Data mining & Machine Learning

CS 373
Purdue University

Dan Goldwasser
dgoldwas@purdue.edu

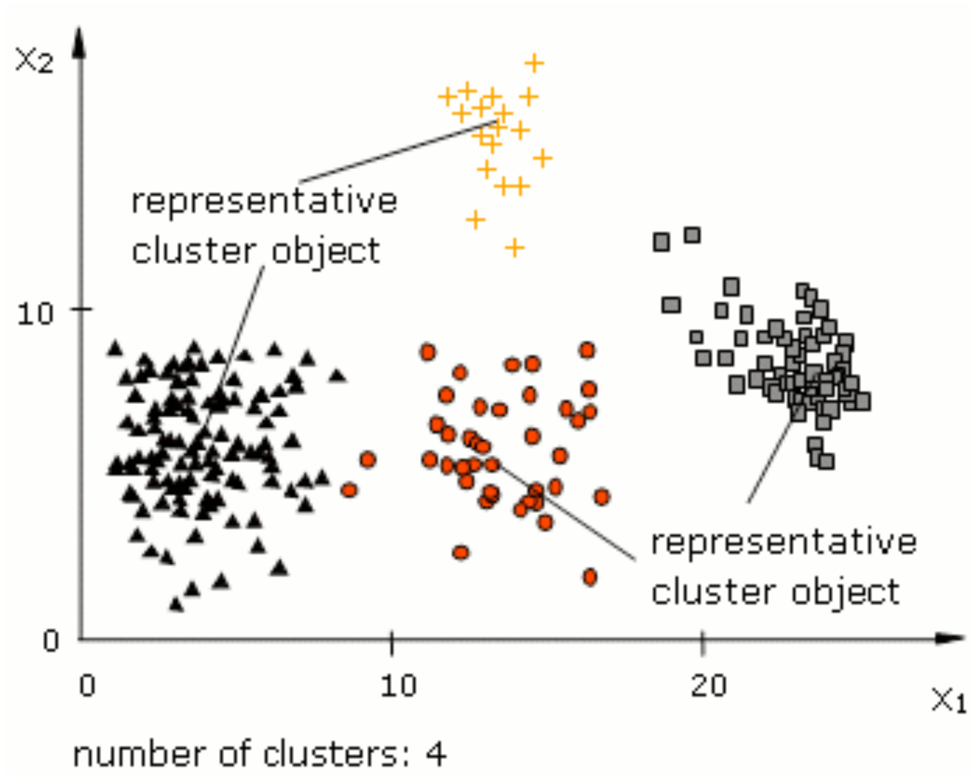
Today's Lecture

Clustering

- *Useful tool for finding structure in data,* ***but not perfect!***
  - *Success depends on initialization, choice of K, distance..*
  - *Also, what happens if a point belongs to more than one group?*

How can we represent complex structures and deal with cluster membership ambiguity?

Example: K-means



Groups represented by *canonical* item description(s)

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:

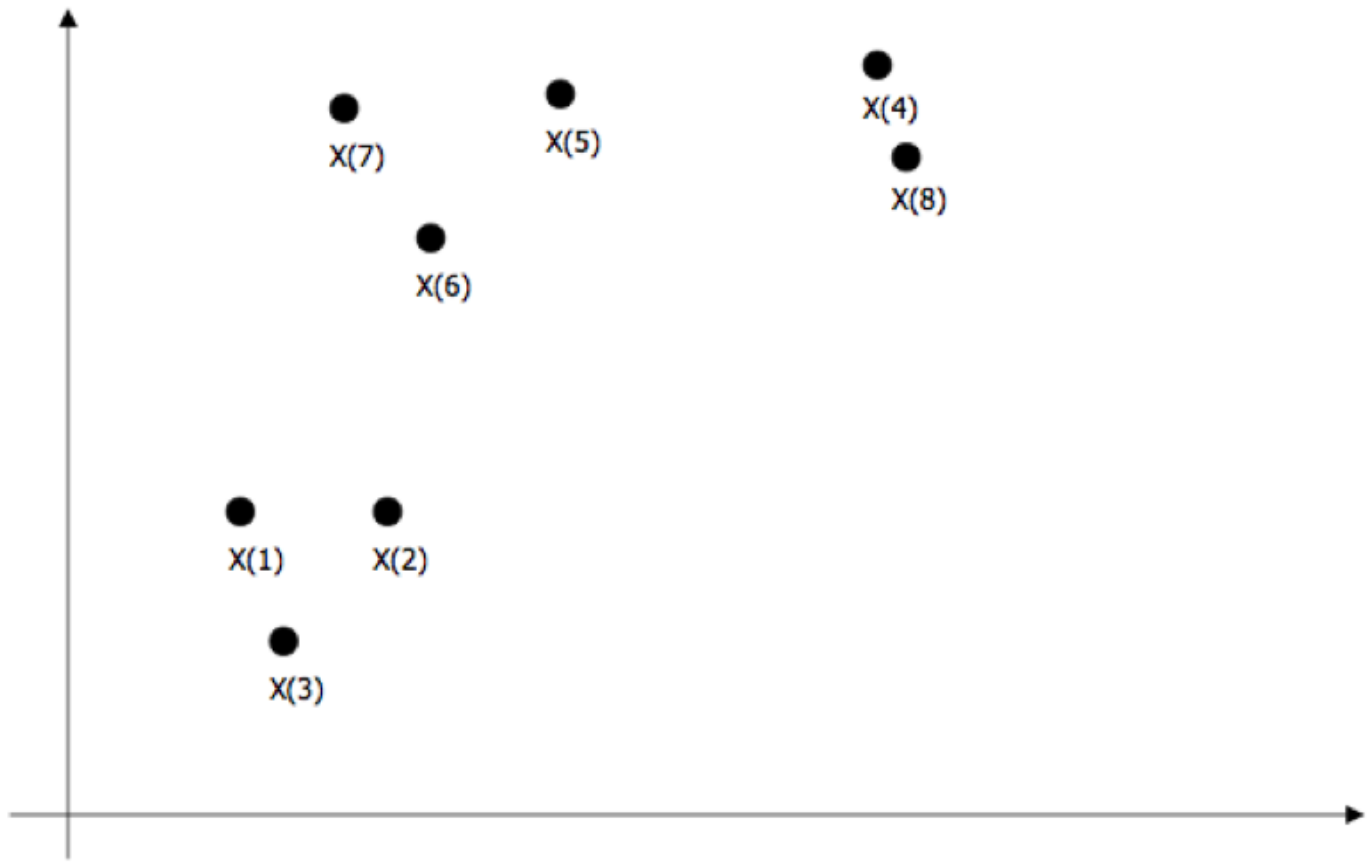
$$bc(C) = \sum_{1 \leq j < k \leq K} d(r_j, r_k)^2$$

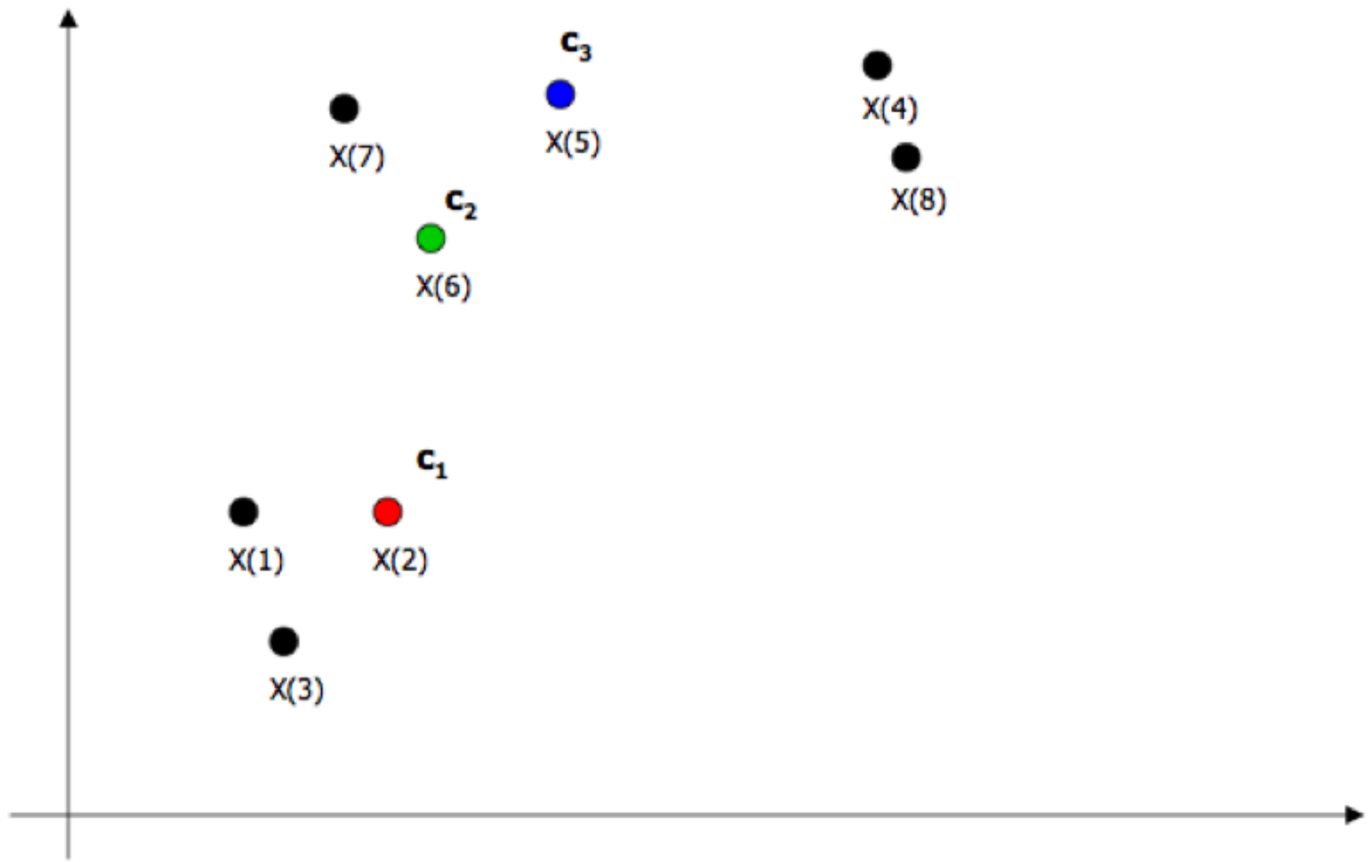
within-cluster distance:

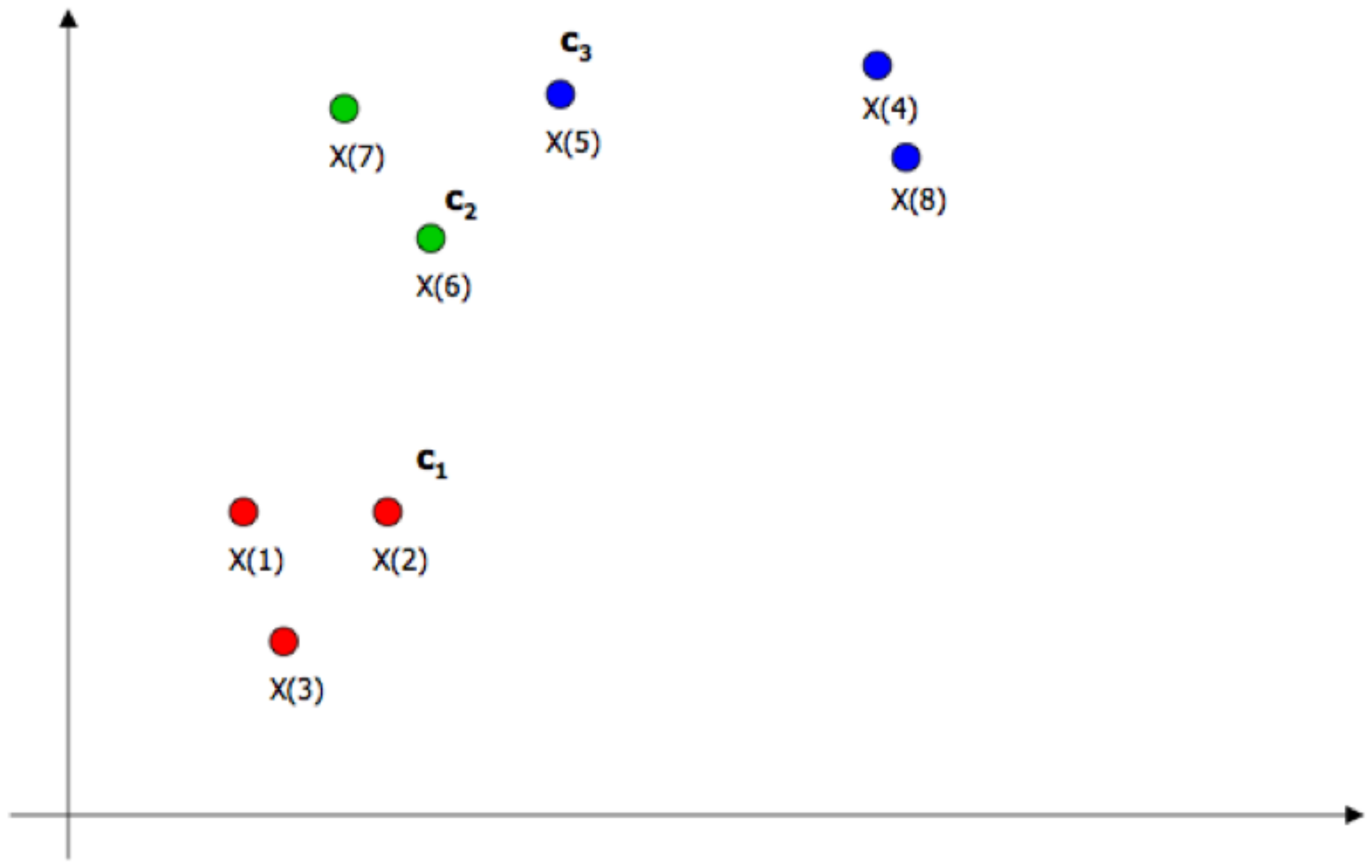
$$wc(C) = \sum_{k=1}^{K} wc(C_k) = \sum_{k=1}^{K} \sum_{x(i) \in C_k} d(x(i), r_k)^2$$

Example: K-means

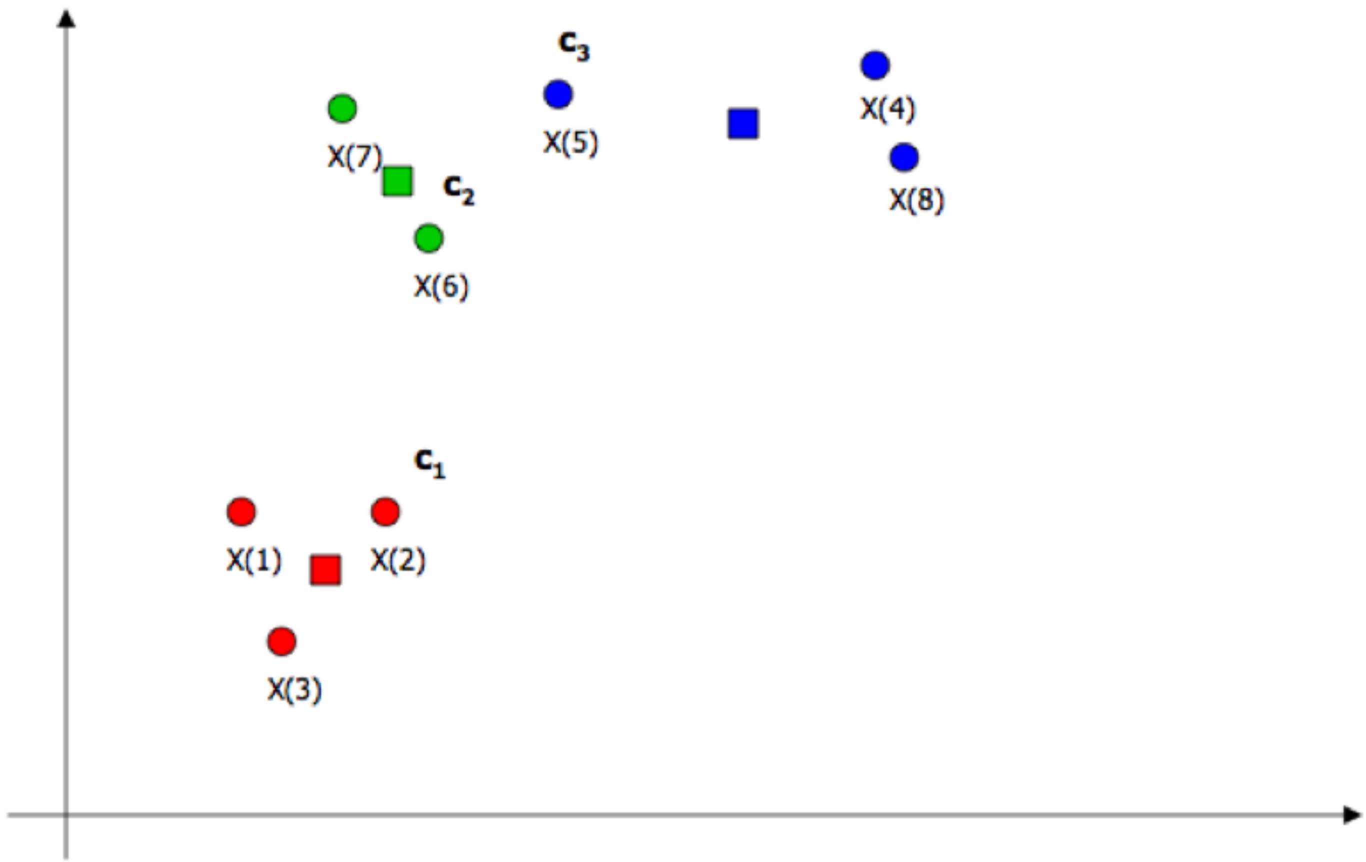
- **Algorithm idea:**
  - Start with k randomly chosen **centroids**
    - **Centroids** characterize the cluster
  - Repeat until no changes in assignments
    - Assign instances to closest centroid
    - Recompute cluster centroids


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

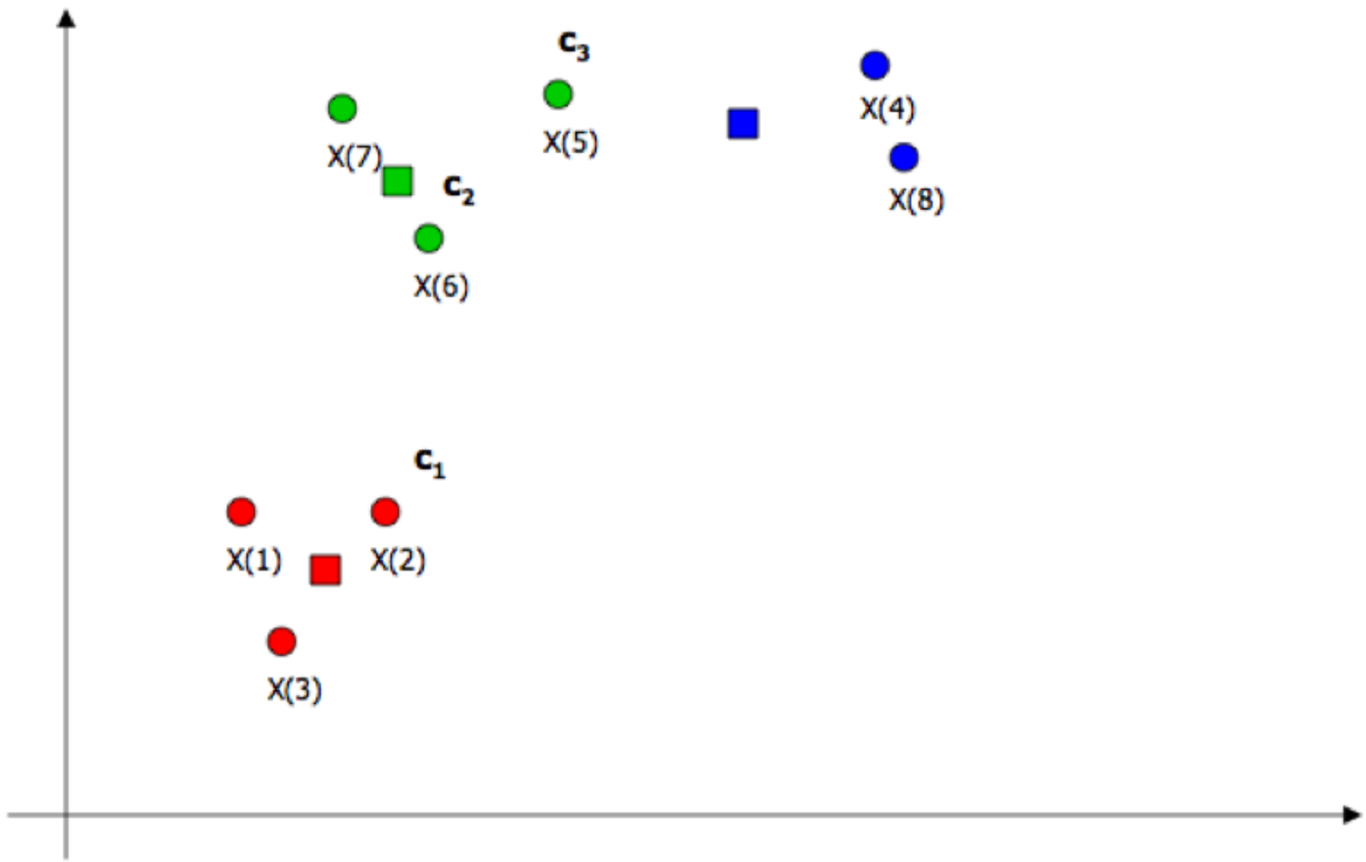

c3
X(4)
X(7)
X(5)
X(8)
c2
X(6)
c1
X(1)
X(2)
X(3)



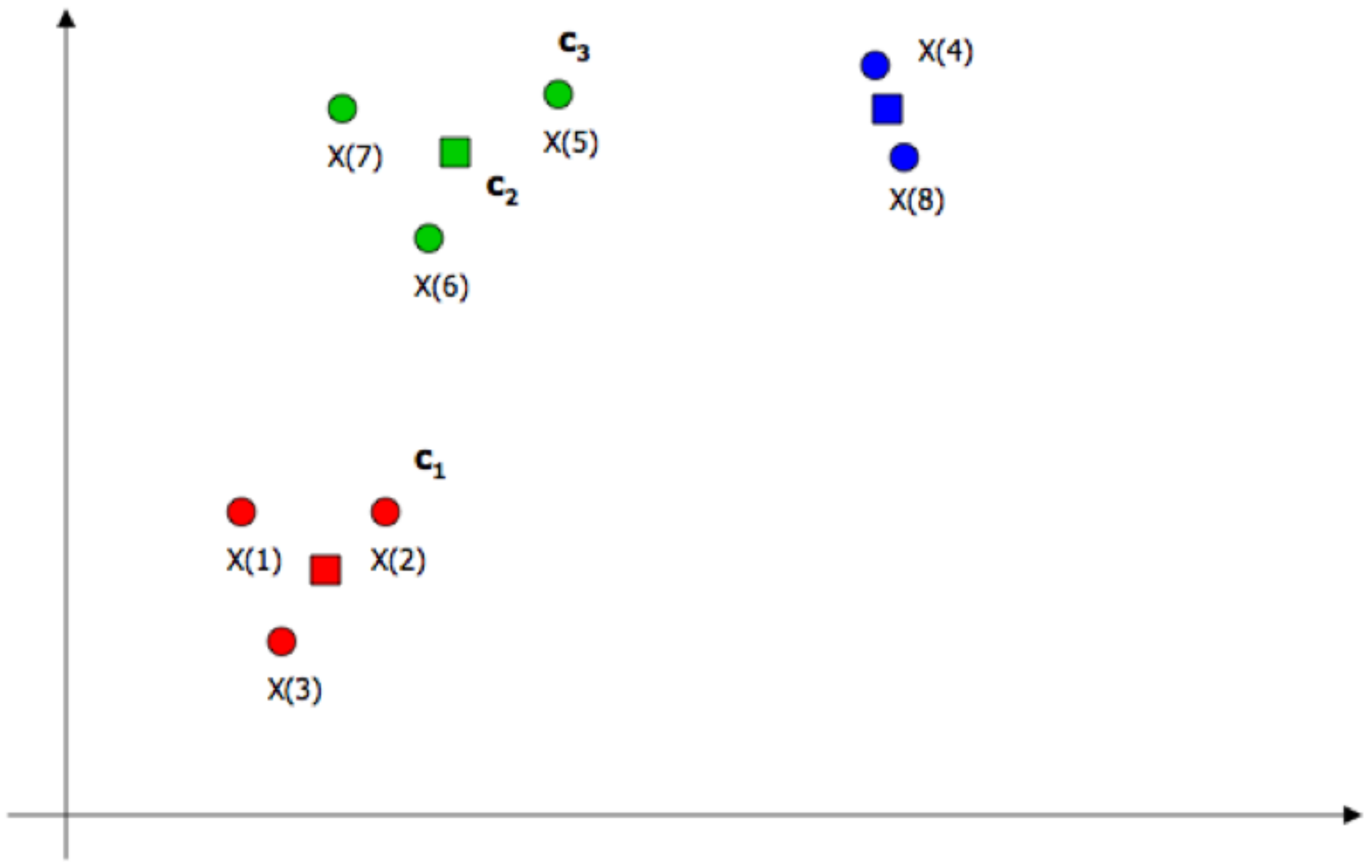
c3
X(4)
X(7)
X(5)
X(8)
c2
X(6)
c1
X(1)
X(2)
X(3)



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



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


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

K-Means as Unsupervised Learning

- Most learning algorithms involve iterative search over assignments due to score functions which require combinatorial optimization
  - Not feasible to exhaustively search for optimal solution
- Let’s think about it as a learning algorithm.
  - What is the search space?
  - What is the scoring function? Should we **minimize** or **maximize**?
  - How does the search procedure work?
-

K-Means as Unsupervised Learning

- Learning algorithms can be defined using an objective *(=scoring*) function.
- The objective function allows us to compare two models.
- K-means: **`min`** Sum of Squared Errors
  - Also known as just squared L_2 distance (..also known as Euclidean distance)
  - Local optimum

$$\sum_{i=1}^{N}(argmin_j||\mathbf{x_i} - \mathbf{c_j}||_2^2)$$

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 $\mathbf{m}$ such that m_i is cluster ID of ith point in D

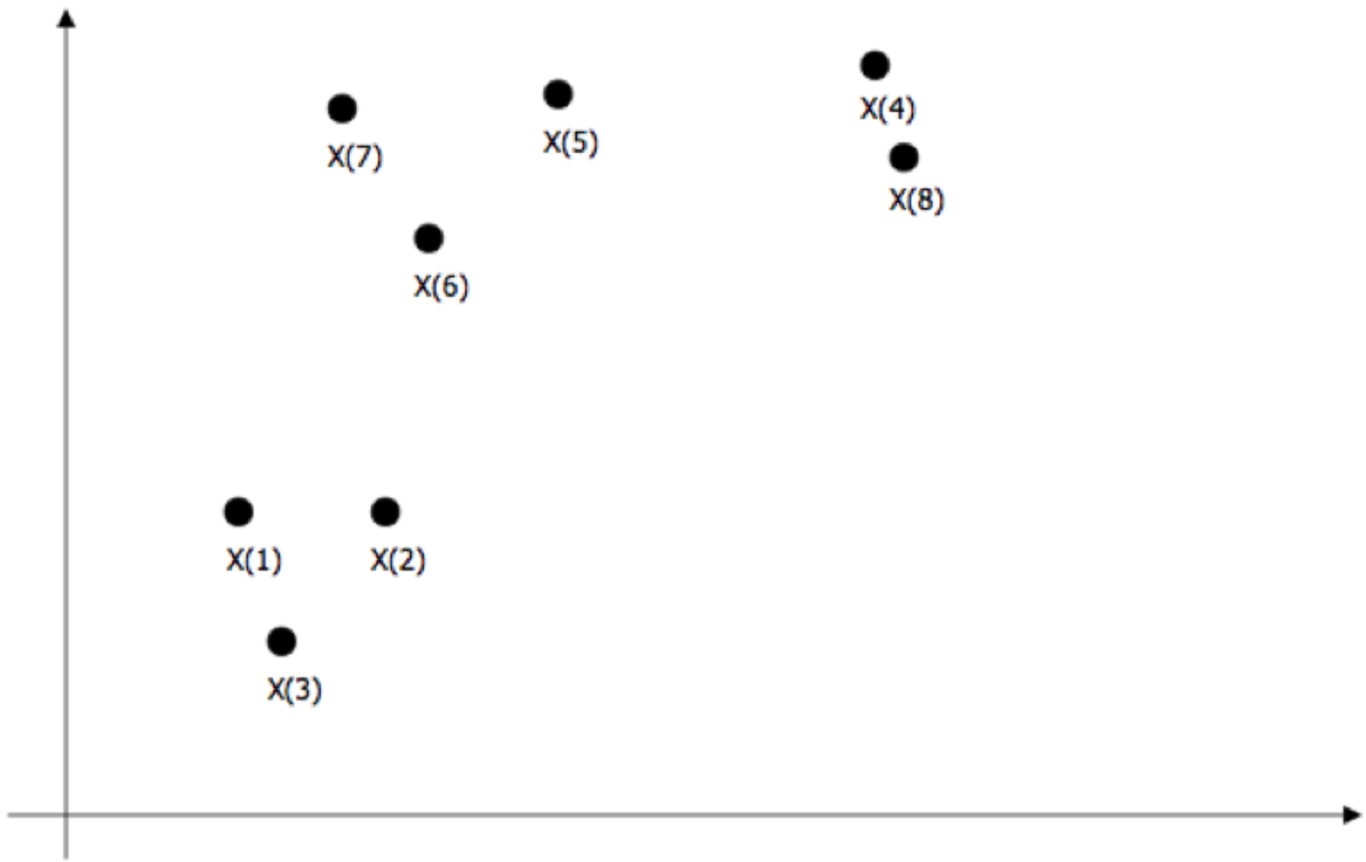
/* Relocation of means */

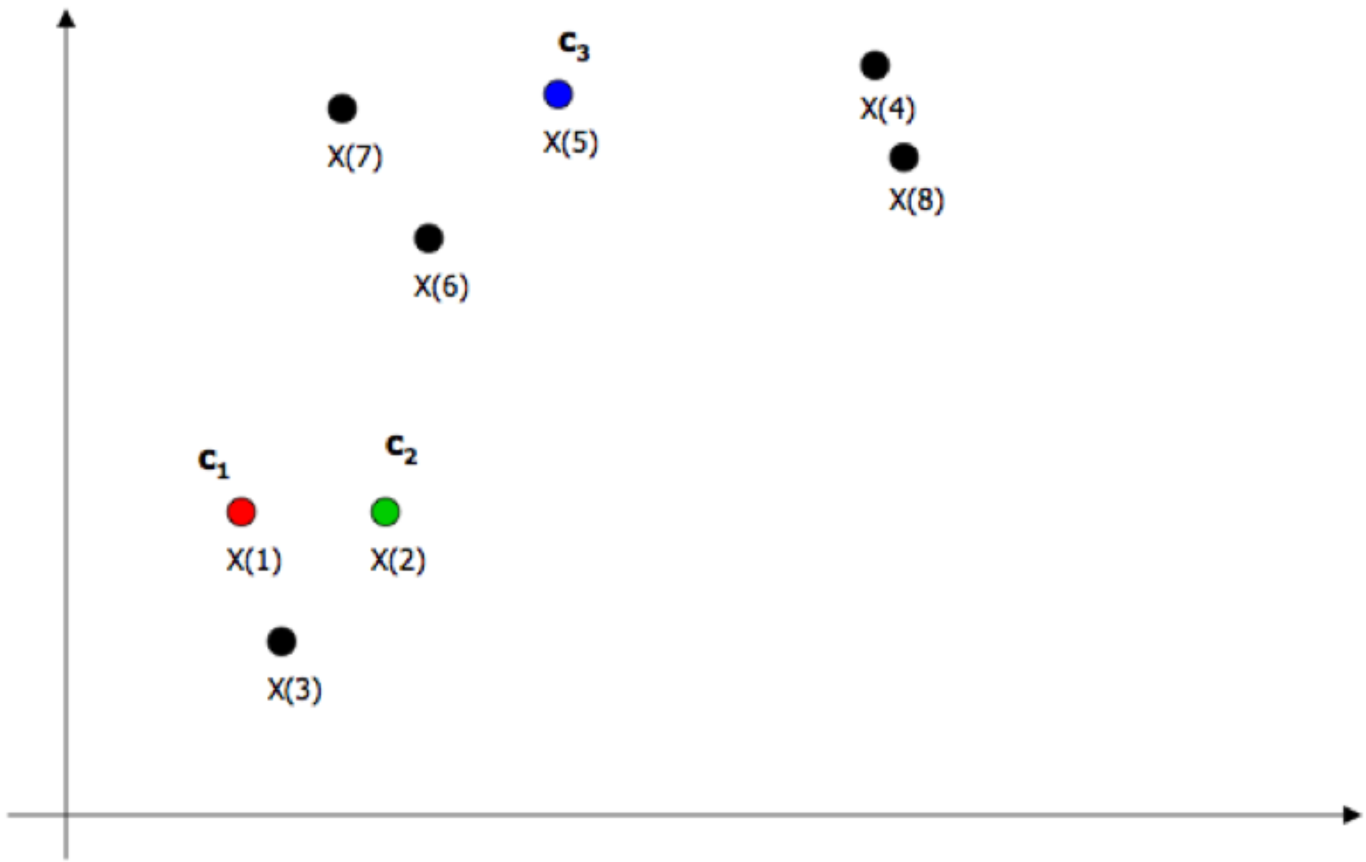
Update C such that c_j is mean of points in jth cluster

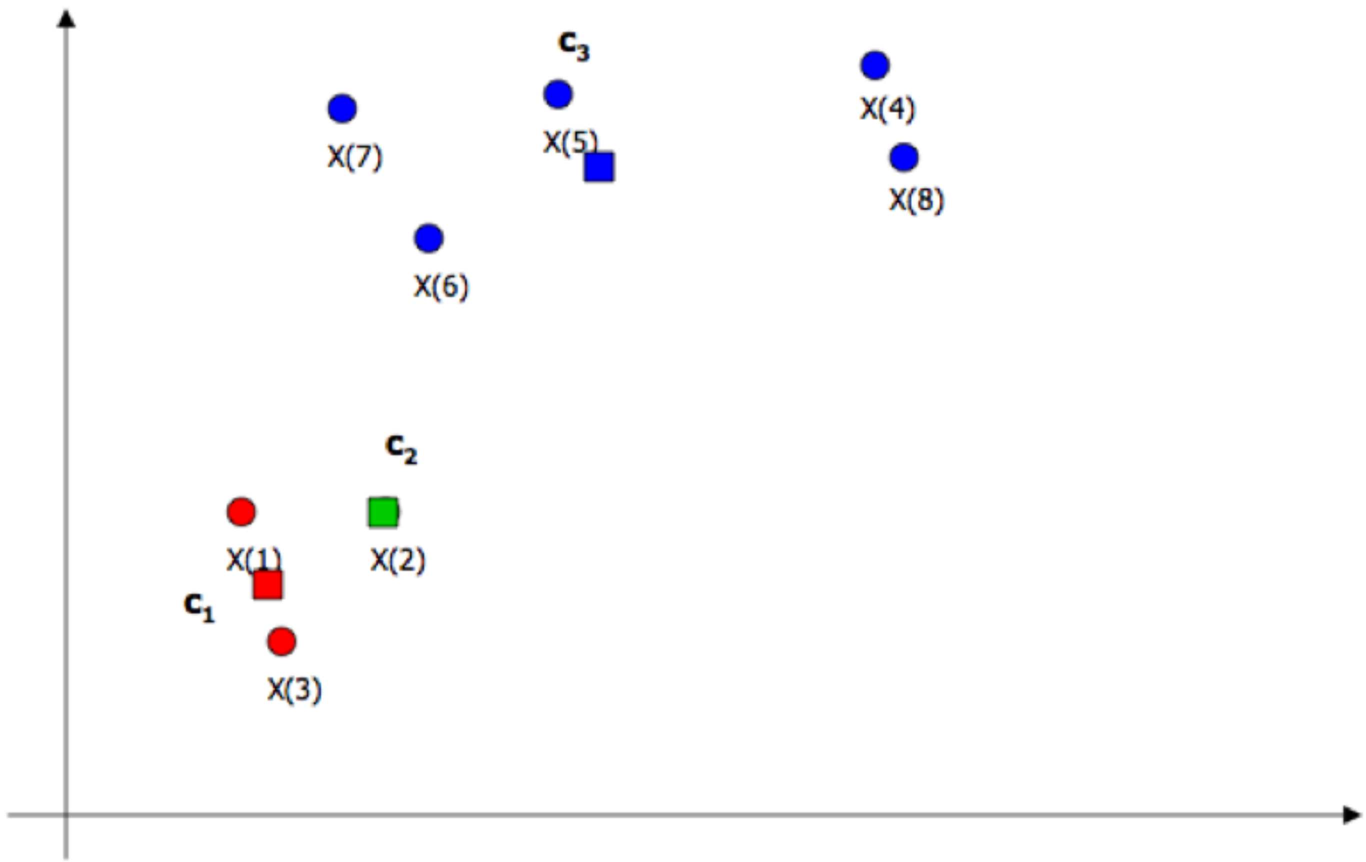
until convergence of objective function $\sum_{i=1}^{N}(argmin_j ||\mathbf{x_i} - \mathbf{c_j}||_2^2)$

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
  - Easy to understand and implement

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

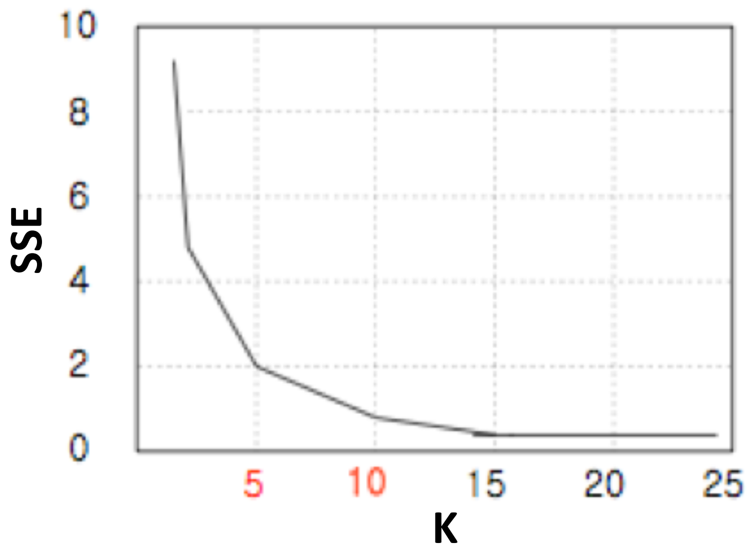
Variations

- **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*

Back to finding structure in data

Question (optional take home quiz)

Consider the co-reference resolution problem:

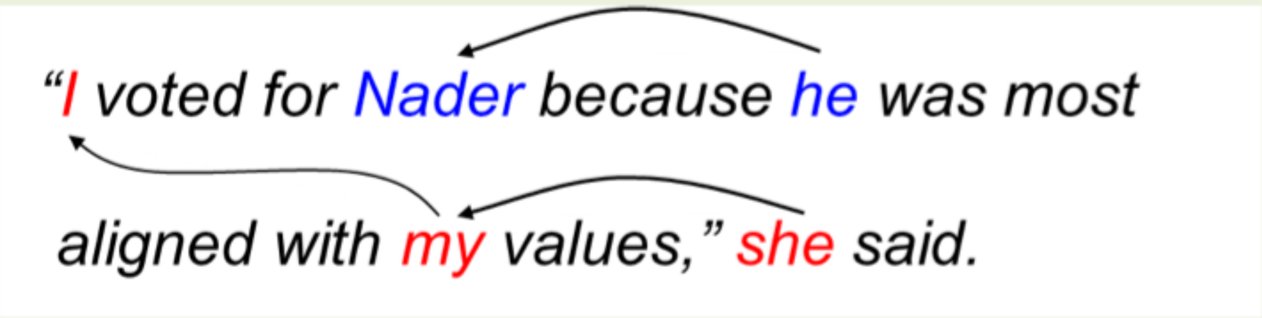


Image: Stanford NLP group

Can you frame it as a clustering problem?

If **no** – why not.

If **yes** – define the *desired* clusters and suggest an appropriate distance metric

Hierarchical clustering

Hierarchical methods

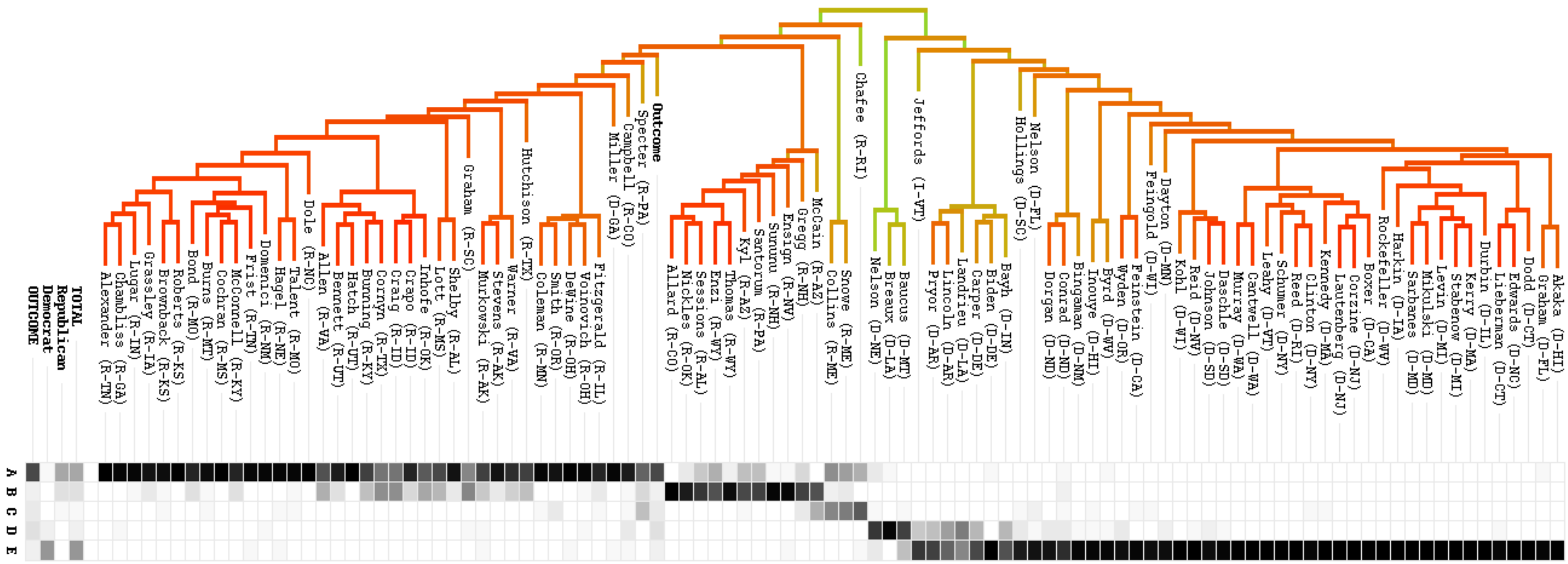
- Construct a *hierarchy* of nested clusters rather than picking k beforehand
- **Approaches**:
  - Agglomerative: merge clusters successively
  - Divisive: divided clusters successively
- Dendrogram (tree diagram) depicts sequences of merges or splits and height indicates distance

Agglomerative

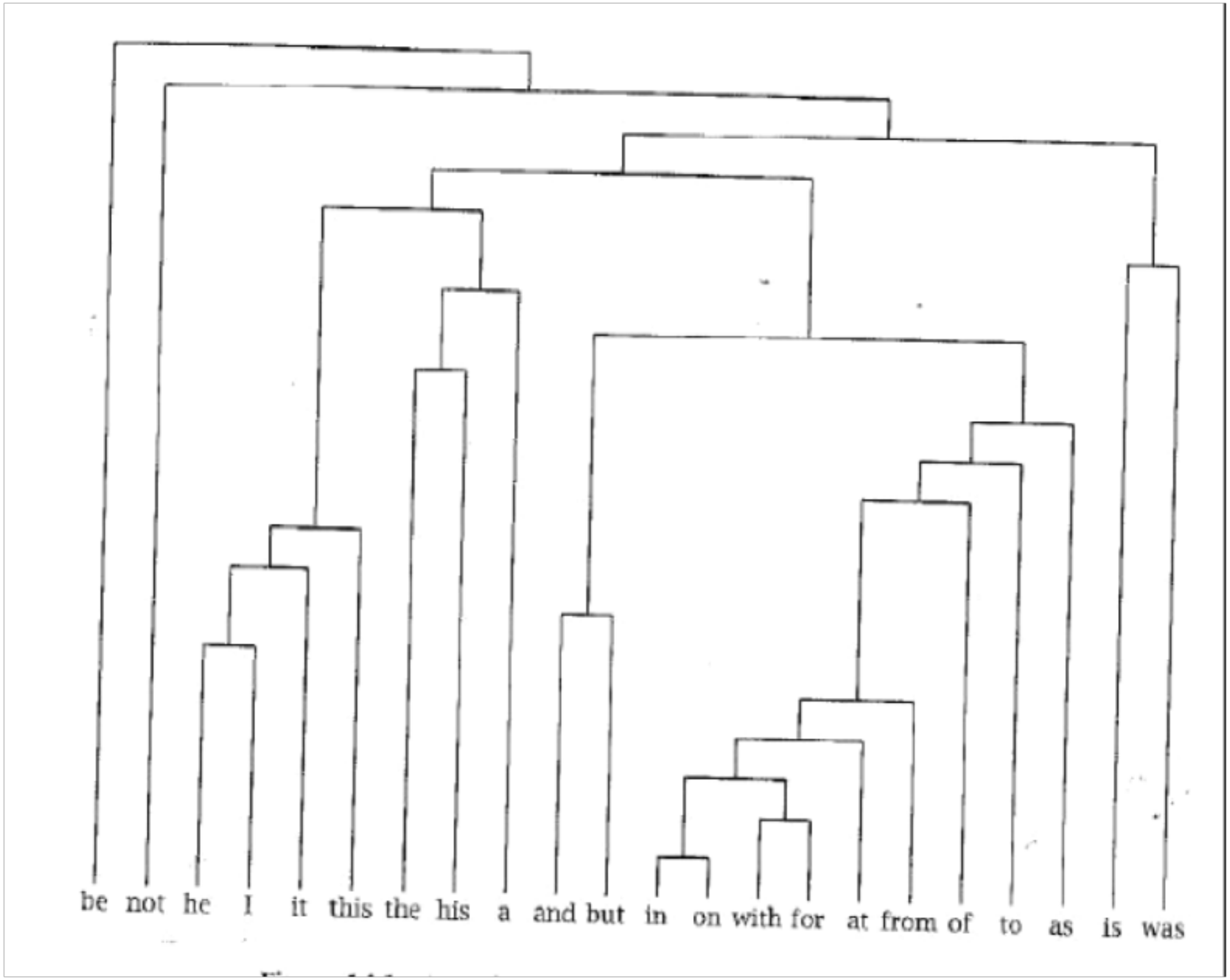
- For i = 1 to n:
  - Let $C_i = \{x(i)\}$
- While $|C|>1$:
  - Let C_i and C_j be the pair of clusters with min $D(C_i,C_j)$
  - $C_i=C_i \cup C_j$
  - Remove C_j

Distance measures between clusters

- **Single-link/nearest neighbor:**
  - $D(C_i,C_j) = \mathbf{min}\{\, d(x,y) \mid x \in C_i,\ y \in C_j \,\}$
    $\Rightarrow$ *can produce long thin clusters*
- **Complete-link/furthest neighbor:**
  - $D(C_i,C_j) = \mathbf{max}\{\, d(x,y) \mid x \in C_i,\ y \in C_j \,\}$
    $\Rightarrow$ *is sensitive to outliers*
- **Average link:**
  - $D(C_i,C_j) = \mathbf{avg}\{\, d(x,y) \mid x \in C_i,\ y \in C_j \,\}$
    $\Rightarrow$ *compromise between the two*



Clustering represented with dendogram



A hierarchical clustering of 22 frequent English words represented as a dendrogram.

Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University Press. 2008.

lawyer	1000001101000
newspaperman	100000110100100
stewardess	100000110100101
toxicologist	10000011010011
slang	1000001101010
babysitter	100000110101100
conspirator	1000001101011010
womanizer	1000001101011011
mailman	10000011010111
salesman	100000110110000
bookkeeper	1000001101100010
troubleshooter	10000011011000110
bouncer	10000011011000111
technician	1000001101100100
janitor	1000001101100101
saleswoman	1000001101100110
...	
Nike	1011011100100101011100
Maytag	10110111001001010111010
Generali	10110111001001010111011
Gap	1011011100100101011110
Harley-Davidson	10110111001001010111110
Enfield	101101110010010101111110
genus	101101110010010101111111
Microsoft	10110111001001011000
Ventritex	101101110010010110010
Tractebel	1011011100100101100110
Synopsys	1011011100100101100111
WordPerfect	1011011100100101101000
....	
John	101110010000000000
Consuelo	101110010000000001
Jeffrey	101110010000000010
Kenneth	10111001000000001100
Phillip	101110010000000011010
WILLIAM	101110010000000011011
Timothy	10111001000000001110
Terrence	101110010000000011110
Jerald	101110010000000011111
Harold	101110010000000100
Frederic	101110010000000101
Wendell	10111001000000011

Brown Clusters:

- Hierarchical clustering of words based on their context

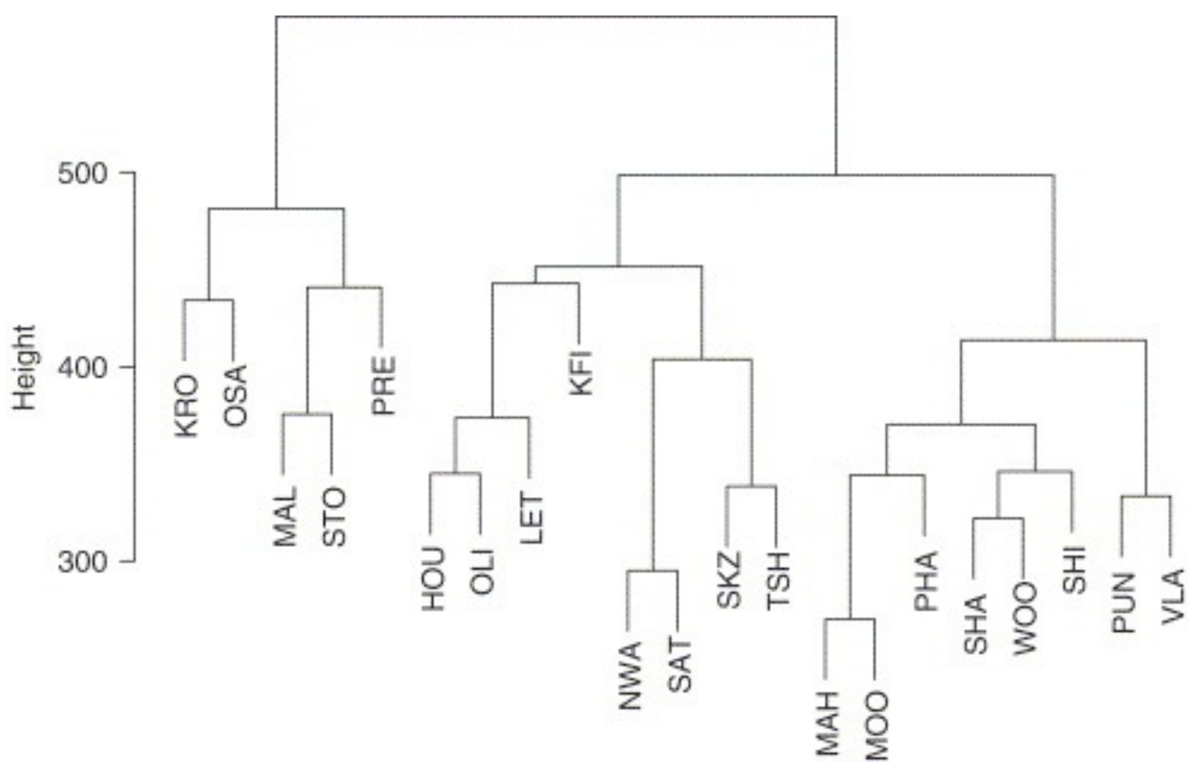
- Clusters correspond to topically-related words

- Bit-String encoding: capture tree path from the root to the word

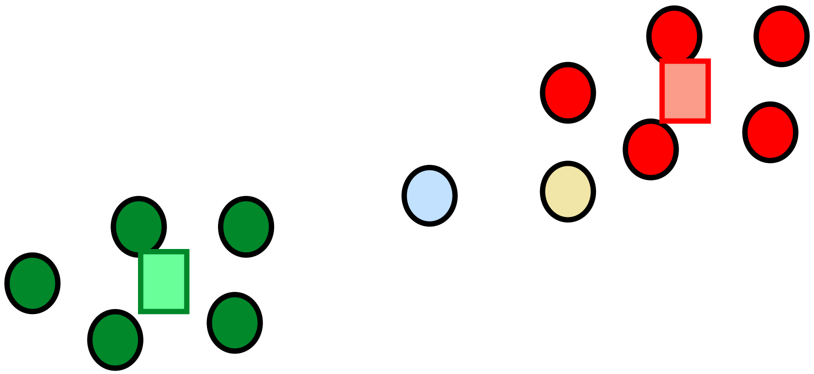
- What are the **properties** of this representation?
- How can we use it?

Example

- Agglomerative clustering results of surface water availability in areas of Kruger National Park, South Africa. Three primary clusters can be distinguished, which correspond to a north, south, and far south spatial division of the KNP.



Some thought on K-Means



- What should be the cluster assignment for blue and yellow circles?
- What can we say about the certainty of this assignment?
- What would it matter?

How can we put these observations into an algorithm?
Stay tuned.. Coming up!