Data mining & Machine Learning

CS 373
Purdue University

Dan Goldwasser
dgoldwas@purdue.edu

Today's Lecture

Clustering

- *The data we get is often too complicated simply "eyeball"*
- *Clustering helps summarize the data*
- *Without external supervision --*

How can we identify structure in complex data?

Structure in data? What does that mean?

Structure *in data?*

What does that mean?

Descriptive models

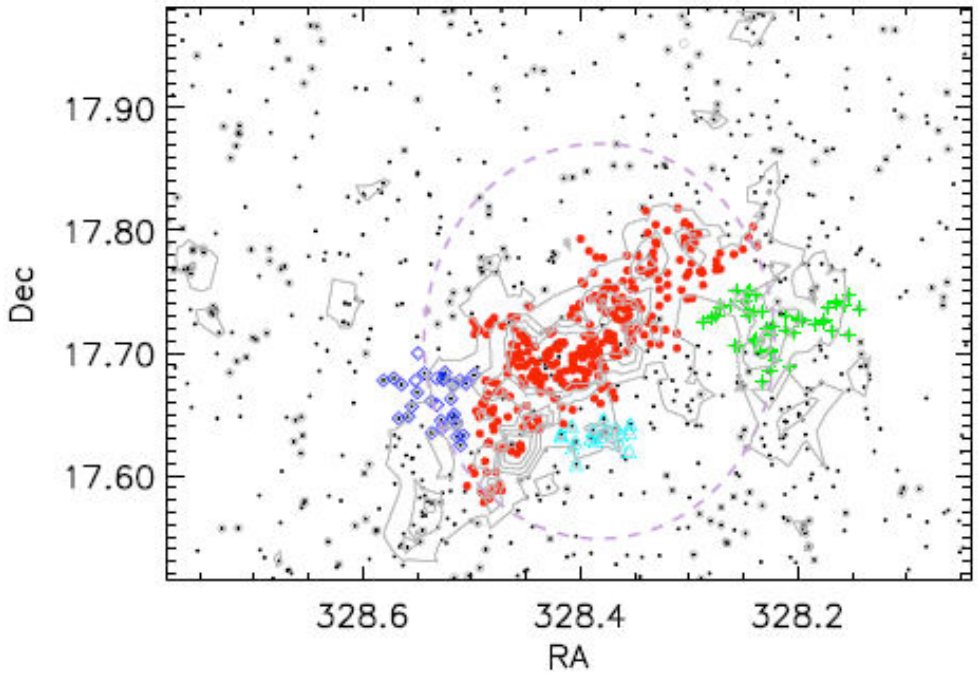
- Descriptive models summarize the data
  - Global summary
  - Model main features of the data
- **Two main approaches:**
  - Cluster analysis (*we will focus mostly on this topic*)
  - Density estimation

Modeling task

- **Data representation**: training set of ***x****(i) instances*
- **Task**—depends on approach
  - **Clustering**: partition the instances into groups of similar instances
  - **Density estimation**: based on observed instances, estimate an unobserved probability density function

Cluster analysis

- Decompose or partition instances into groups s.t.:
  - **Intra**-group similarity is ***high***
  - **Inter**-group similarity is ***low***
- *Measure of distance/similarity is crucial*



Cluster analysis

- **Huge body of work!**
  - Also known as *unsupervised learning*, segmentation, etc.
- Difficult to evaluate success (*why*?)

was it an issue in the supervised setting?

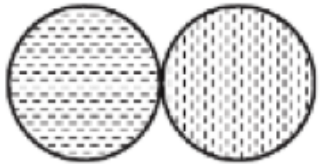
  - If goal is to find "interesting" clusters, then it is difficult to quantify
  - If goal is to find "similar" clusters, then success depends on distance measure (circular)

What makes a “good” cluster?

Well separated clusters

(a) Well-separated clusters. Each point is closer to all of the points in its cluster than to any point in another cluster.

Center-based

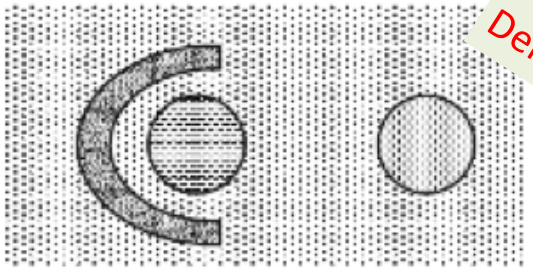


(b) Center-based clusters. Each point is closer to the center of its cluster than to the center of any other cluster.

Contiguity-based

(c) Contiguity-based clusters. Each point is closer to at least one point in its cluster than to any point in another cluster.

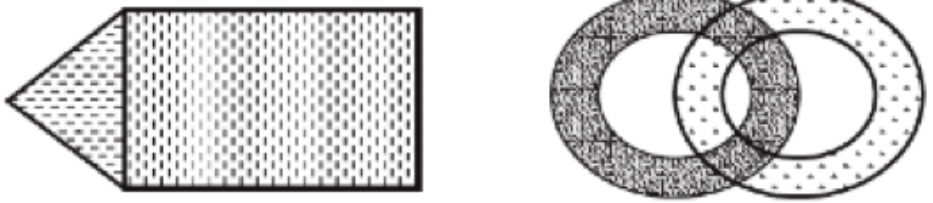
Density-based



(d) Density-based clusters. Clusters are regions of high density separated by regions of low density.

What makes a "good" cluster?

Conceptual-
Points are arranged according to a global property.



(e) Conceptual clusters. Points in a cluster share some general property that derives from the entire set of points. (Points in the intersection of the circles belong to both.)

Figure 8.2. Different types of clusters as illustrated by sets of two-dimensional points.

Application examples

- **Marketing**: discover distinct groups in customer base to develop targeted marketing programs
- **Land use**: identify areas of similar use in an earth observation database to understand geographic similarities
- **City-planning**: group houses according to house type, value, and location to identify “neighborhoods”
- **Earthquake studies**: Group observed earthquakes to see if they cluster along continent faults

Clustering example: Image segmentation

[Slide from James Hayes, adapted from D.Sontag

Clustering example: Image segmentation

Question: *is this the same as classification?*

[Slide from James Hayes, adapted from D.Sontag

Clustering example: Color quantization

Clustering Example: Topic clusters

"The recent rise in the markets"

"Inflation grows by 3% annually.."

""Unemployment rates drop by 5%"

"Chicago Cubs win world series"

"This week on football news"

"LeBron James signed a new contract"

Clustering algorithms

- **Types**:
  - Partition-based methods
  - Hierarchical clustering
    - Agglomerative: “bottom up” (merge clusters)
    - Divisive: “top down” (split clusters)
  - Probabilistic model-based methods
- **Different algorithms find clusters of different “shapes”**
  - Appropriate shape will depend on application, match method to objectives

Algorithm examples

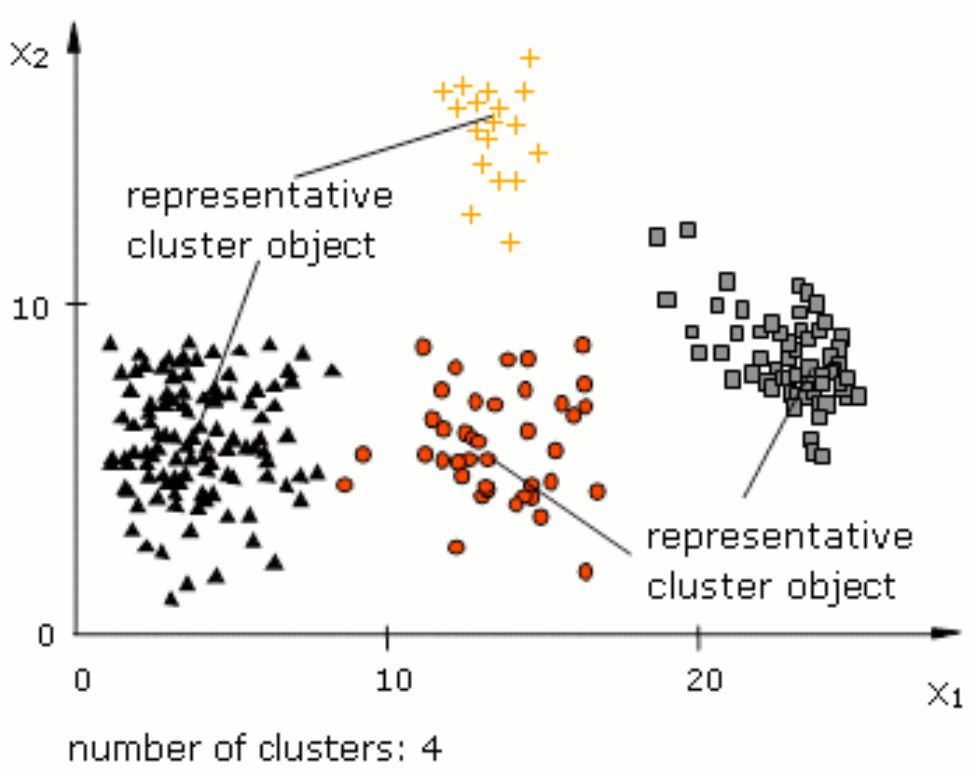
- K-means clustering (partition-based)
- Spectral clustering (hierarchical-divisive)
- Mixture models (probabilistic model-based)

Partition-based clustering

Partition-based

- **Input**: data D={**x**(1),**x**(2),...,**x**(n)}
- **Output**: k clusters C={C_1,...,C_k} such that each x(i) is assigned to a unique C_j
- **Evaluation**: Score(C,D) is maximized/minimized
- **Combinatorial optimization**: search among n^k allocations of n objects into k classes to maximize score function
- *Exhaustive search is intractable*
- Most approaches use iterative improvement algorithms

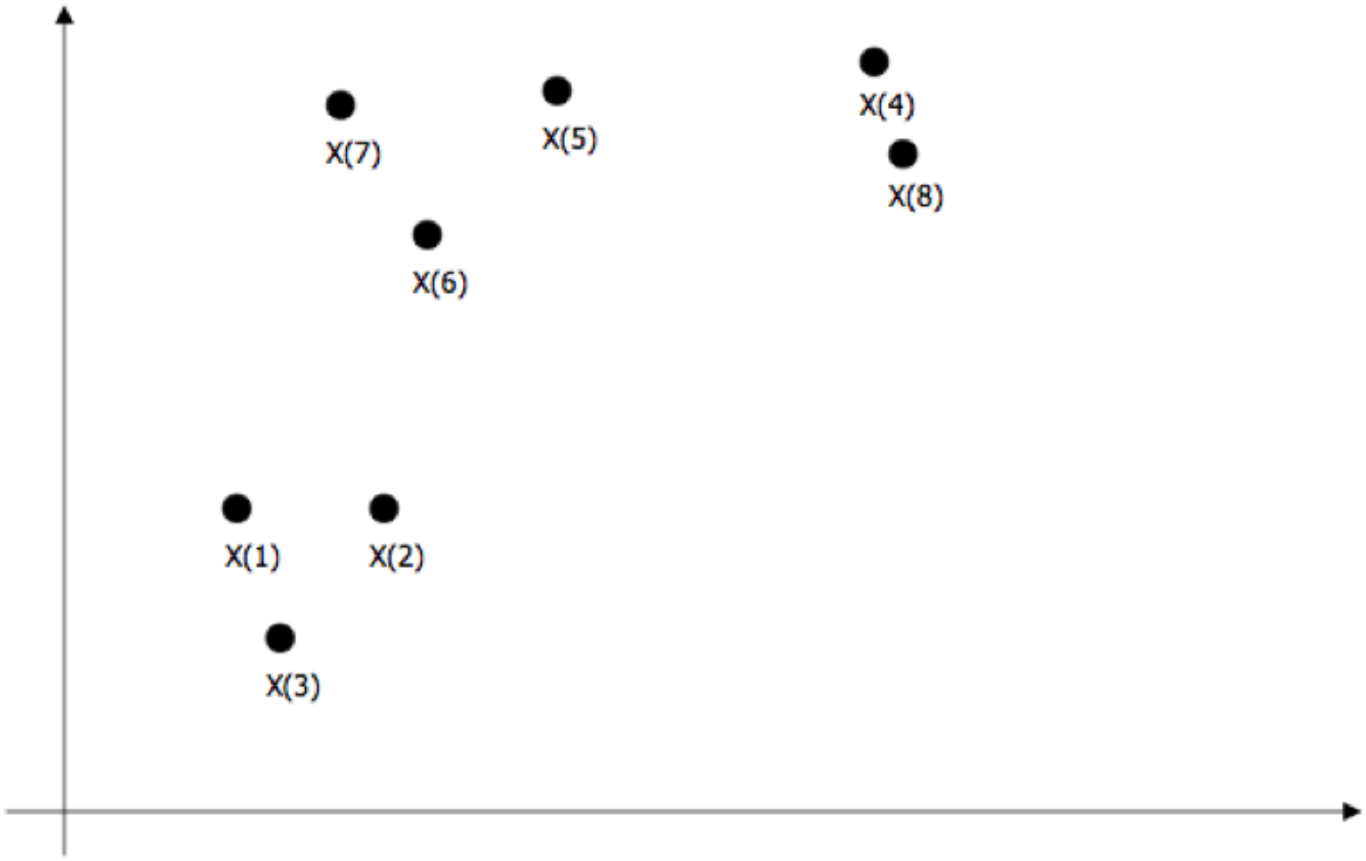
Example: K-means

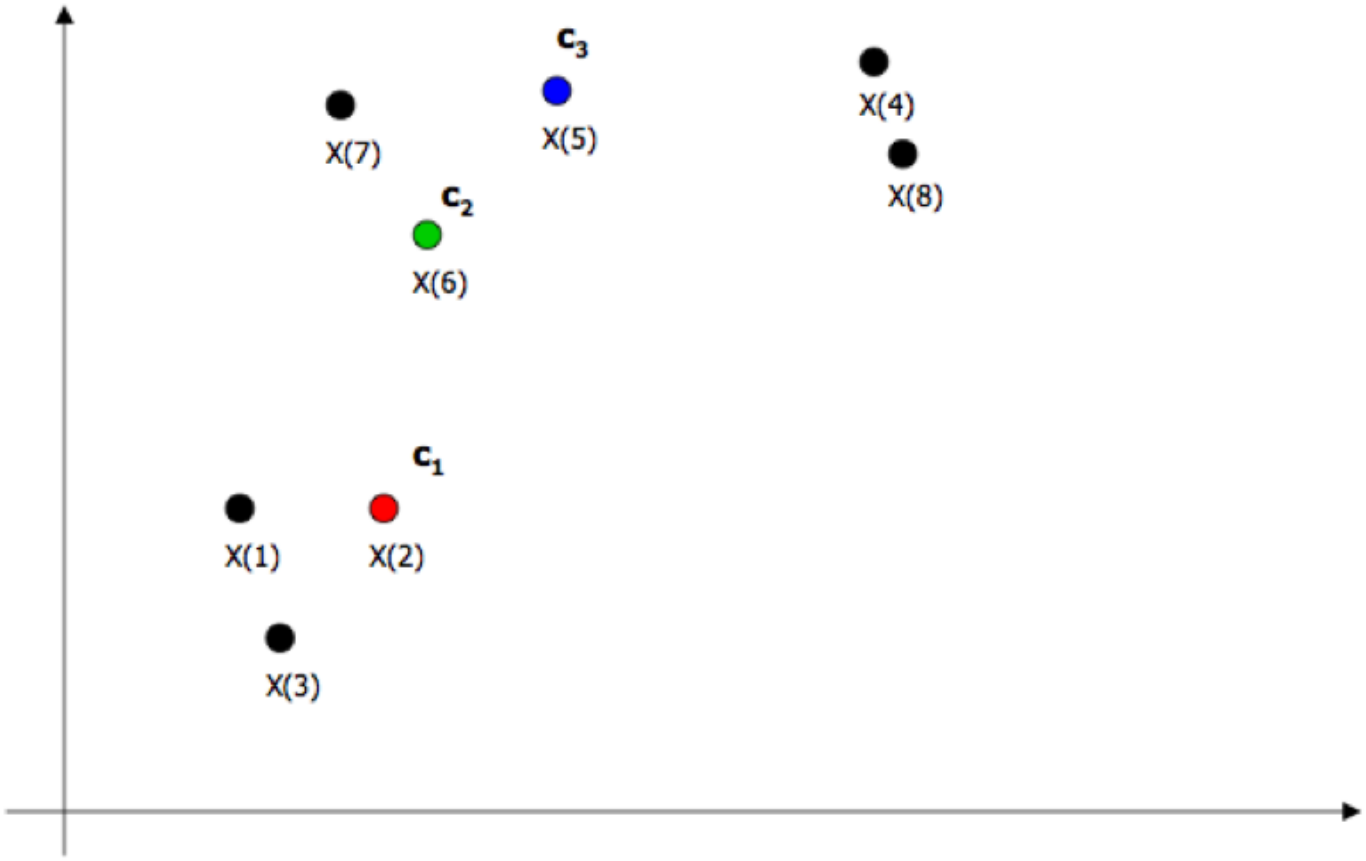


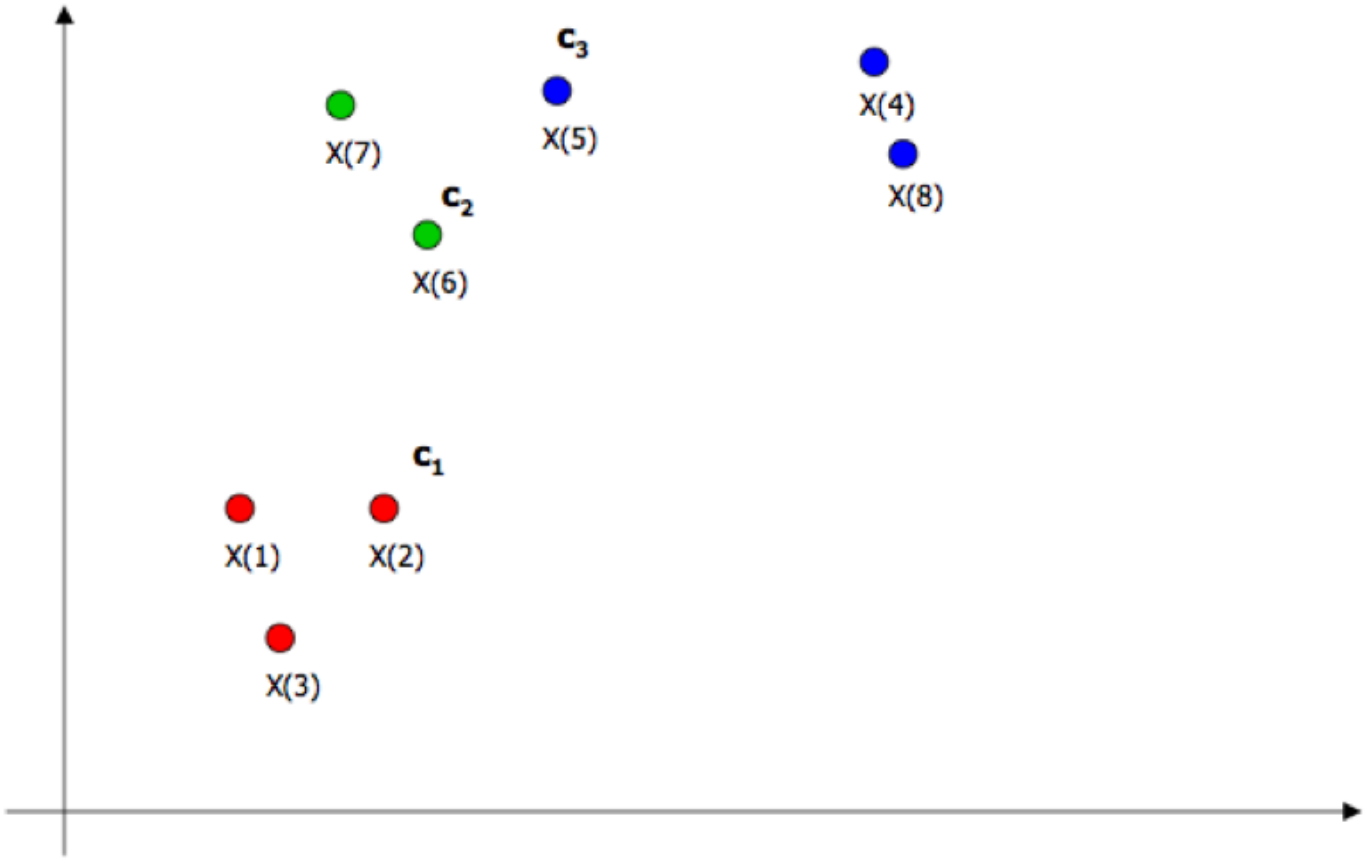
Groups represented by *canonical* item description(s)

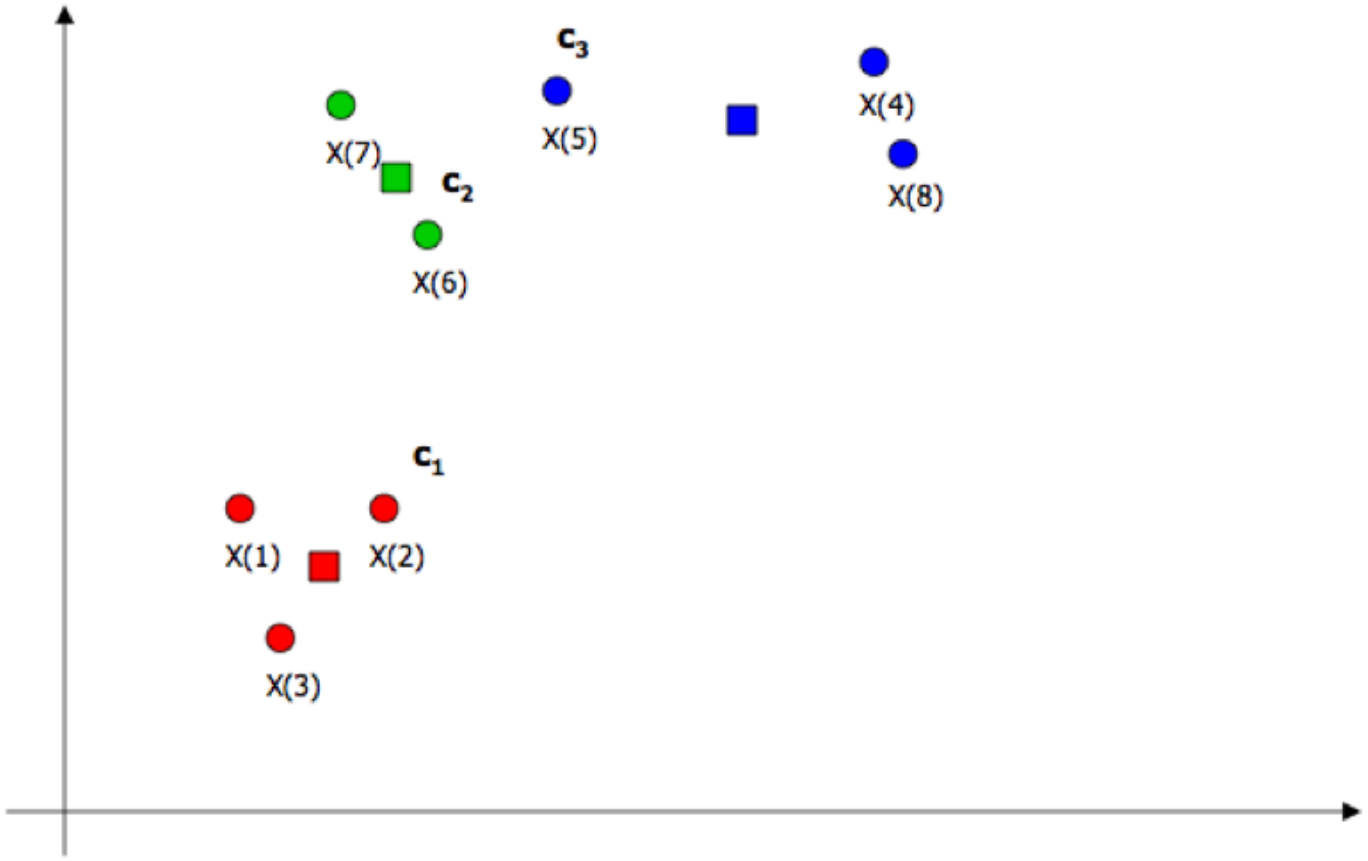
Example: K-means

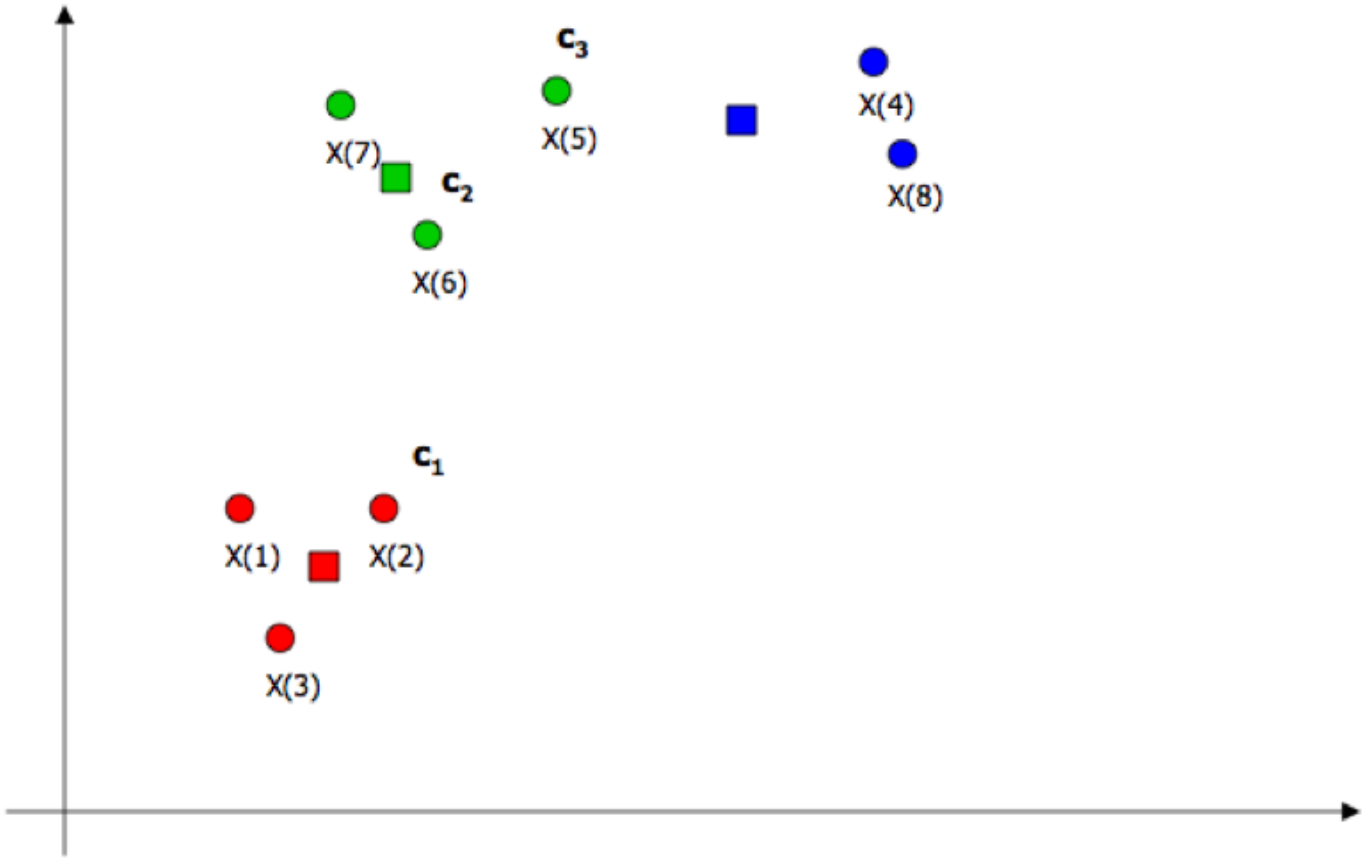
- **Algorithm idea:**
  - Start with k randomly chosen **centroids**
    - **Centroids** characterize the cluster
  - Repeat until no changes in assignments
    - Assign instances to closest centroid
    - Recompute cluster centroids

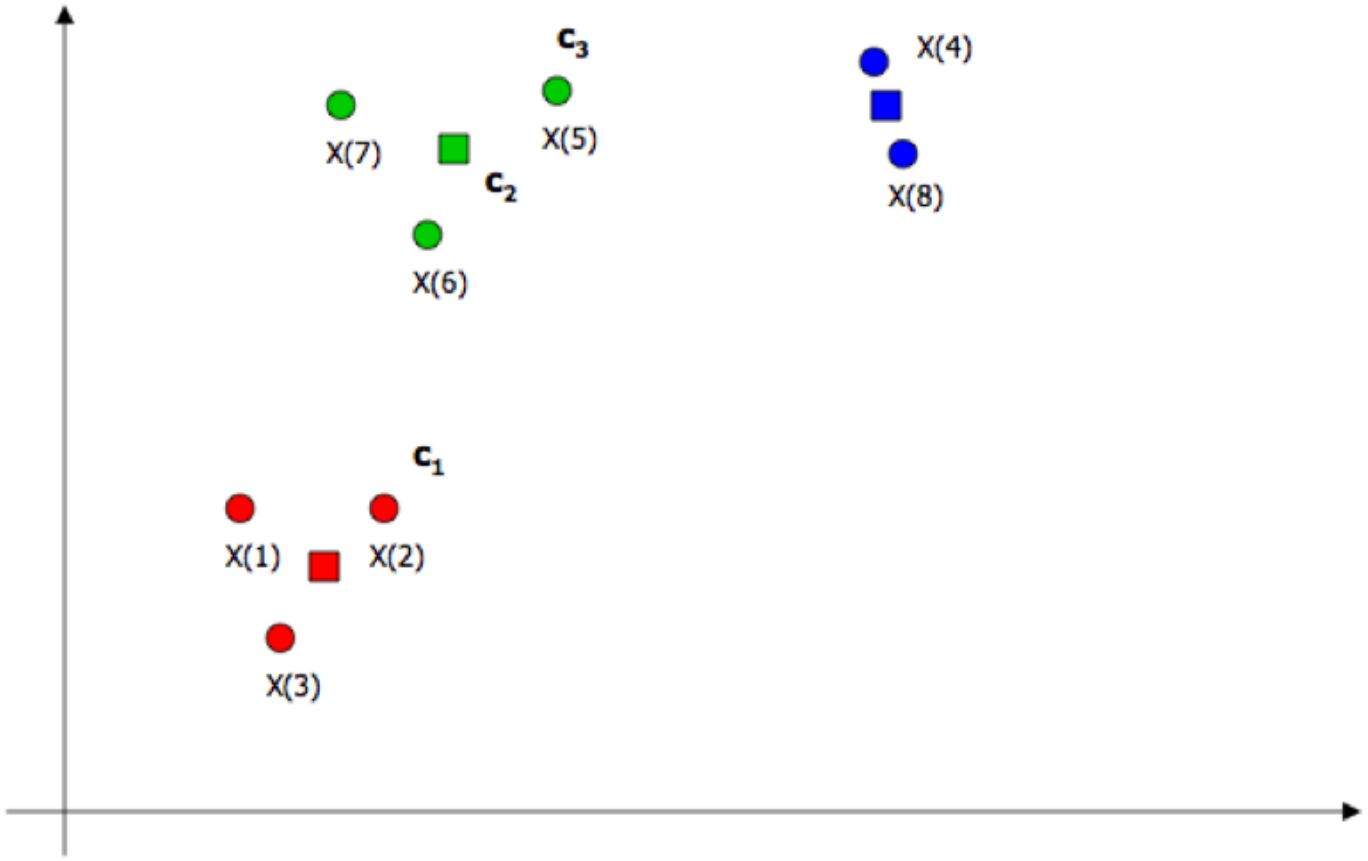

X(7)
X(5)
X(4)
X(8)
X(6)
X(1)
X(2)
X(3)


c3
X(7)
X(5)
X(4)
X(8)
c2
X(6)
c1
X(1)
X(2)
X(3)


c3
X(4)
X(5)
X(7)
X(8)
c2
X(6)
c1
X(1)
X(2)
X(3)


c3
X(4)
X(5)
X(7)
c2
X(8)
X(6)
c1
X(1)
X(2)
X(3)


c3
X(4)
X(5)
X(7)
c2
X(8)
X(6)
c1
X(1)
X(2)
X(3)


c3
X(4)
X(5)
X(7)
c2
X(8)
X(6)
c1
X(1)
X(2)
X(3)

Clustering score functions

- **Goal**:
  - ***Compact clusters***: minimize *within* cluster distance (wc)
  - ***Separated clusters***: maximize *between* cluster distance (bc)

- **Score(C,D) = *f(wc(C), bc(C))***
  - Score measures quality of clustering C for dataset D
  - *Many score functions are a combination of within-cluster (wc) and between-cluster (bc) distance measures*

Clustering score functions

- **Score(C,D) = *f(wc(C), bc(C))***

cluster centroid: $$r_k = \frac{1}{n_k} \sum_{x(i) \in C_k} x(i)$$

between-cluster distance: $$bc(C) = \sum_{1 \le j < k \le K} d(r_j, r_k)^2$$

within-cluster distance: $$wc(C) = \sum_{k=1}^{K} wc(C_k) = \sum_{k=1}^{K} \sum_{x(i) \in C_k} d(x(i), r_k)^2$$

Clustering search

- Most learning algorithms involve iterative search over assignments due to score functions which require combinatorial optimization

K-means clustering

Algorithm 2.1 The `k-means` algorithm

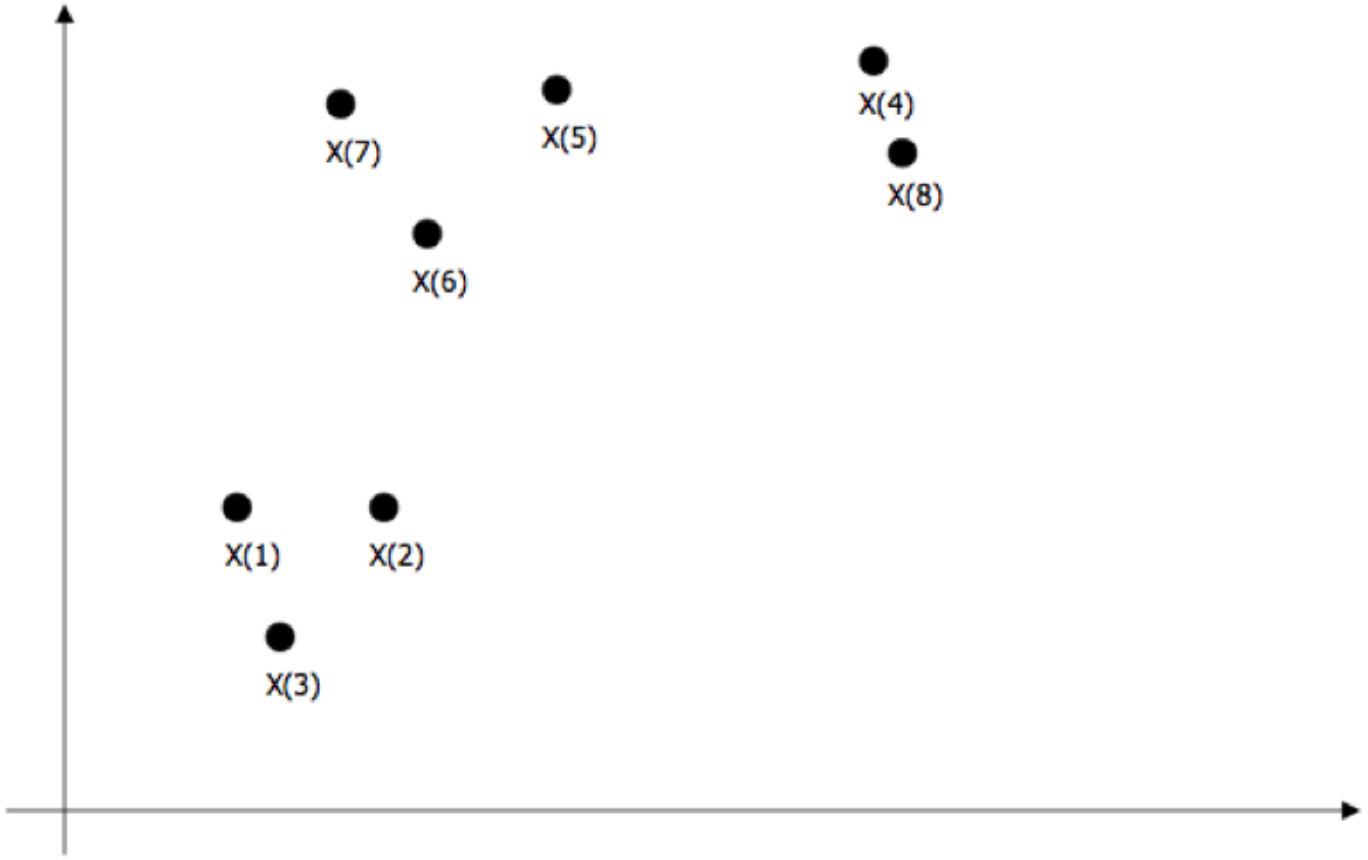
Input: Dataset D, number clusters k

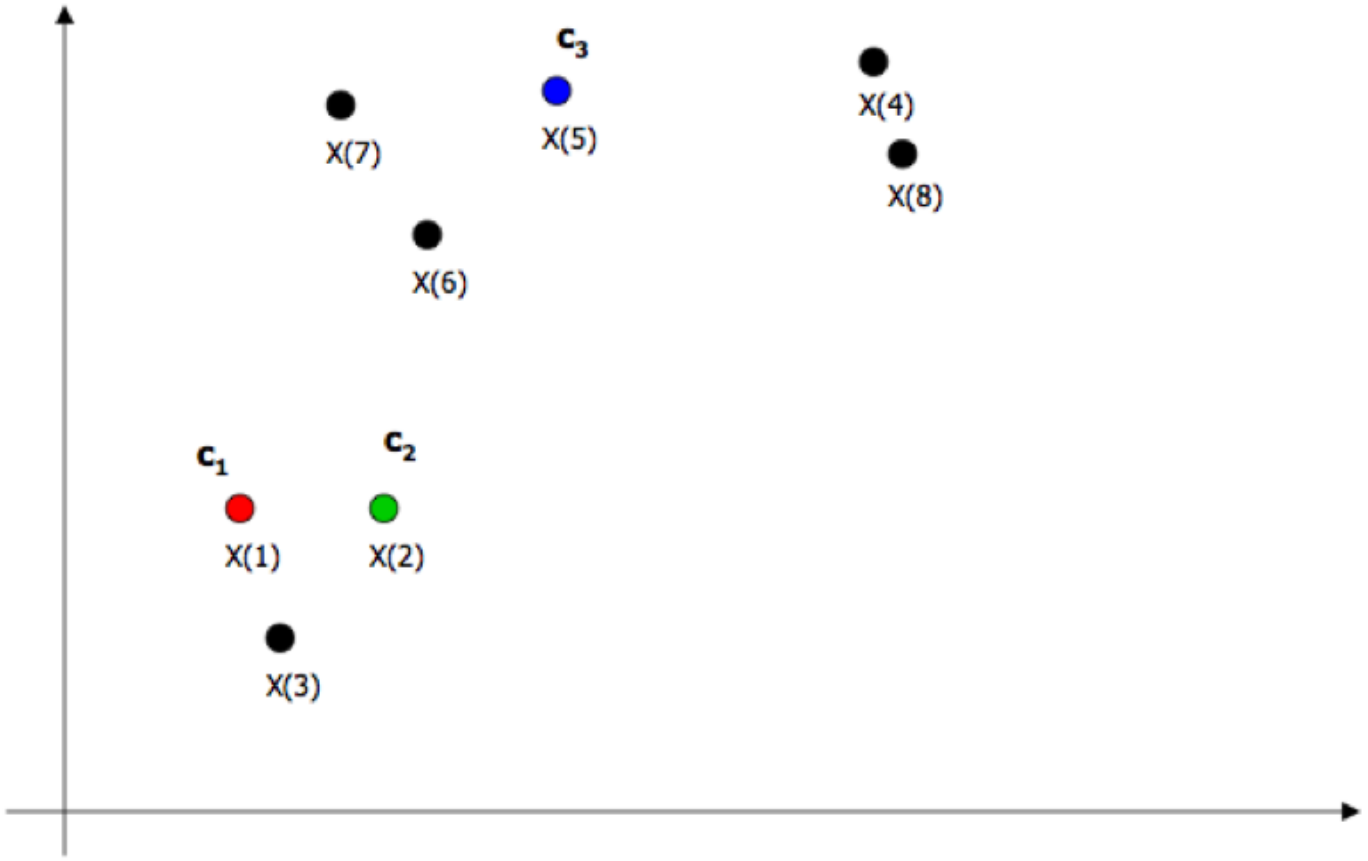
Output: Set of cluster representatives C, cluster membership vector $\mathbf{m}$

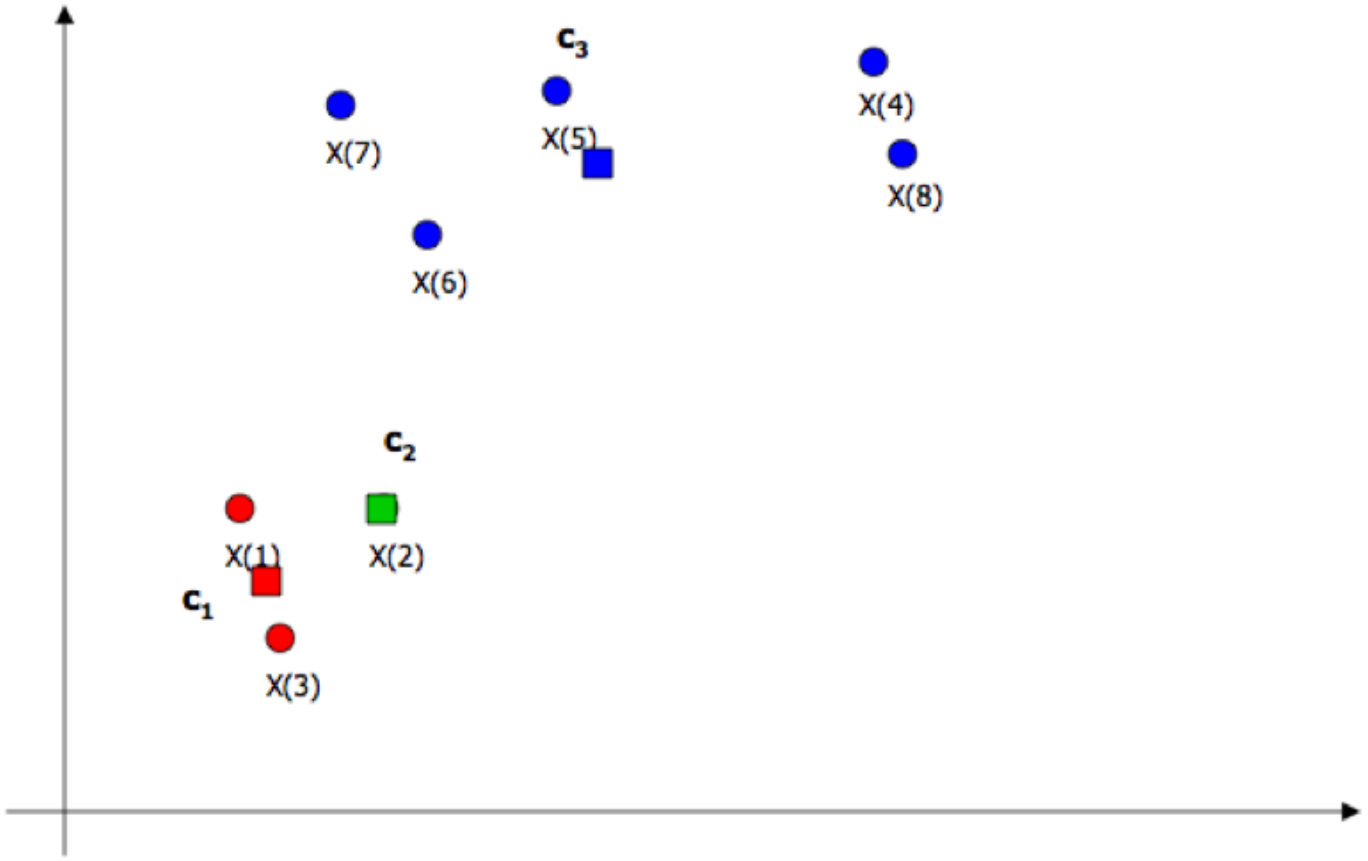
```
    /* Initialize cluster representatives C */
    Randomly choose k data points from D
    Use these k points as initial set of cluster representatives C
    repeat
        /* Data Assignment */
        Reassign points in D to closest cluster mean
        Update m such that m_i is cluster ID of ith point in D
        /* Relocation of means */
        Update C such that c_j is mean of points in jth cluster
    until convergence of objective function sum_{i=1}^{N} (argmin_j ||x_i - c_j||_2^2)
```

Score function: $$wc(C) = \sum_{k=1}^{K} wc(C_k) = \sum_{k=1}^{K} \sum_{x(i) \in C_k} d(x(i), r_k)^2$$

K-means example II


X(7)
X(5)
X(4)
X(8)
X(6)
X(1)
X(2)
X(3)


c3
X(7)
X(5)
X(4)
X(8)
X(6)
c1
c2
X(1)
X(2)
X(3)


c_3
X(7)
X(5)
X(4)
X(8)
X(6)
c_2
X(1)
X(2)
c_1
X(3)

Algorithm details

- **Does it terminate?**
  - *Yes! The objective function decreases on each iteration. It usually converges quickly.*
- **Does it converge to an optimal solution?**
  - *No! The algorithm terminates at a **local optima** which depends on the starting seeds.*
- **What is the time complexity?**
  - O(k·n·i),*where i is the number of iterations*

K-means

- **Strengths:**
  - Relatively efficient
  - Finds spherical clusters
- **Weaknesses**:
  - Terminates at local optimum (sensitive to initial seeds)
  - Applicable only when mean is defined
  - Need to specify k
  - Susceptible to outliers/noise

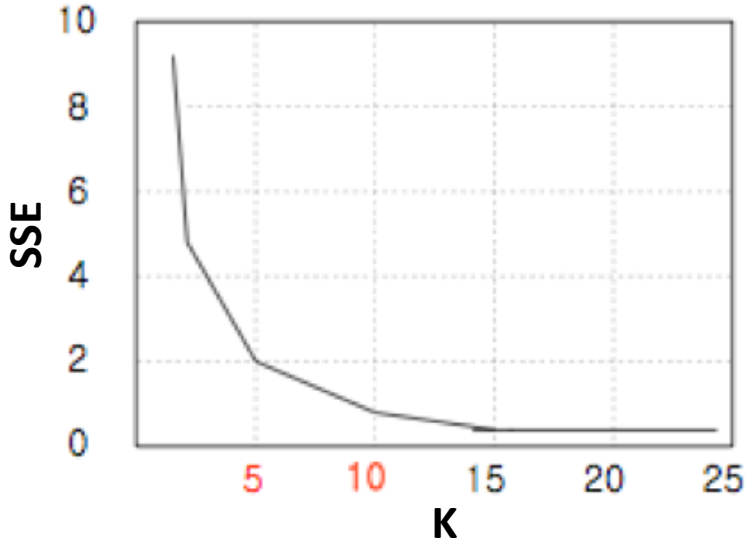
Variations

- **Selection of initial centroids**
  - *Run with multiple random selections, pick result with best score*
  - Use hierarchical clustering to identify likely clusters and pick seeds from distinct groups

- **Algorithm modifications:**
  - Recompute centroid after each point is assigned
  - Allow for merge and split of clusters (e.g., if cluster becomes empty, start a new one from randomly selected point)

Variations

- **How to select k?**
  - Plot objective function (within cluster SSE) as a function of k, look for *knee* in plot



K-means summary

- **Knowledge representation**
  - *K clusters are defined by canonical members (e.g., centroids)*
- **Model space the algorithm searches over?**
  - All possible partitions of the examples into k groups
- **Score function?**
  - *Minimize within-cluster Euclidean distance*
- **Search procedure?**
  - *Iterative refinement correspond to greedy hill-climbing*