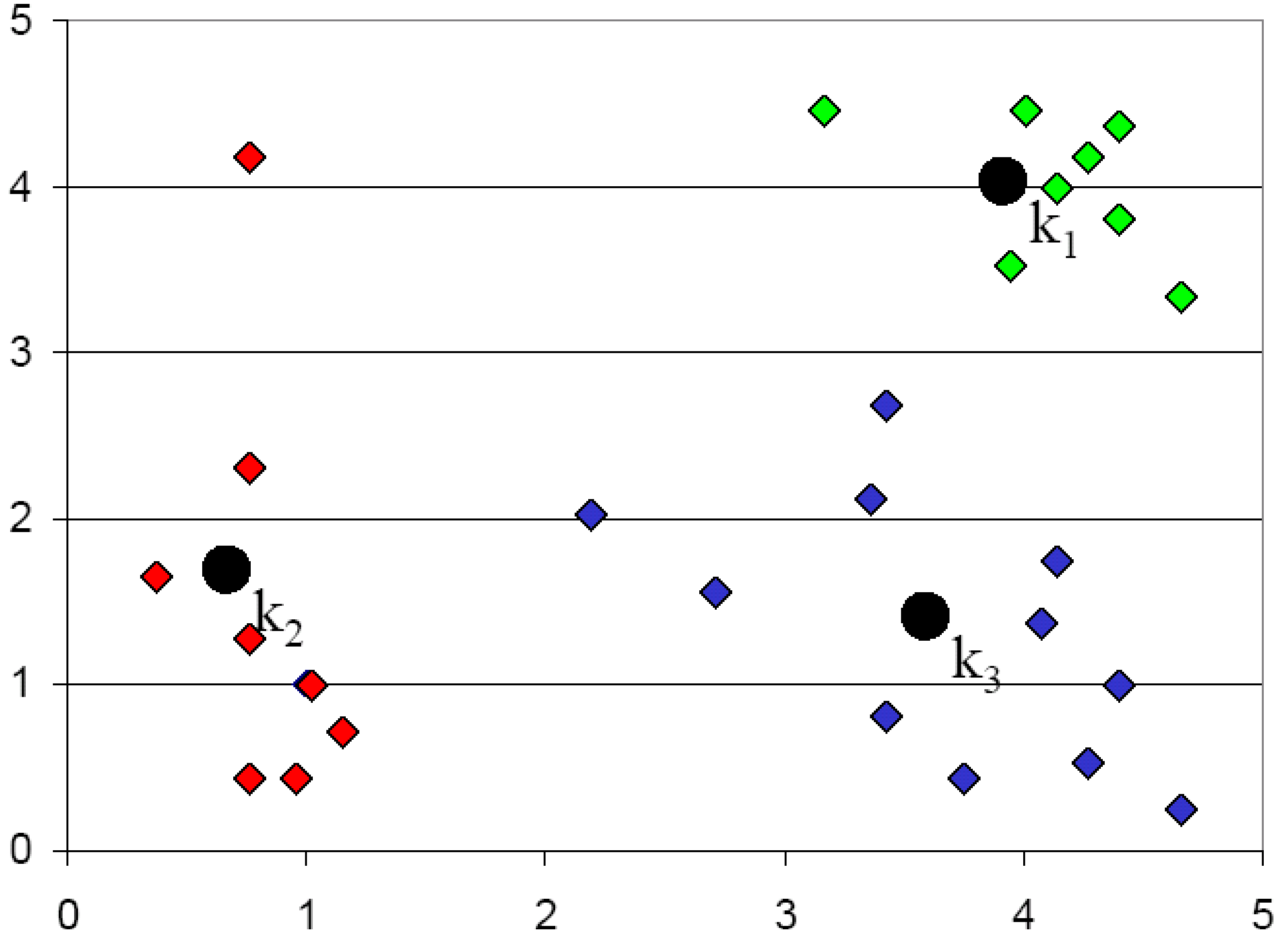
K-means Clustering: Step 4
Re-determine membership



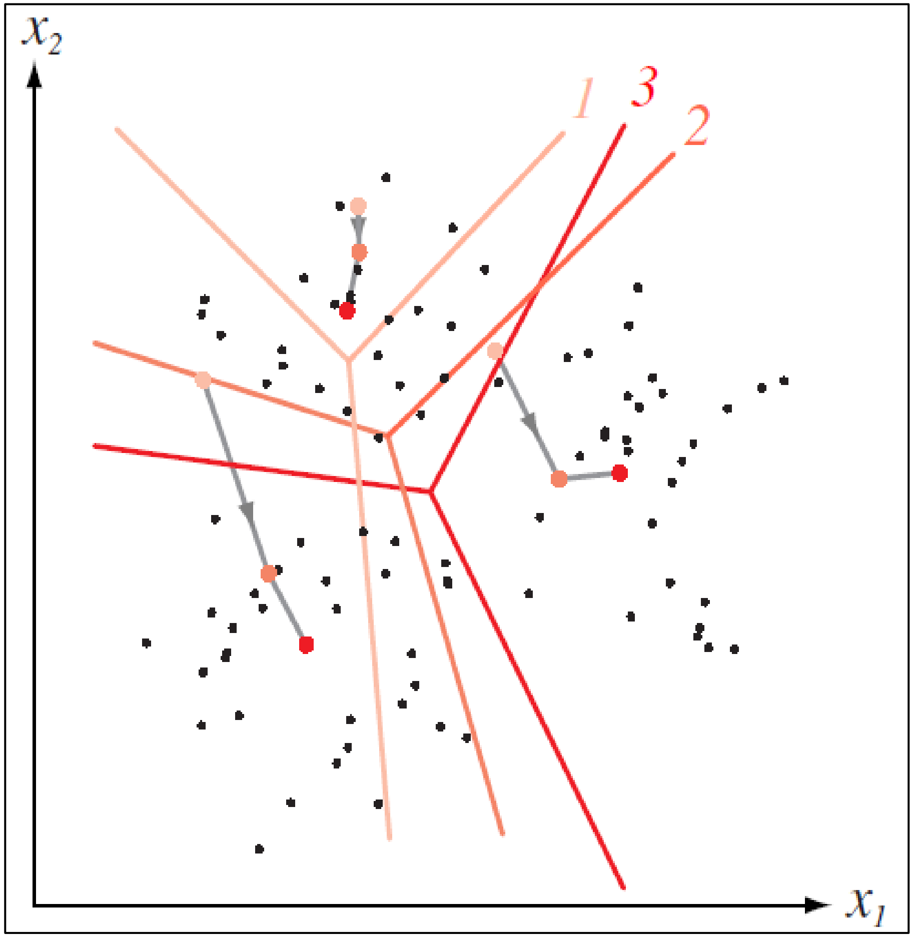
$$\vec{\mu}_k = \frac{1}{\mathcal{C}_k} \sum_{i \in \mathcal{C}_k} \vec{x}_i$$

Blue cluster gets more points

K-means Clustering: Step 5
Re-adjust cluster centers



How K-means partitions?



For each set of *K* centroids (when fixed),

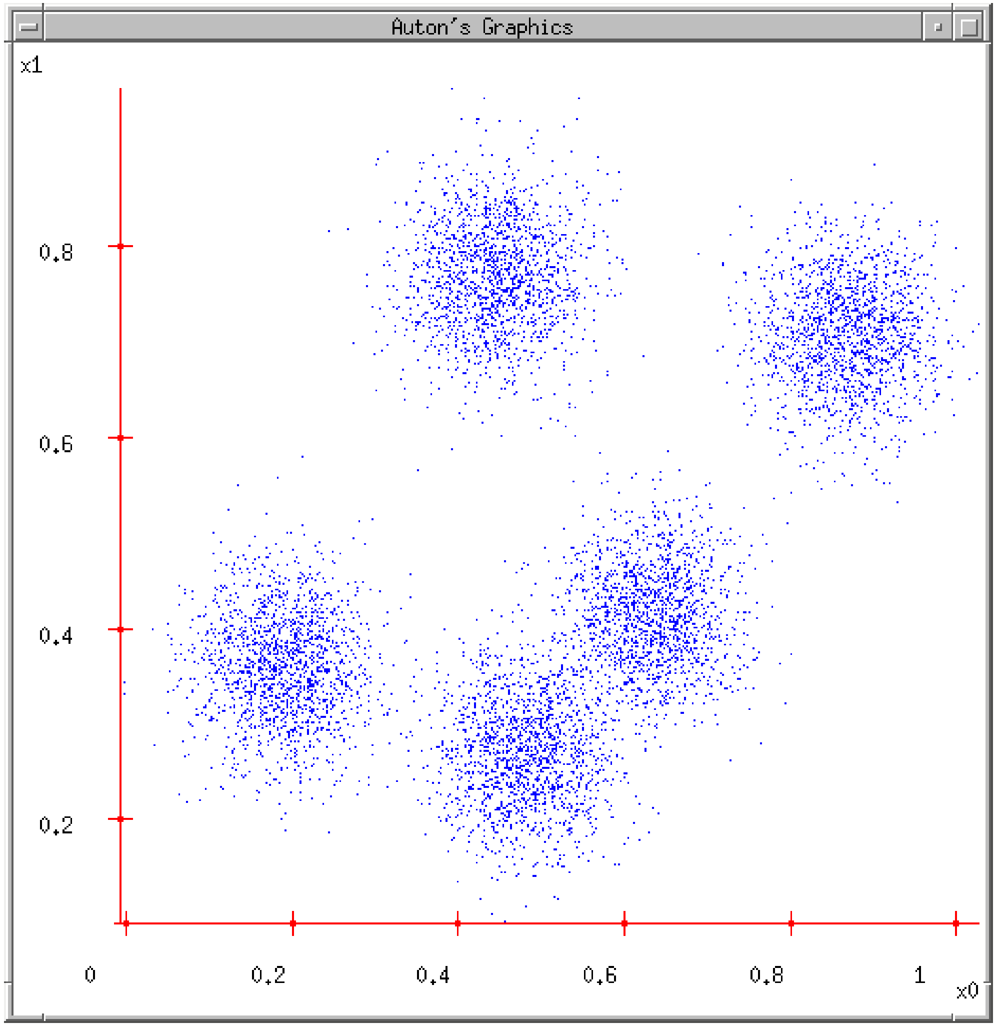
they partition the whole data space into *K* mutually exclusive subspaces to form a partition.

Changing positions of *K* centroids leads to a new partitioning.

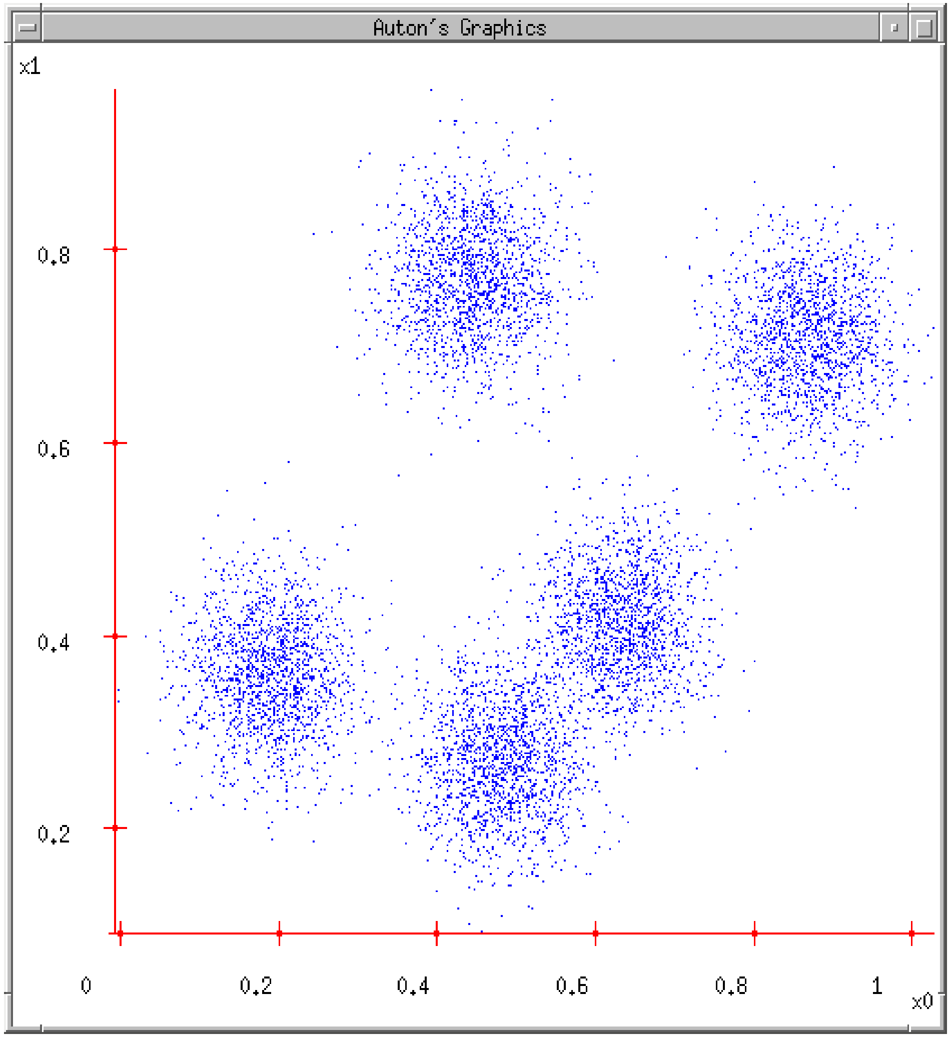
K-means: another Demo

- K-means
  - Start with a random guess of cluster centers
  - Determine the membership of each data points
  - Adjust the cluster centers

$$\vec{\mu}_k = \frac{1}{\mathcal{C}_k}\sum_{i\in\mathcal{C}_k}\vec{x}_i$$

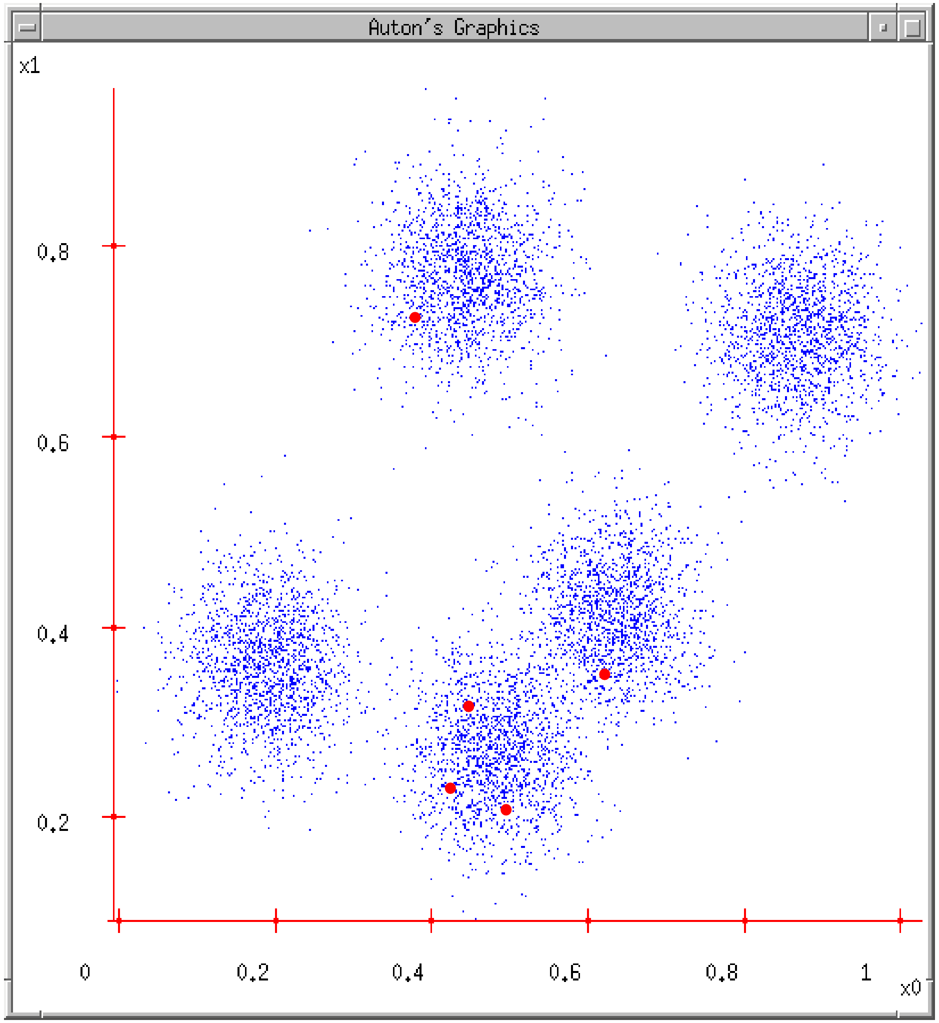


K-means: another Demo



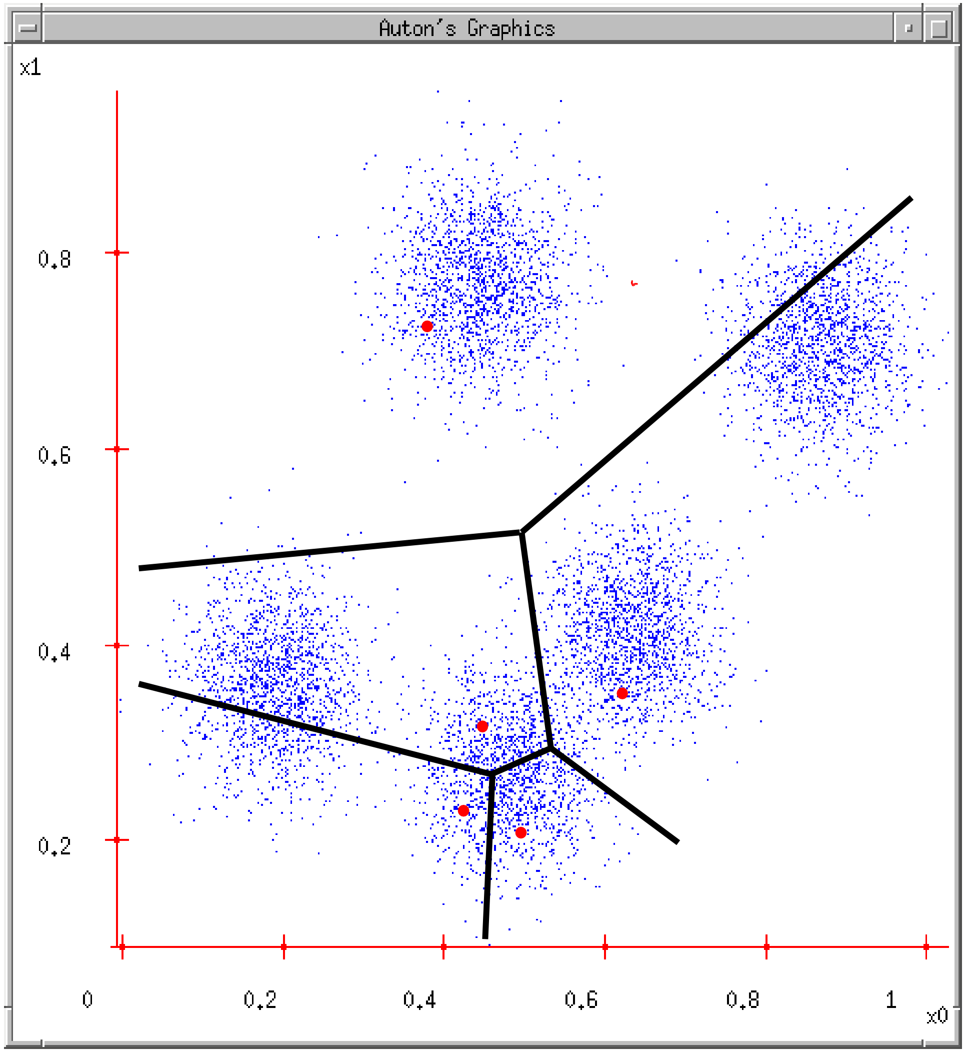
1. User set up the number of clusters they'd like. *(e.g. k=5)*

K-means: another Demo



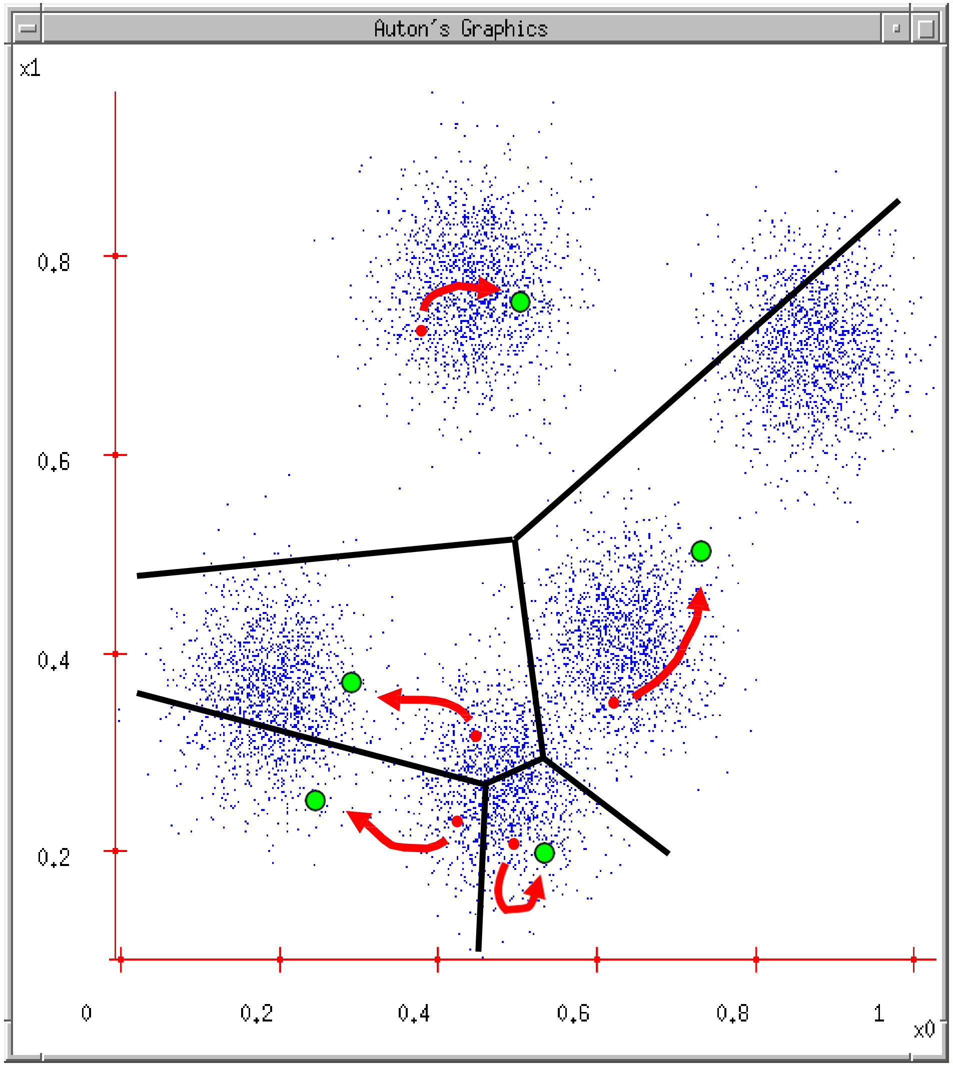
1. User set up the number of clusters they'd like. *(e.g. K=5)*
2. Randomly guess K cluster Center locations

K-means: another Demo



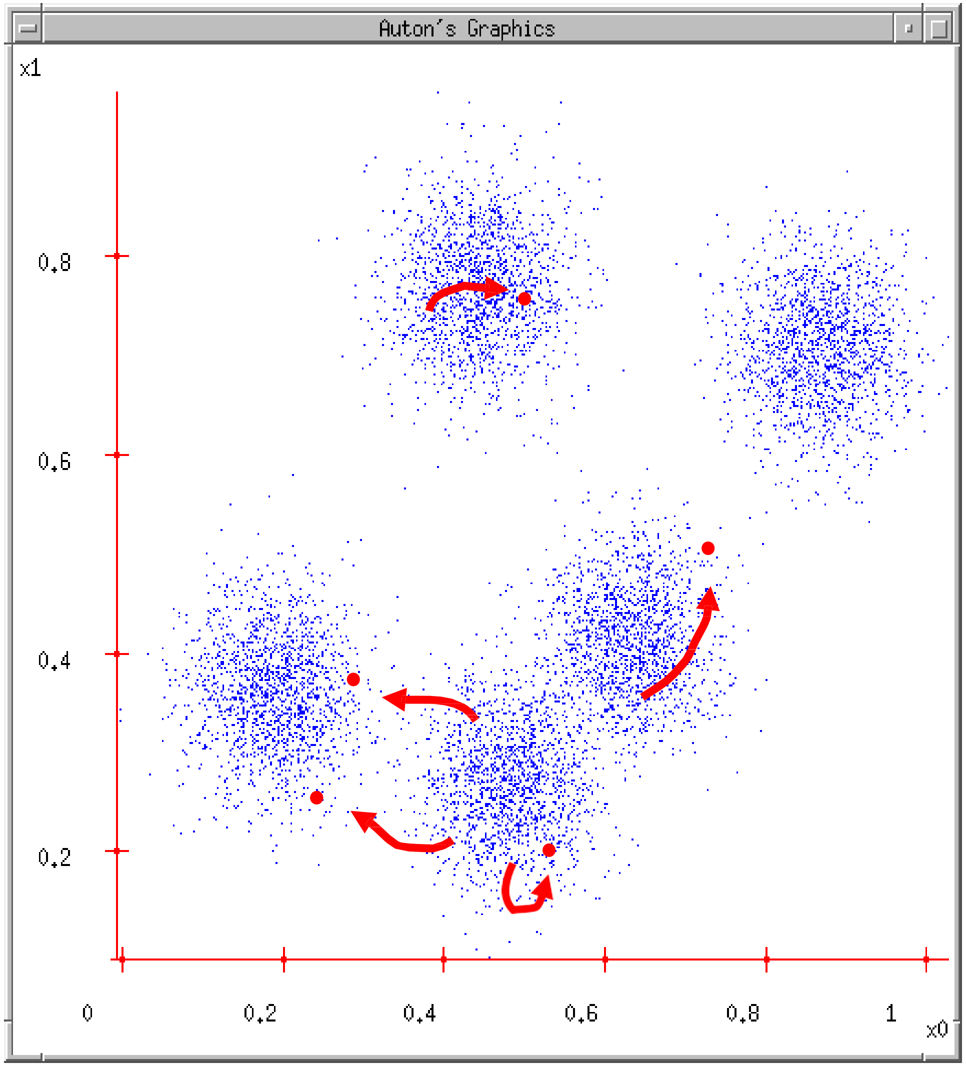
1. User set up the number of clusters they'd like. *(e.g. K=5)*
2. Randomly guess *K* cluster Center locations
3. Each data point finds out which Center it's closest to. (Thus each Center "owns" a set of data points)

K-means: another Demo



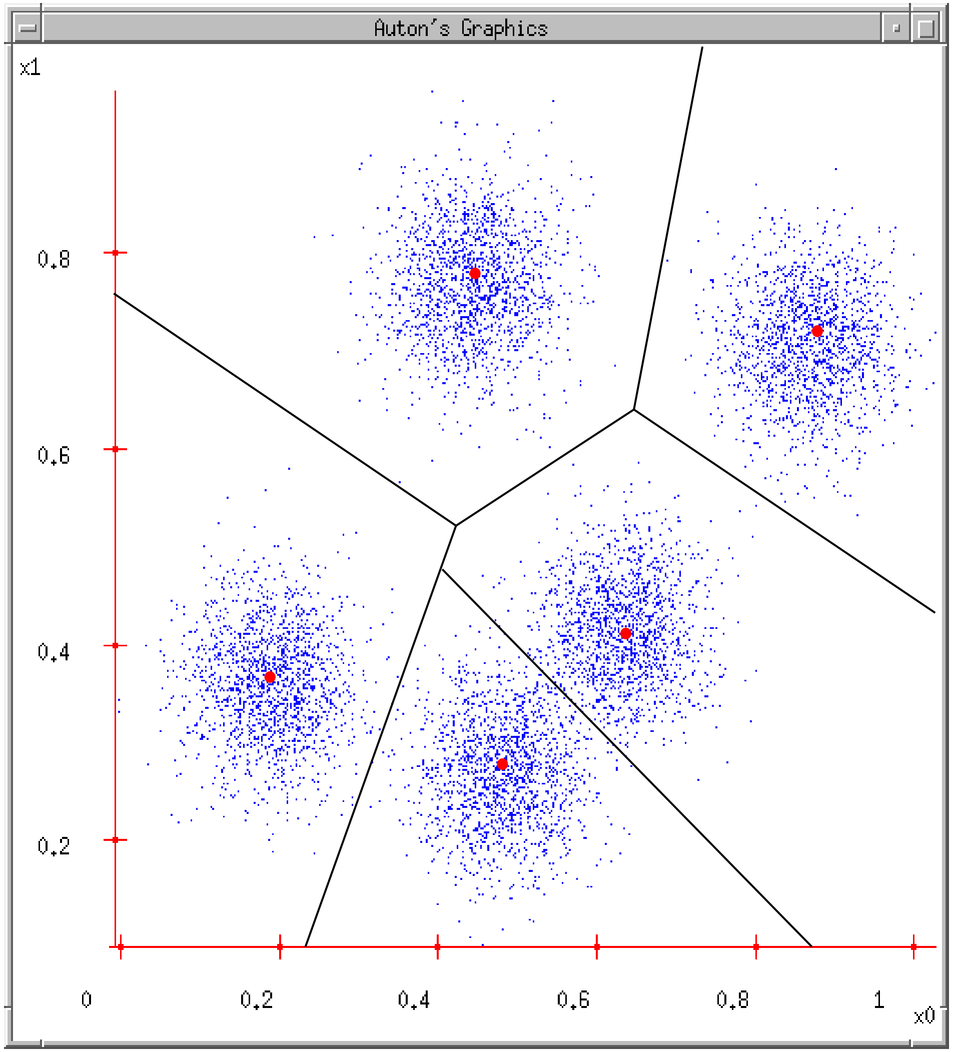
1. User set up the number of
- clusters they'd like. (e.g. K=5)
2. Randomly guess K cluster centre locations
3. Each data point finds out which centre it's closest to. (Thus each Center "owns" a set of data points)
4. Each centre finds the centroid of the points it owns

K-means: another Demo



1. User set up the number of clusters they'd like. (e.g. K=5)
2. Randomly guess K cluster centre locations
3. Each data point finds out which centre it's closest to. (Thus each centre "owns" a set of data points)
4. Each centre finds the centroid of the points it owns
5. …and jumps there

K-means: another Demo

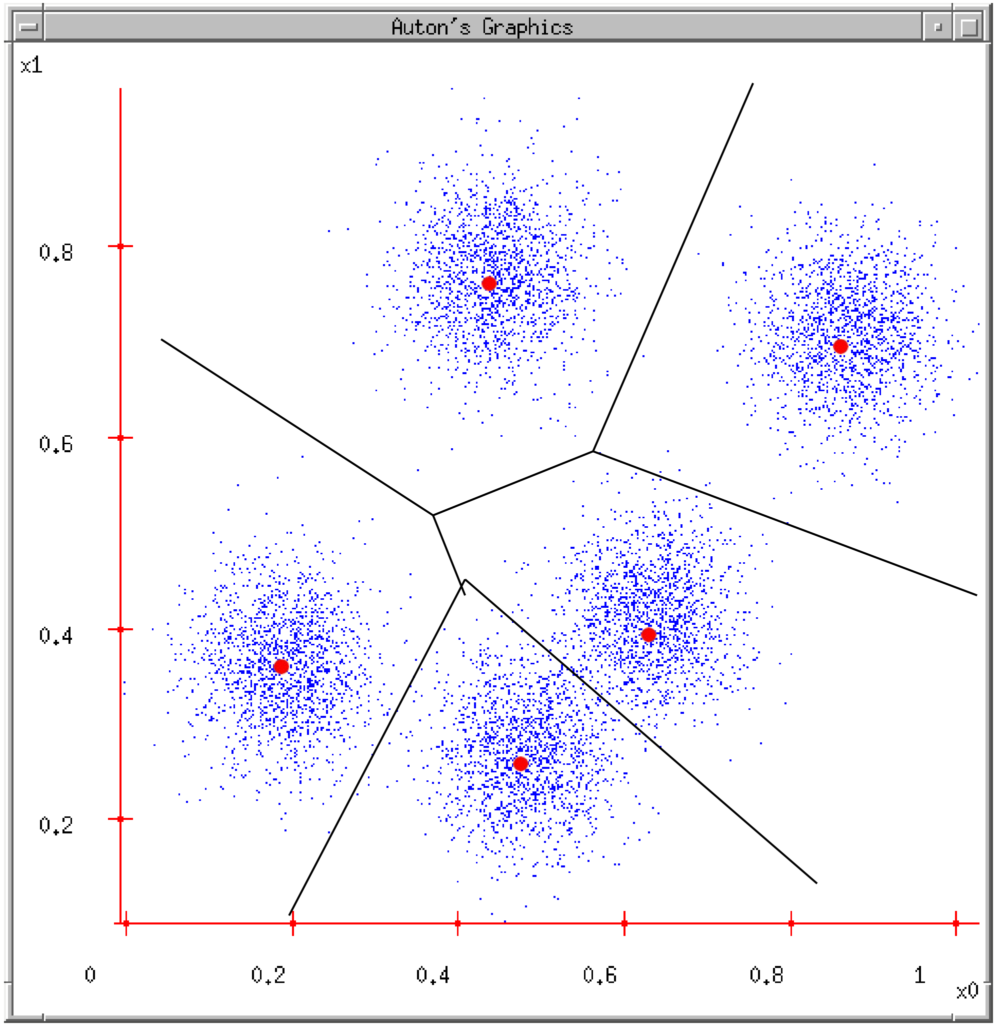


1. User set up the number of clusters they'd like. (e.g. K=5)
2. Randomly guess K cluster centre locations
3. Each data point finds out which centre it's closest to. (Thus each centre "owns" a set of data points)
4. Each centre finds the centroid of the points it owns
5. …and jumps there
6. …Repeat until terminated!

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

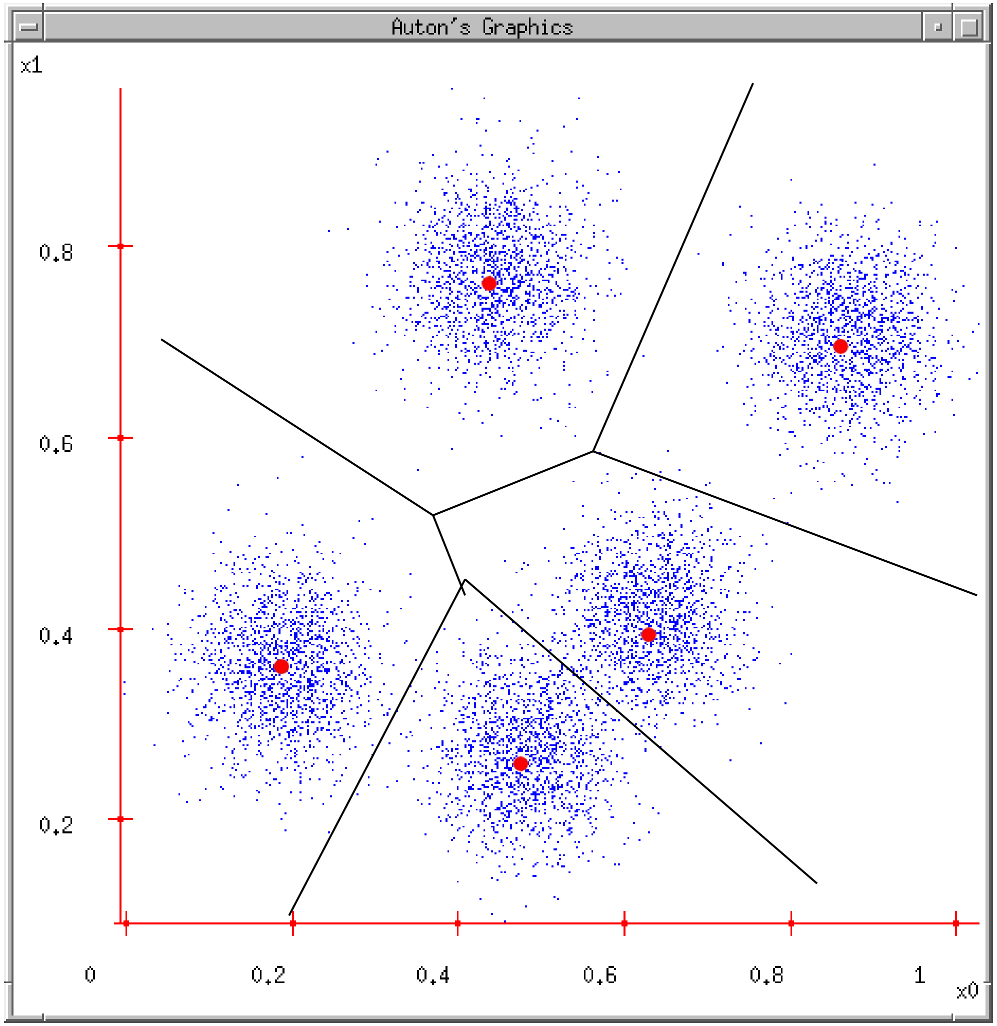
Any Computational Problem?



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

Any Computational Problem?



Time Complexity

- Computing distance between two objs is $O(p)$ where p is the dimensionality of the vectors.

- Reassigning clusters: $O(Knp)$ distance computations

- Computing centroids: Each obj gets added once to some centroid: $O(np)$.

$$\vec{\mu}_k = \frac{1}{\mathcal{C}_k} \sum_{i \in \mathcal{C}_k} \vec{x}_i$$

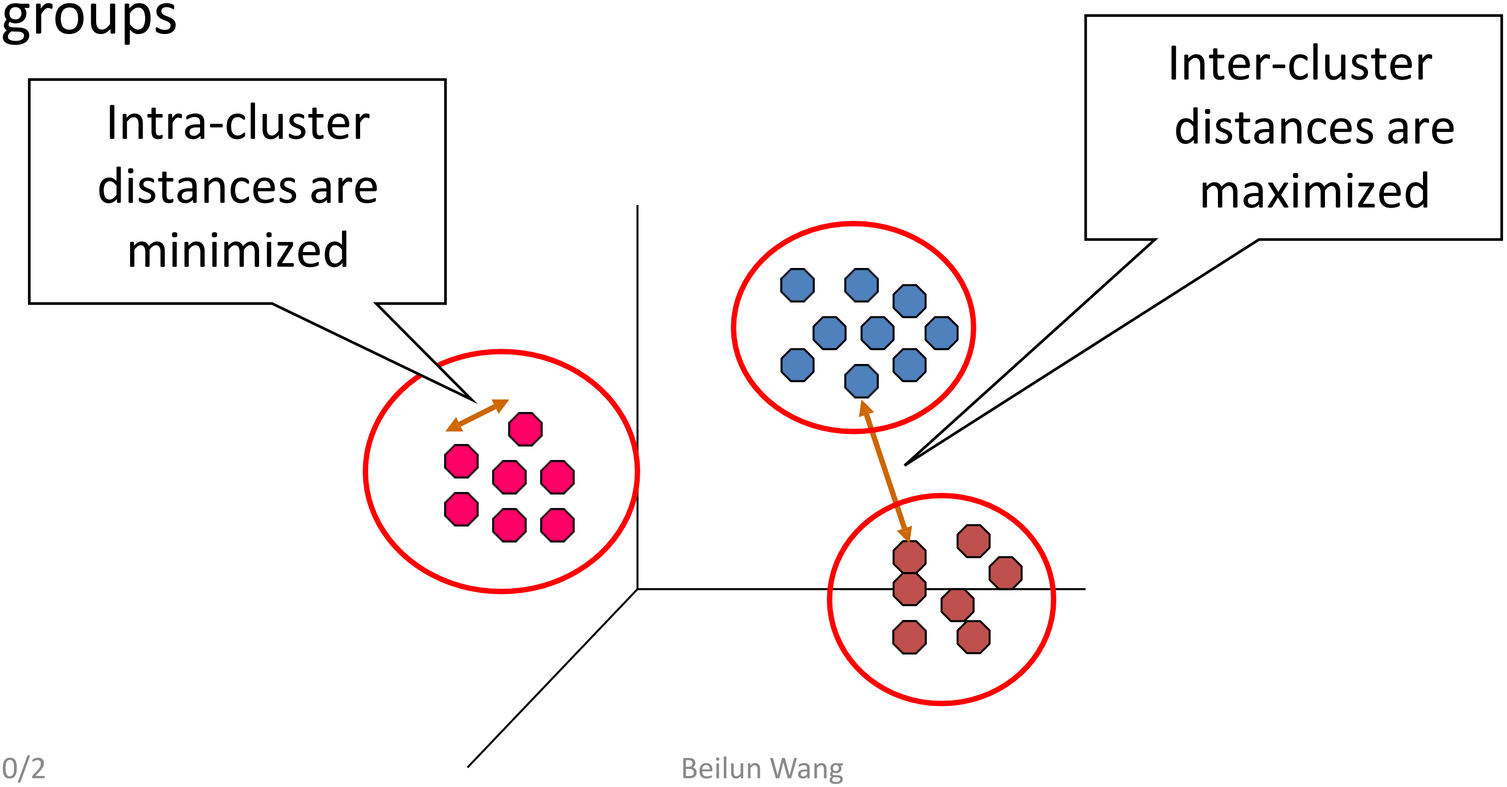
- Assume these two steps are each done once for l iterations: $O(lKnp)$.

Today

- Definition of "groupness"
- Definition of "similarity/distance"
- Clustering Algorithms
  - Hierarchical algorithms
  - Partitional algorithms
- ➡ Formal foundation and convergence
- How many clusters?

How to Find good Clustering?

- Find groups (clusters) of data points such that data points in a group will be similar (or related) to one another and different from (or unrelated to) the data points in other groups



Intra-cluster distances are minimized

Inter-cluster distances are maximized

How to Find good Clustering? E.g.

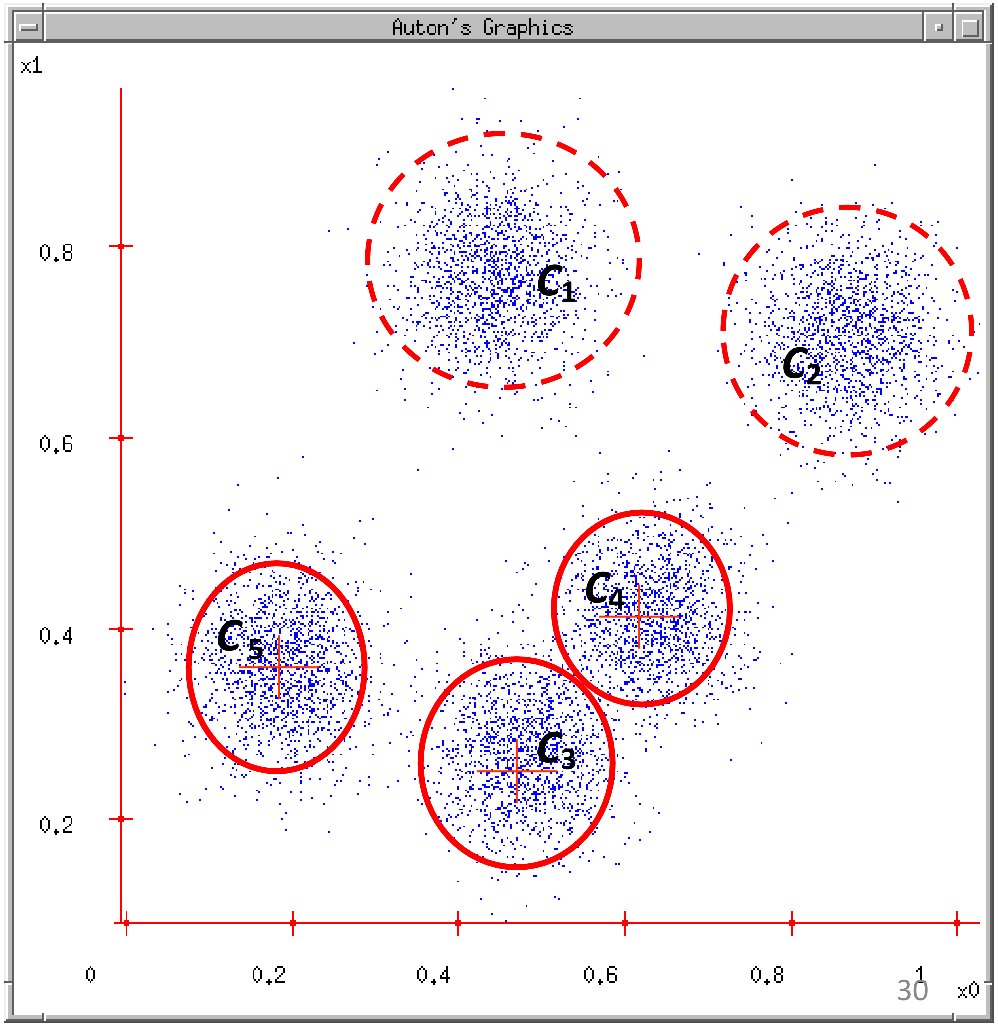
- Minimize the sum of distance within clusters

$$\underset{\{\vec{C_j},m_{i,j}\}}{argmin} \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j}(\vec{x_i}-\vec{C_j})^2$$

$$m_{i,j} = \begin{cases} 1 & \vec{x_i} \in the\ j-th\ cluster \\ 0 & \vec{x_i} \notin the\ j-th\ cluster \end{cases}$$

$$\sum_{j=1}^{K=5} m_{i,j} = 1$$

$$\rightarrow any\ \vec{x_i} \in a\ singer\ cluster$$



$$\underset{\{\overrightarrow{C_j}, m_{i,j}\}}{argmin} \sum_{j=1}^{k=5}\sum_{i=1}^{n} m_{i,j}(\overrightarrow{x_i}-\overrightarrow{C_j})^2$$

$$\rightarrow when\ given\ \{m_{i,j}\}, \mathrm{loss}(\overrightarrow{C_j}) = \sum_{j=1}^{k=5}\sum_{i=1}^{n} m_{i,j}(\overrightarrow{x_i}-\overrightarrow{C_j})^2$$

$$\frac{\partial \mathrm{loss}(\overrightarrow{C_j})}{\partial \overrightarrow{C_j}} = 0 \quad \Rightarrow \quad \overrightarrow{C_j} = \frac{\sum_{i=1}^{n} m_{i,j}\overrightarrow{x_i}}{\sum_{i=1}^{n} m_{i,j}}$$

$$\rightarrow when\ given\ \{\overrightarrow{C_j}\}, \frac{\partial \mathrm{loss}(m_{i,j})}{\partial m_{i,j}} = 0$$

$$\Rightarrow \quad m_{i,j} = \begin{cases} 1 & j = \underset{k}{argmin}\ (\overrightarrow{x_i}-\overrightarrow{C_j})^2 \\ 0 & \qquad\qquad otherwise \end{cases}$$

Iterative Optimization

$$\underset{\{\overrightarrow{C_j}, m_{i,j}\}}{argmin} \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} (\overrightarrow{x_i} - \overrightarrow{C_j})^2$$

$Memberships\ \ \{m_{i,j}\}\ and\ centers\ \{\overrightarrow{C_j}\}\ are\ correlated.$

$$Given\ centers\ \{\overrightarrow{C_j}\}, m_{i,j} = \begin{cases} 1 & j = \underset{k}{argmin}\ (\overrightarrow{x_i} - \overrightarrow{C_j})^2 \\ 0 & otherwise \end{cases}$$

$$Given\ memberships\ \ \{m_{i,j}\}, \overrightarrow{C_j} = \frac{\sum_{i=1}^{n} m_{i,j} \overrightarrow{x_i}}{\sum_{i=1}^{n} m_{i,j}}$$

Convergence

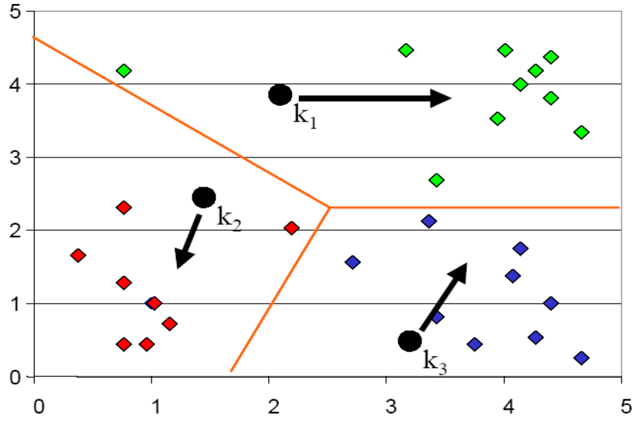
- Why should the K-means algorithm ever reach a fixed point?
  - A state in which clusters don’t change.
- K-means is a special case of a general procedure known as the Expectation Maximization (EM) algorithm.
  - EM is known to converge.
  - Number of iterations could be large.

Convergence

- Optimize the goodness measure (i.e., minimize the Loss function)
    - sum of squared distances from cluster centroid.
- Reassignment monotonically decreases the goodness measure since each vector is assigned to the closest centroid.

Seed Choice

- Results can vary based on random seed selection.



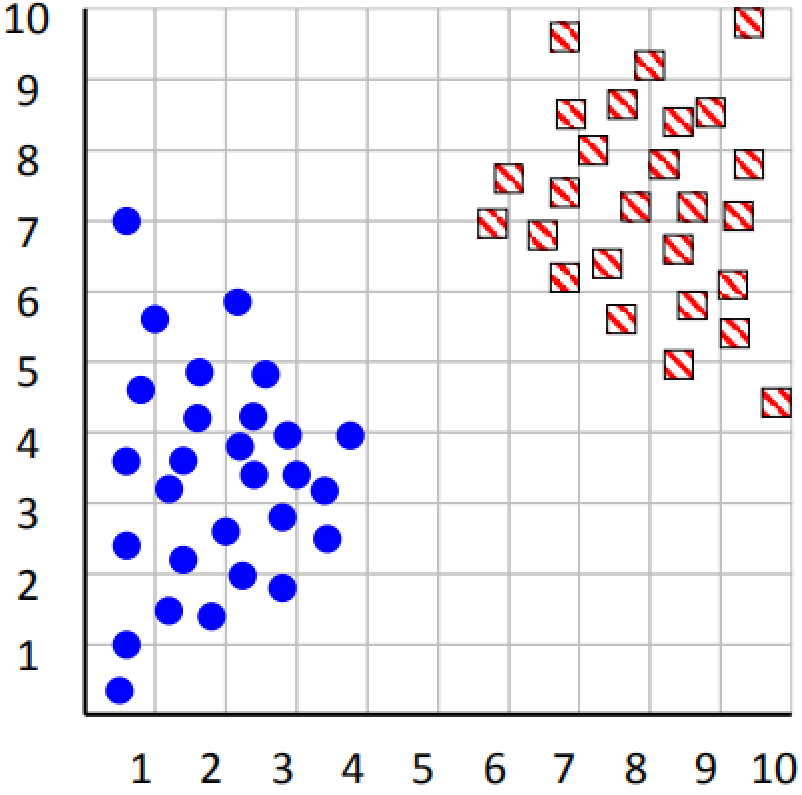
- Some seeds can result in poor convergence rate, or convergence to sub-optimal clustering.
  - Select good seeds using a heuristic (e.g., sample least similar to any existing mean)
  - Try out multiple starting points (very important!)
  - Initialize with the results of another method.

Today

- Definition of "groupness"
- Definition of "similarity/distance"
- Clustering Algorithms
  - Hierarchical algorithms
  - Partitional algorithms
- Formal foundation and convergence
- How many clusters?

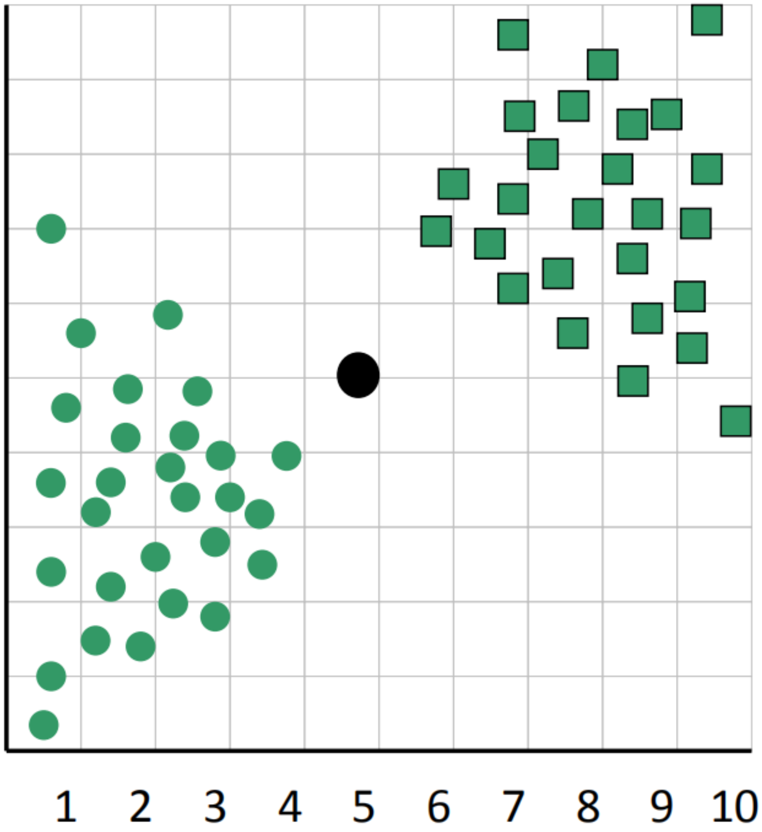
How can we tell the right number of clusters?

- In general, this is a unsolved problem. However there exist many approximate methods.



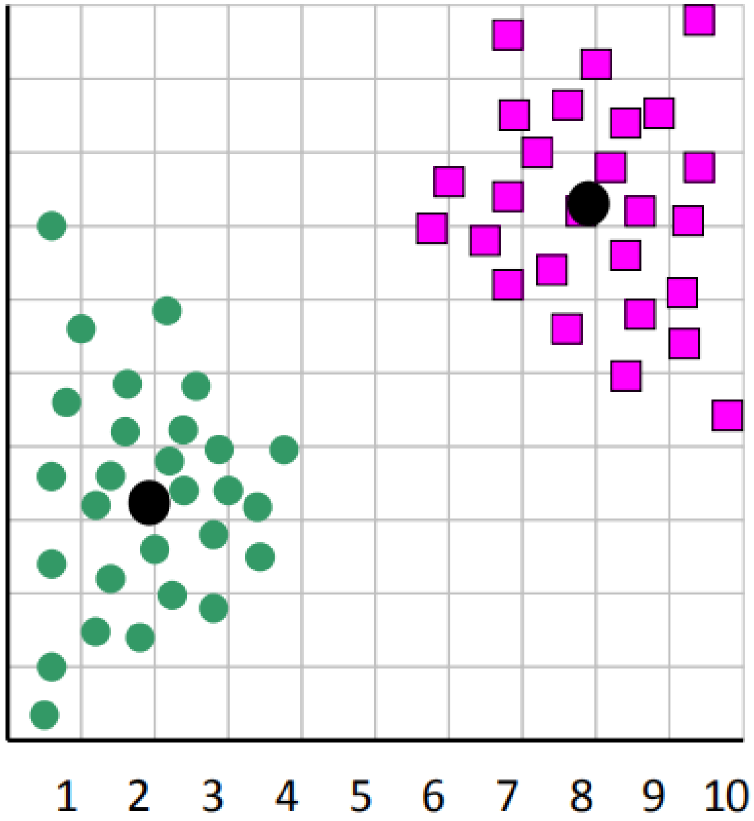
$$\underset{\{\overrightarrow{C_j}, m_{i,j}\}}{argmin} \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} (\overrightarrow{x_i} - \overrightarrow{C_j})^2$$

When k = 1, the objective function is 873.0



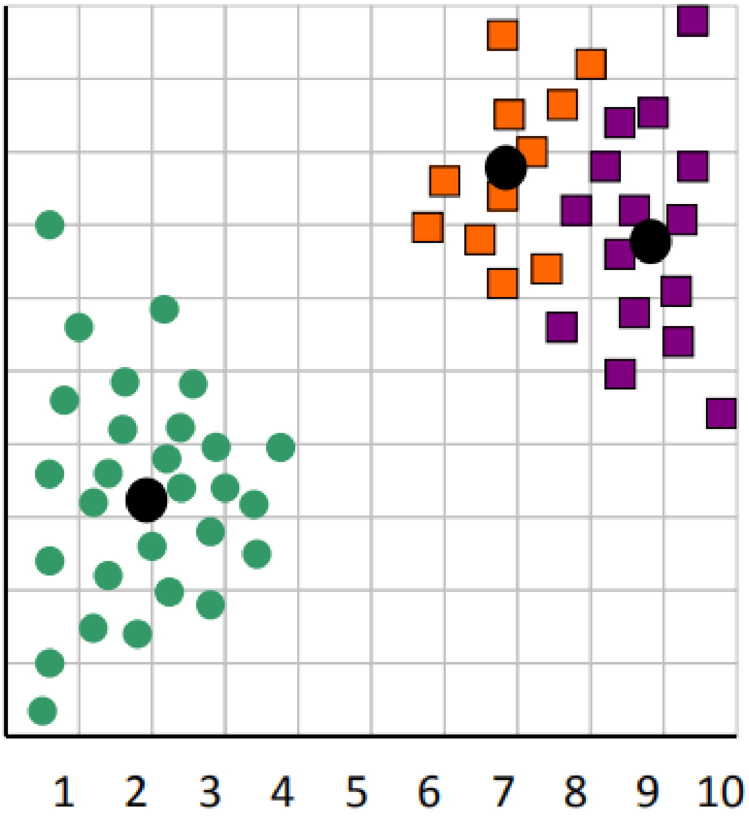
$$\underset{\{\overrightarrow{C_j}, m_{i,j}\}}{argmin} \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} (\overrightarrow{x_i} - \overrightarrow{C_j})^2$$

When k = 2, the objective function is 173.1

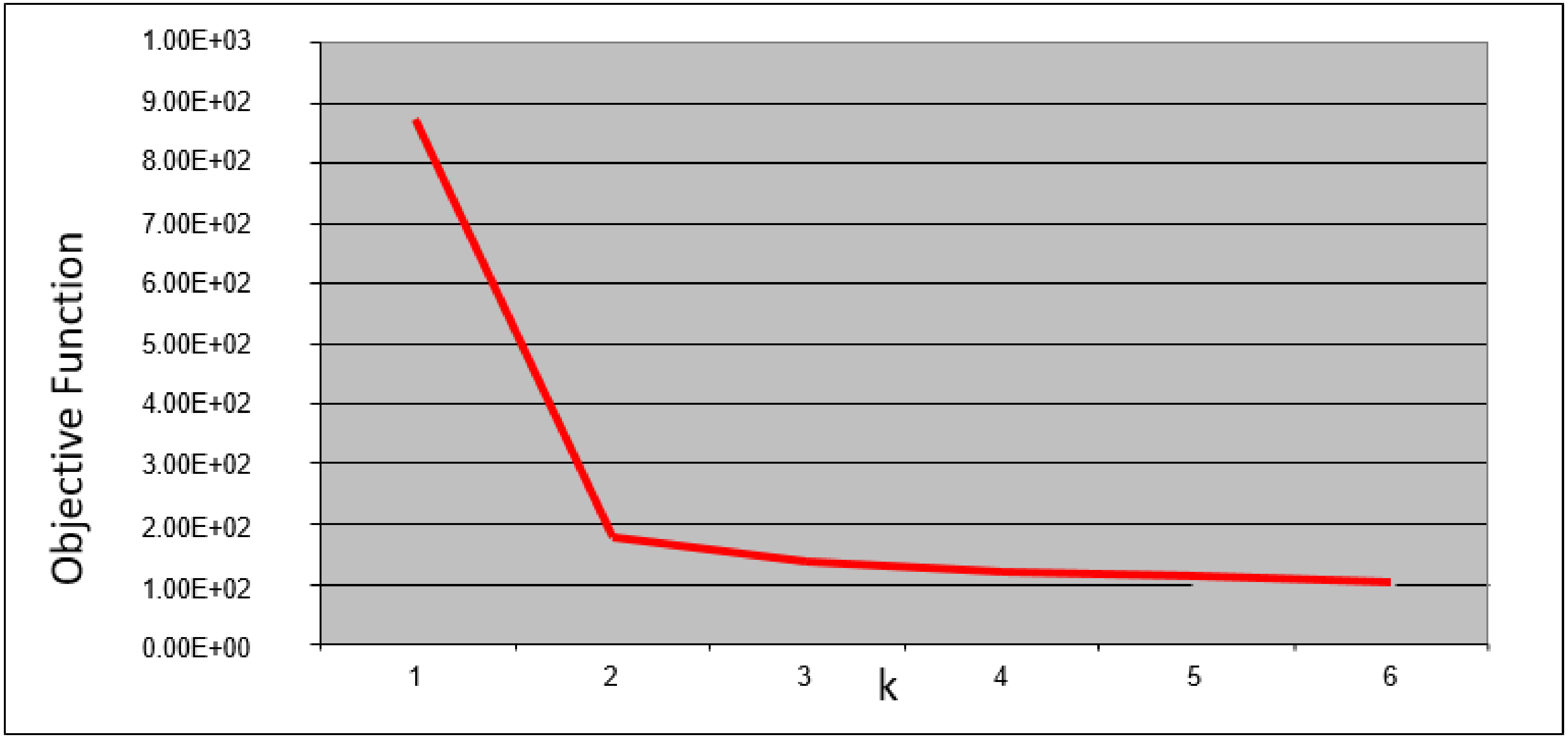


$$\underset{\{\overrightarrow{C_j}, m_{i,j}\}}{argmin} \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} (\overrightarrow{x_i} - \overrightarrow{C_j})^2$$

When k = 3, the objective function is 133.6



We can plot the objective function values for k equals 1 to 6… The abrupt change at k = 2, is highly suggestive of two clusters in the data. This technique for determining the number of clusters is known as “knee finding” or “elbow finding”.



Note that the results are not always as clear cut as in this toy example.

What Is A Good Clustering?

- Internal criterion: A good clustering will produce high quality clusters in which:
  - the intra-cluster similarity is high
  - the inter-cluster similarity is low
  - The measured quality of a clustering depends on both the data representation and the similarity measure used

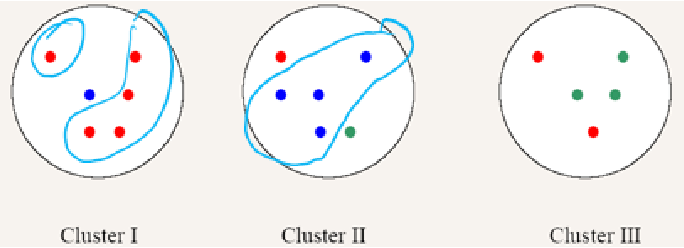
- External criteria for clustering quality
  - Quality measured by its ability to discover some or all of the hidden patterns or latent classes in gold standard data
  - Assesses a clustering with respect to ground truth
  - Example:
    - Purity
    - entropy of classes in clusters (or mutual information between classes and clusters)

External Evaluation of Cluster Quality e.g. using purity

- Simple measure: purity, the ratio between the dominant class in the cluster and the size of cluster
  - Assume data samples with C gold standard classes/groups, while the clustering algorithms produce K clusters, $\omega_1, \omega_2, \ldots, \omega_K$ with n_i members.

$$Purity(w_i) = \frac{1}{n_i} \max_j (n_{ij}) \quad j \in C$$

- Example



Cluster I: Purity = 1/6 (max(5, 1, 0)) = 5/6

Cluster II: Purity = 1/6 (max(1, 4, 1)) = 4/6

Cluster III: Purity = 1/5 (max(2, 0, 3)) = 3/5

References

- https://qiyanjun.github.io/2019f-UVA-CS6316-MachineLearning/
- Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.
- Big thanks to Prof. Eric Xing @ CMU for allowing me to reuse some of his slides
- Big thanks to Prof. Ziv Bar-Joseph @ CMU for allowing me to reuse some of his slides
- clustering slides from Prof. Rong Jin @ MSU

Thanks for listening