Data mining & Machine Learning

CS 373
Purdue University

Dan Goldwasser
dgoldwas@purdue.edu

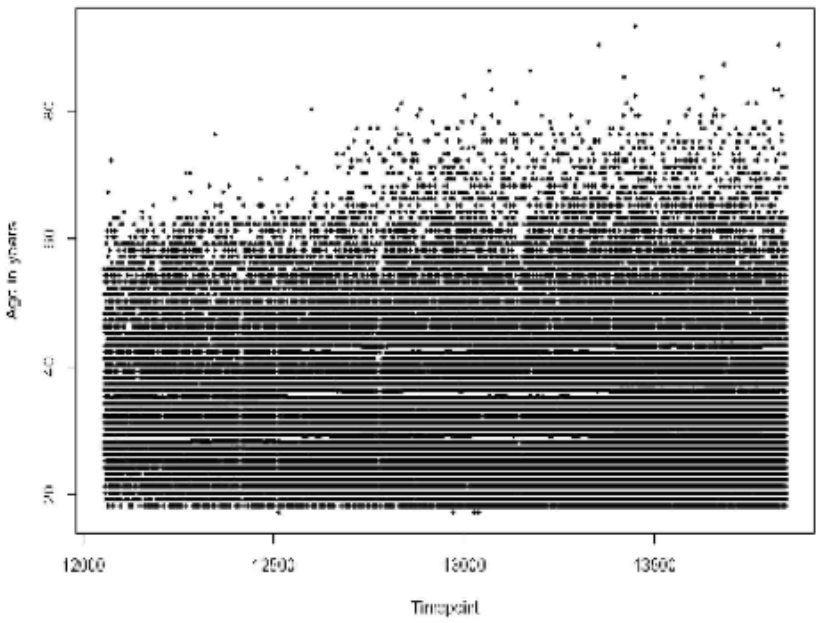
Today's Lecture

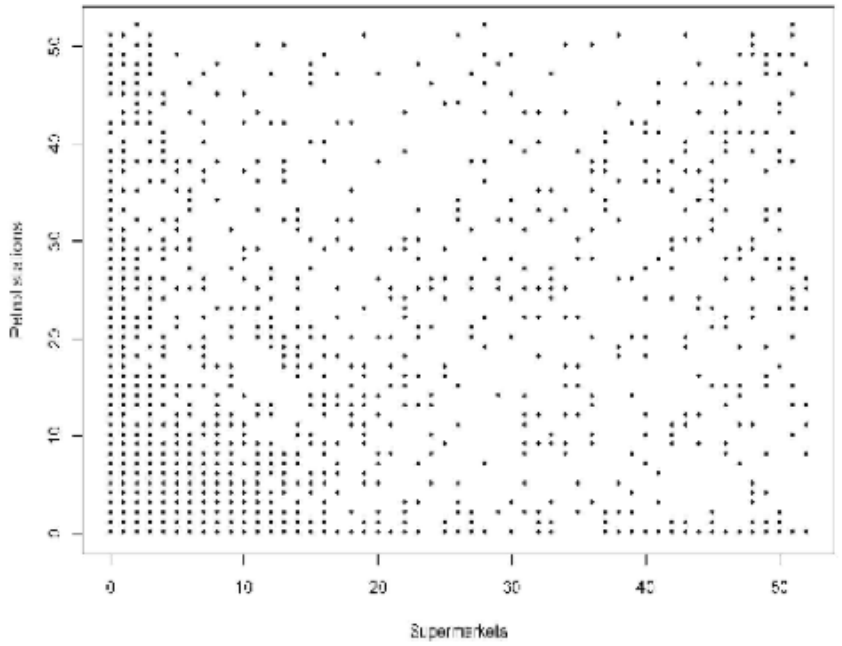
It's Complicated!

- *The data we get is often too complicated simply "eyeball"*
- It's too complex to visualize in a meaningful way
- *We don't understand it enough to start annotating*

How can make sense of complex data?

It’s complicated!





Unsupervised methods

- **In other cases we would like to use algorithms to make sense of the data.**
  - Can we simplify the data, such that preserve as much of the its properties?
  - Can we identify patterns in the data? Can we simplify the data based on these patterns?
    - Automatically identify that there are 3 groups in the data, and analyze them separately.
  - Can we learn meaningful representations, such that these patterns can be identified?
  - Can we identify outliers? Detect anomalous behavior?
- **In all these cases, we want to exploit the properties of the data, rather than relying on an external judgment (annotation)**

Dimensionality Reduction using PCA

Dimensionality reduction

- Identify and describe the “dimensions” that underlie the data
  - May be more fundamental than those directly measured but hidden to the user
- Reduce dimensionality of modeling problem
  - Benefit is simplification, it reduces the number of variables you have to deal with in modeling
- Can identify set of variables with similar behavior

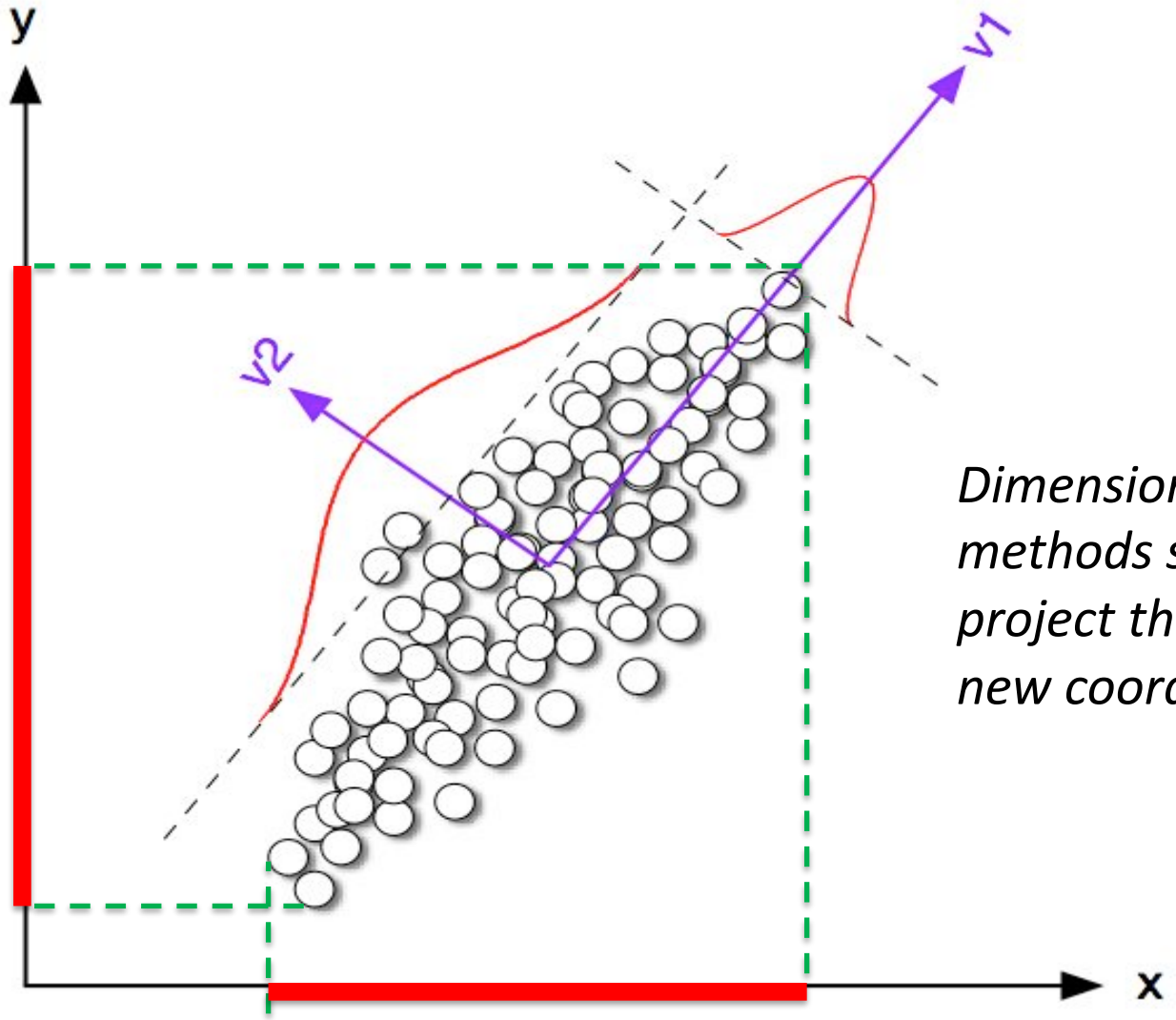
Dimensionality reduction methods

- ***Feature Selection***
  - Select a subset of the features (manual process!)

VS.

- **Principal component analysis (PCA)**
  - Linear transformation, minimize unexplained variance

Dimensionality Reduction Intuition

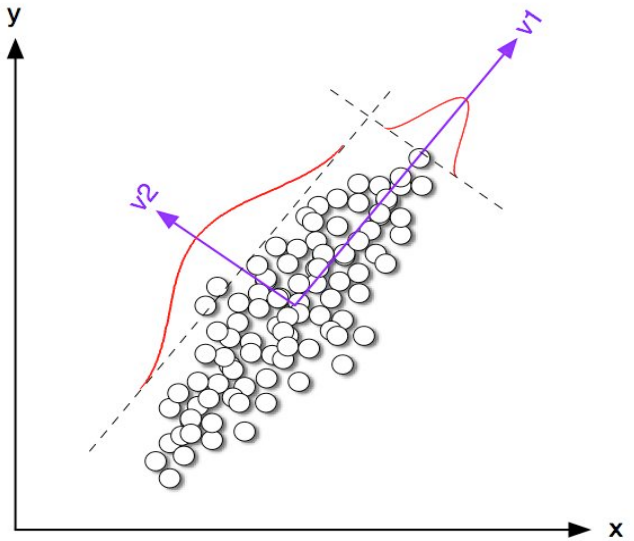


Dimensionality Reduction methods such as PCA project the data into a new coordinate system.

*Projecting the data points to the x or y axis is **feature selection***

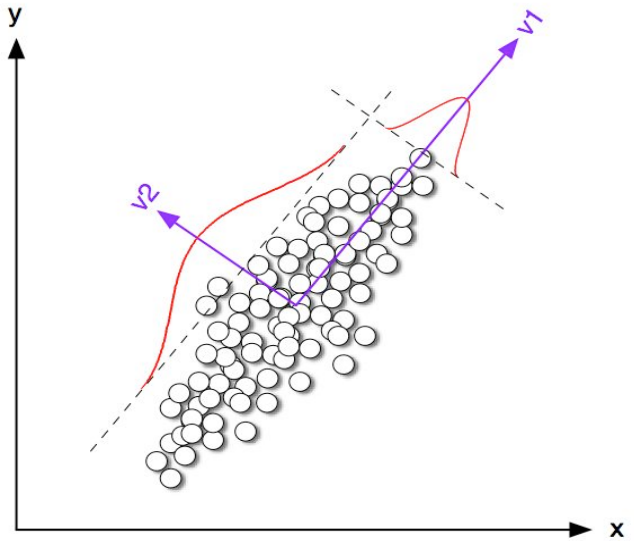
PCA Intuition

- High dimensional data can record many observations that are **highly correlated**
  - *E.g., Income and education, temperature and precipitation*
  - These variables capture the "same information"
- **PCA**: *a transformation to new variables, that are linearly uncorrelated*
- Each variable:
  - Captures as much of the variability in the data
  - Orthogonal to the other variables



PCA Intuition

- If you maintain all the principle components, this is a linear transformation of the data, so you can reconstruct the original dataset.
- Choosing a subset of principle components is similar to lossy compression, reducing the dimensionality of the data while maintain as much of its variance.



Principal component analysis (PCA)

- Given data matrix **D** with **p** dimensions:
  - Preprocess **D** so that the mean of each attribute is 0, call this matrix **X**
  - Compute pxp covariance matrix: $\Sigma = X^T X$
  - Compute eigenvectors/eigenvalues of covariance matrix:

$$\mathbf{A}\Sigma\mathbf{A}^{-1} = \Lambda$$

$$(\Sigma - \lambda\mathbf{I})\mathbf{a} = 0$$

$\mathbf{A}$: matrix of eigenvectors

Λ : diagonal matrix of eigenvalues

$\mathbf{a}$: 1st principal component, eigenvector assoc. with largest eigenvalue (λ)

  - Eigenvectors $\mathbf{A}$ are the **principal component** vectors, where each $\mathbf{a}$ is a px1 column vector of projection weights

Framework for learning methods

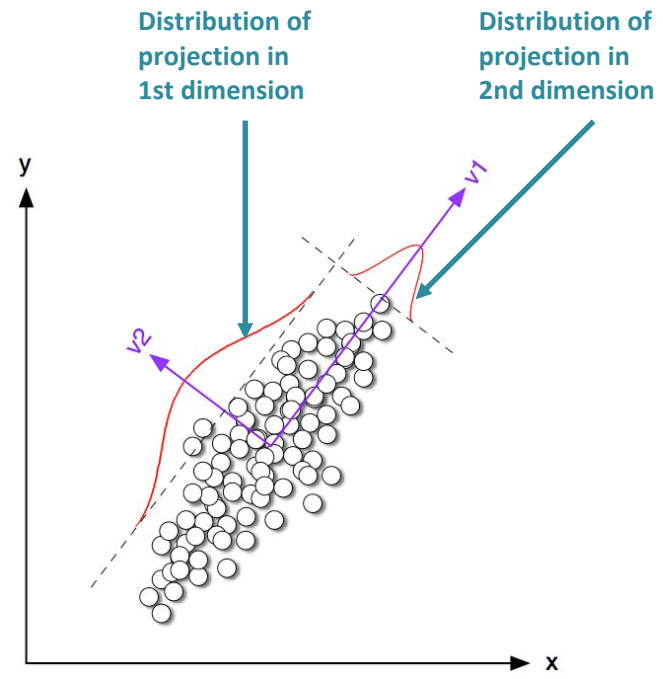
- **Model space**
  - Choice of knowledge representation defines a set of possible models or patterns
- **Scoring function**
  - Associates a numerical value (score) with each member of the set of models/patterns
- **Search technique**
  - Defines a method for generating members of the set of models/patterns and determining their score

What is the *model space* for PCA?

$$\mathbf{A}\Sigma\mathbf{A}^{-1} = \Lambda$$

$\mathbf{A}$: matrix of eigenvectors
Λ : diagonal matrix of eigenvalues

- **A** is a *p x p* matrix of principal components (if data is p-dimensional) each column is a basis vector, each cell is a projection weight
- **Model space**: is defined by **A**
  - Method needs to choose the p^2 weights that populate **A**, i.e., set of p basis vectors
  - **Constraints**: Basis vectors must be **orthonormal**, i.e., each has a norm of 1 and any pair of basis vectors have dot-product of 0
  - E.g., any orthogonal set of v_1 and v_2



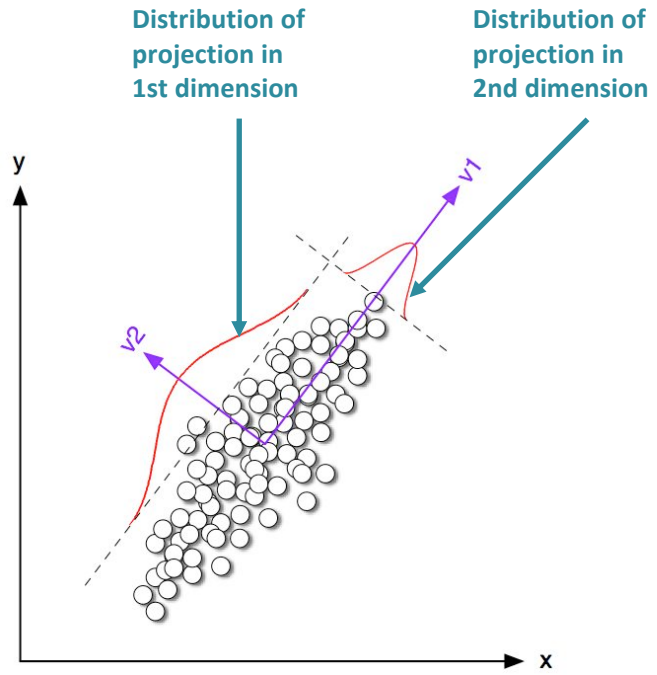
What is the score function for PCA?

$$\mathbf{A}\Sigma\mathbf{A}^{-1} = \Lambda$$

$\mathbf{A}$: matrix of eigenvectors
Λ : diagonal matrix of eigenvalues

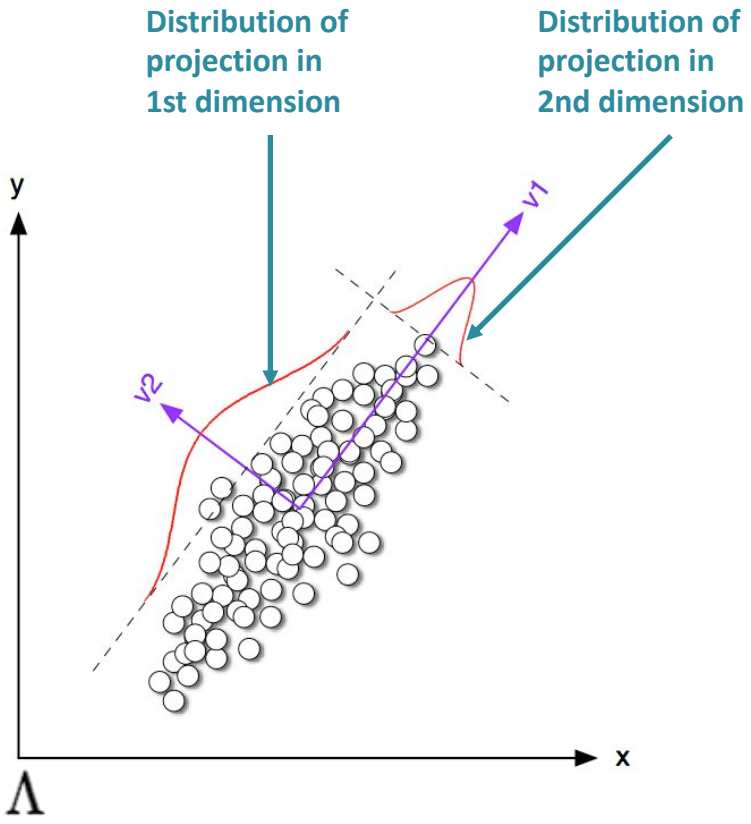
- **Λ** is diagonal matrix of *p* eigenvalues
  - Each eigenvalue λ_i corresponds to the variance of dimension **i**
- **Score function**: sum of eigenvalues in **Λ** $\sum_{j=1}^{p} \lambda_j$
- Sum of eigenvalues is equal to the sum of the variances of the original attributes:

$$\sum_{j=1}^{p} \sigma_j^2 = \sum_{j=1}^{p} \lambda_j$$



What is the search method for PCA?

- **Goal:** find basis vectors that maximize variance
  - 1st basis (eg. v_1) ***maximizes*** variance of projected data
  - 2nd basis (eg. v_2) again ***maximizes*** variance of projected data, but has to be orthogonal to previous bases, …
  - New dimensions are orthogonal, thus transformed features have 0 covariance
- **Search**: Solving eigensystem corresponds to finding the $\mathbf{A}\Sigma\mathbf{A}^{-1} = \Lambda$ orthonormal basis that **maximize** variance of projected data



Applying PCA

- Choose number of target dimensions (i.e., select m < p)
  - Transform data vectors by projecting them onto the first *m* principal components, which correspond to top m eigenvectors)

$$\mathbf{x} = [x_1, x_2, \ldots, x_p] \text{ (original instance)}$$
$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_p] \text{ (principal components)}$$

$$x'_1 = \mathbf{a}_1\mathbf{x} = \sum_{j=1}^{p} a_{1j}x_j$$

Linear transformation from the original coordinates to the first PC

$$\ldots$$

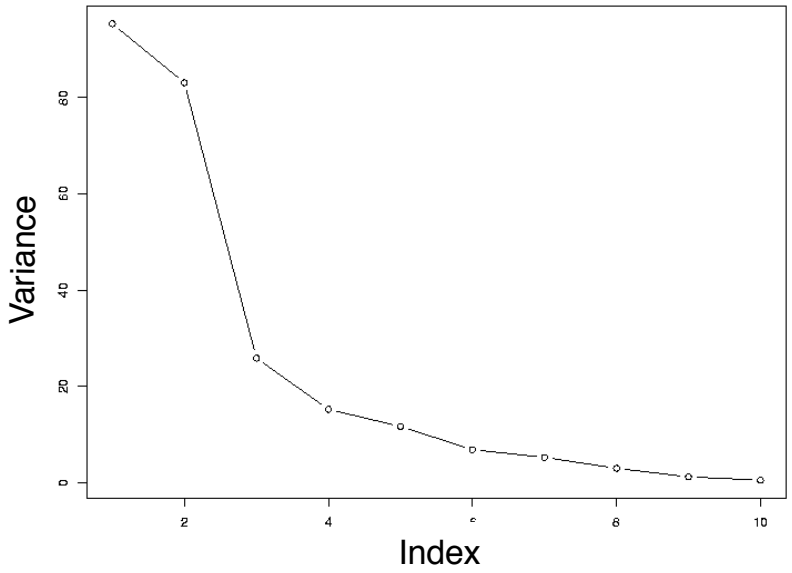
$$x'_m = \mathbf{a}_m\mathbf{x} = \sum_{j=1}^{p} a_{mj}x_j \quad for\ m < p$$

If ***m=p*** then data is transformed
If ***m<p*** then transformation is lossy and dimensionality is reduced

$$\mathbf{x}' = [x'_1, x'_2, \ldots, x'_m] \text{ (transformed instance)}$$

Applying PCA (cont')

- **Goal**: *Find a new (smaller) set of dimensions that captures most of the variability of the data*
- Can use **scree plot** to choose number of dimensions
  - Choose $m < p$ so projected data captures much of the variance of original data

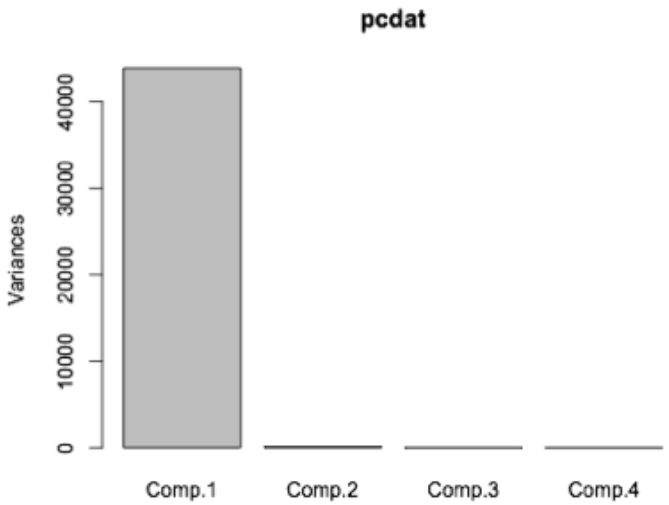


PCA example on Iris data

Choose m=1

```
> x <- scale(as.matrix(d[,1:4]),scale=FALSE)
> sigma <- t(x)%*% x
> sigma
           V1       V2       V3        V4
V1 102.16833  -5.8510 189.7787  77.01867
V2  -5.85100  28.0126 -47.9352 -17.57920
V3 189.77867 -47.9352 463.8637 193.16173
V4  77.01867 -17.5792 193.1617  86.77973

> pcdat <- princomp(d[,1:4])
> summary(pcdat)
Importance of components:
                          Comp.1     Comp.2     Comp.3      Comp.4
Standard deviation     2.0485788 0.49053911 0.27928554 0.153379074
Proportion of Variance 0.9246162 0.05301557 0.01718514 0.005183085
Cumulative Proportion  0.9246162 0.97763178 0.99481691 1.000000000
> plot(pcdat)
> loadings(pcdat)

Loadings:
   Comp.1 Comp.2 Comp.3 Comp.4
V1  0.362 -0.657 -0.581  0.317
V2        -0.730  0.596 -0.324
V3  0.857  0.176        -0.480
V4  0.359         0.549  0.751

               Comp.1 Comp.2 Comp.3 Comp.4
SS loadings      1.00   1.00   1.00   1.00
Proportion Var   0.25   0.25   0.25   0.25
Cumulative Var   0.25   0.50   0.75   1.00
```

First component explains 92% of data variance



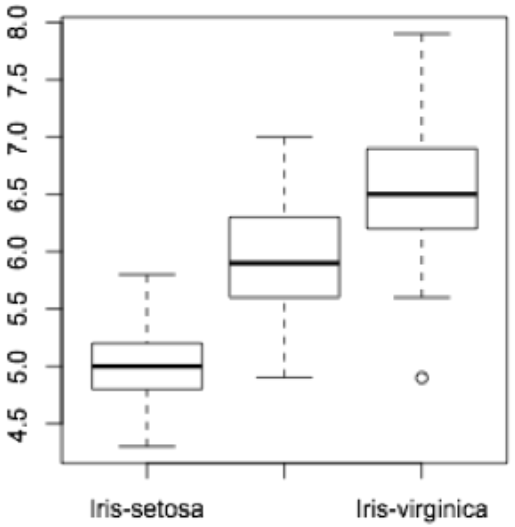
PCA example on Iris data

m=1, transform data to one dimension

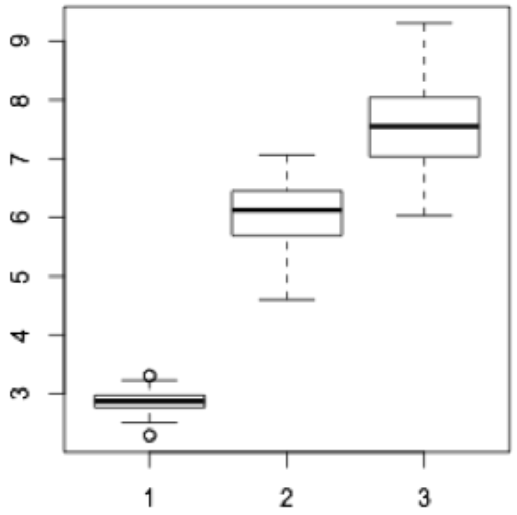
$$\mathbf{x} = [x_1, x_2, \ldots, x_p] \quad \text{(original instance)}$$
$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_p] \quad \text{(principal components)}$$
$$x'_1 = \mathbf{a}_1\mathbf{x} = \sum_{j=1}^{p} a_{1j}x_j$$

```
### Single example
> pcdat$loadings[,1]
         V1          V2          V3          V4
 0.36158968 -0.08226889  0.85657211  0.35884393

> d[1,1:4]
   V1  V2  V3  V4
1 5.1 3.5 1.4 0.2

> as.matrix(d[1,1:4]) %*% pcdat$loadings[,1]
      [,1]
1 2.827136

### All data
> as.matrix(d[,1:4]) %*% pcdat$loadings[,1]
```

Original data (1st variable)



Transformed data



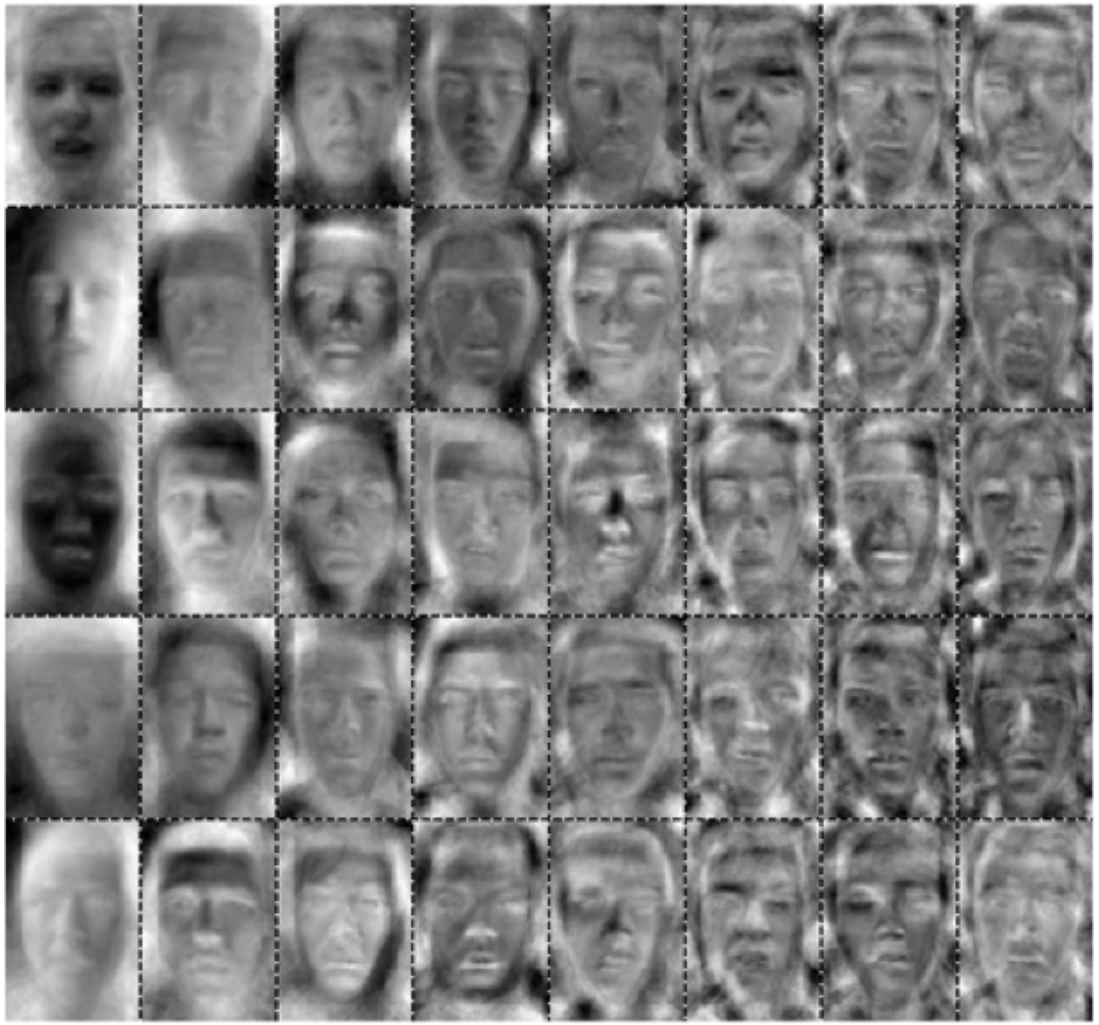
Example: Eigenfaces

PCA applied to images of human faces.

Reduce dimensionality to set of basis images.

All other images are linear combo of these “eigenpictures”.

Used for facial recognition.



First 40 PCA dimensions

PCA Summary

- Automatic method for dimensionality reduction
  - Linear transformation from the original coordinates, such that the variance of the data is maximized
  - The Principal components are decorrelated and used as the new feature representation
- Can be viewed as an unsupervised learning technique.
  - Define a model space, scoring function, search procedure that maximizes the scoring function.
- One of the oldest and most popular methods
  - Key limitation – **Linear transformation**
  - We will revisit these methods when discussing DL

Back to Distances..

Back to Distances..

1. Netflix's popular shows and movies are currently available for streaming world wide.

2. Netflix video content, now available everywhere, is well-liked by many.

3. Manchester united is a football team popular world wide

Back to Distances

- Reminder: sentence representation defined by V-dim vector
  - V is the size of the vocabulary (i.e., all the words in the language)
  - The i-th coordinate corresponds to the number of times the i-th word appears in a given sentence
- A sentence is a V dimensional point

Netflix's popular shows and movies are currently available for streaming world wide.

(0,0,..,1,0,…,1,1,0,1,…,0)

Back to Distances

- A sentence is a V dimensional point

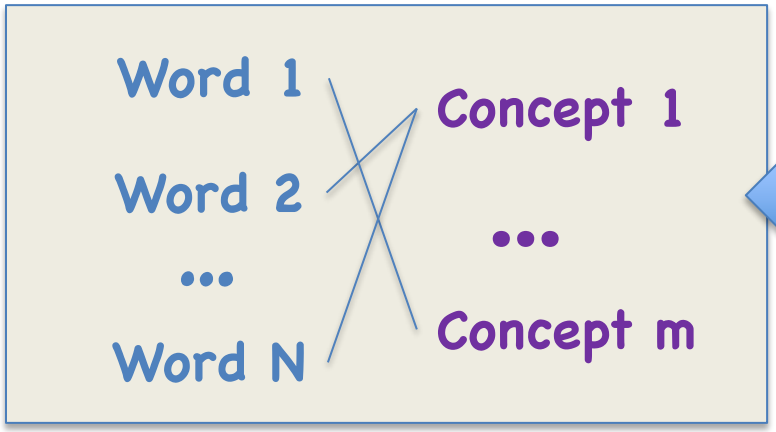
(0,0,..,1,0,…,1,1,0,1,…,0)

- This representation defines a set of coordinates
  - In the context of the PCA discussion, we refer to it as the ”original coordinates”, but instead of *x,y ,* we work in a higher dimensional space (e.g., x_1, x_2, …,x_V)
- *Each word defines a variable (or attribute). Given a corpus of news articles, what variables are likely to be highly correlated?*

- *What will happen if we apply PCA over this dataset?*

Back to Distances

- Key intuition:
  - Similar sentences will use similar vocabulary, as a result related words will co-occur a lot
  - Meaning, their co-variance is likely to be high.
  - PCA aims to decorrelate variables, such that highly correlated variables are represented in the same principle components.
  - As a result, we get a new representation, mapping similar documents into the same space.



Distributed Representation: every attribute is a collection of concepts, and every concept is a collection of attributes.

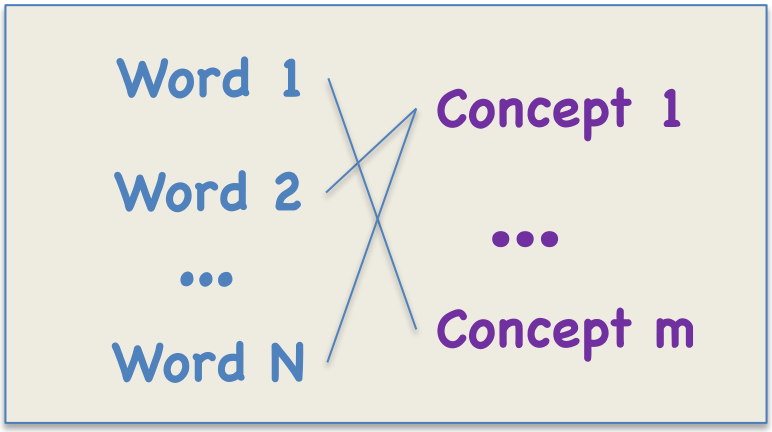
Cool!

Representation Learning

- We just came up with a way to automatically capture semantic relationships in documents.
  - Very similar to a known technique called LSA
- ***Ok, cool.. But now what?***
- ***Can it work for any other domains?***
- Representation learning is a very popular unsupervised learning technique that allows us to identify patterns in the data without explicit supervision.
- These patterns can be used directly (document retrieval, paraphrase detection) or as the basis for advanced applications, as a feature representation.
- ***We'll revisit these methods later in the semester***

Final Thought

- Recall our discussion about bias introduced into machine learning.
- What is the danger with such approaches?



And now –

Clustering!

Descriptive models

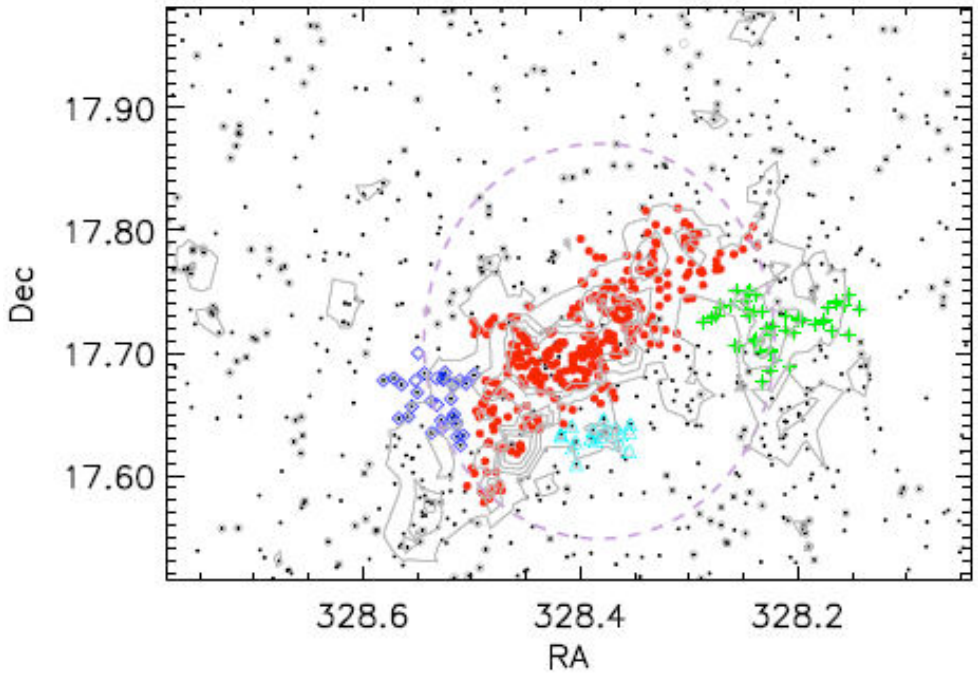
- Descriptive models summarize the data
  - Global summary
  - Model main features of the data
- **Two main approaches:**
  - Cluster analysis (*we will focus mostly on this topic*)
  - Density estimation

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*?)

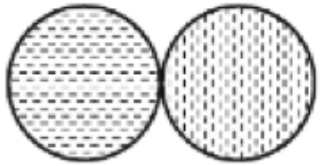
was it an issue in the supervised setting?

  - 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)

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

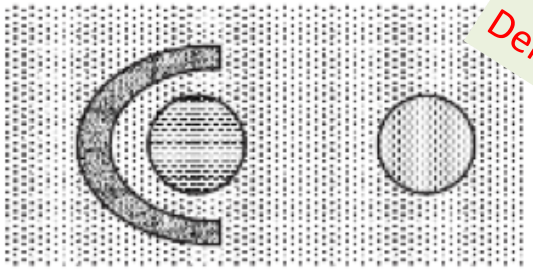


(b) Center-based clusters. Each point is closer to the center of its cluster than to the center of any other cluster.

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.

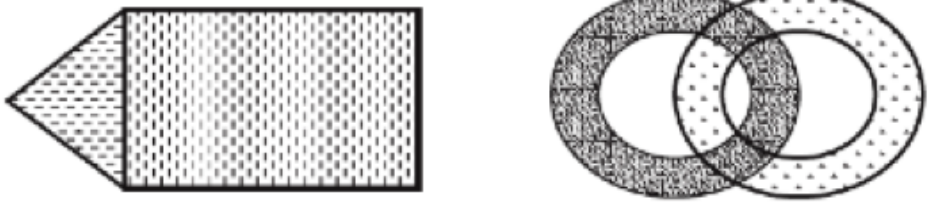
Density-based



(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)

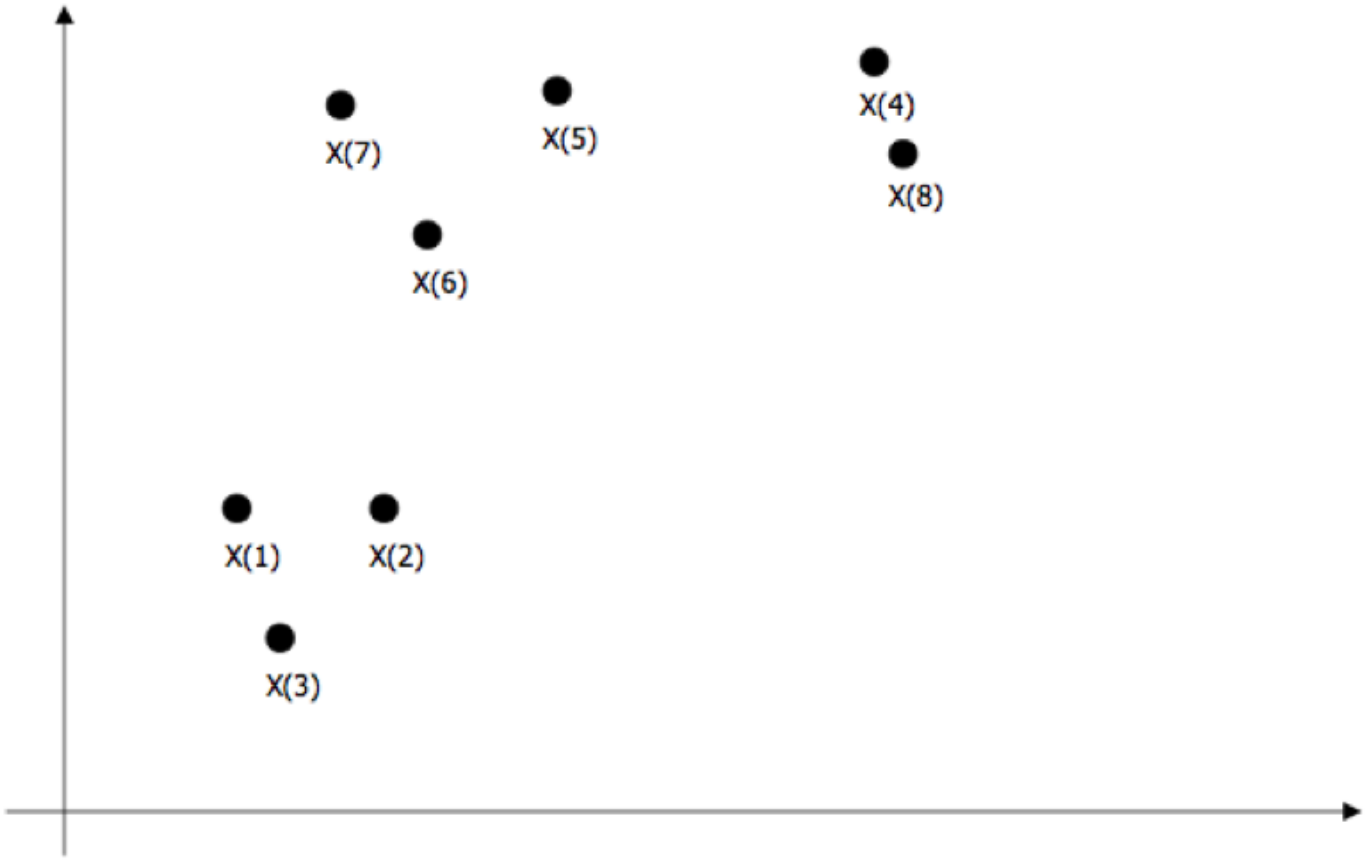
Partition-based clustering

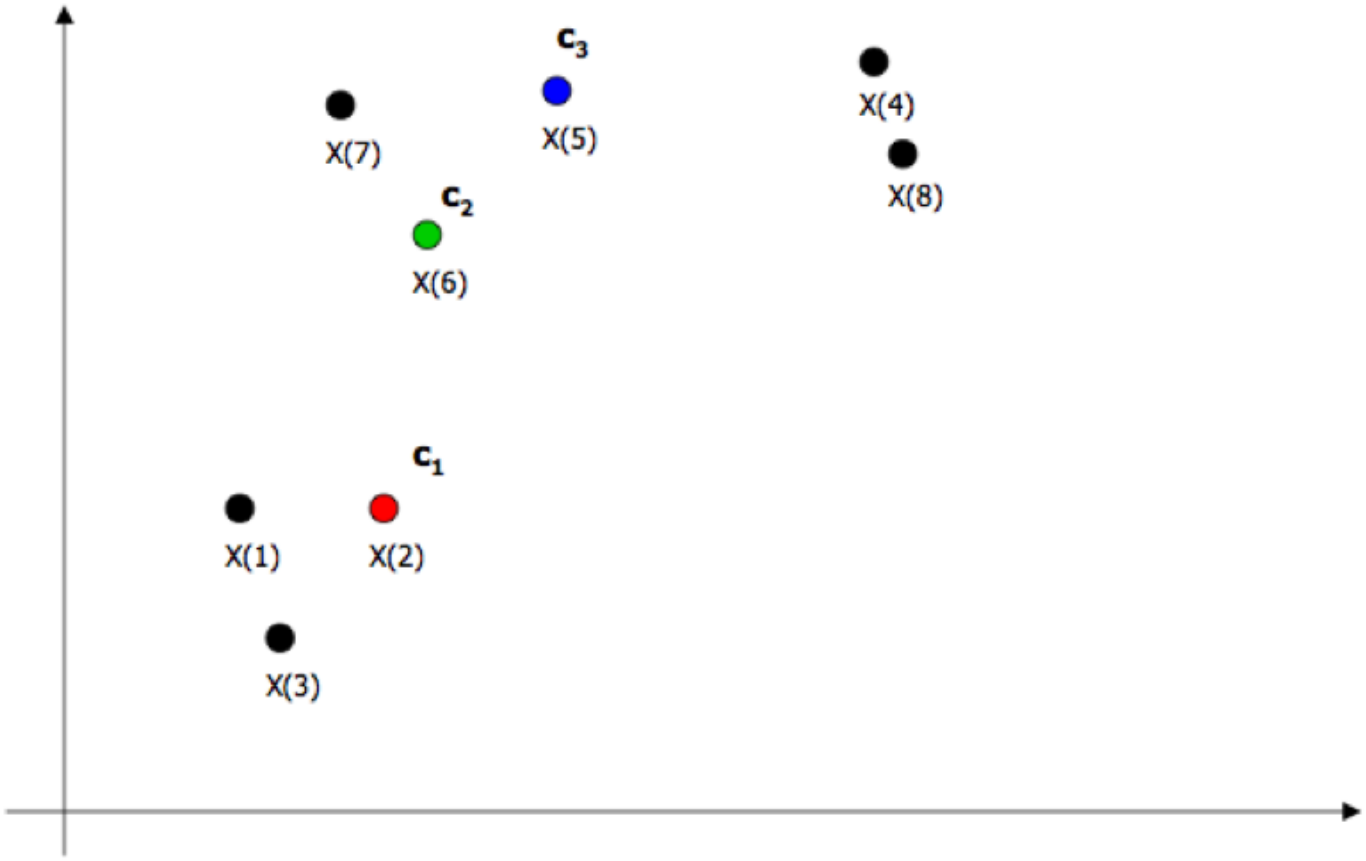
Partition-based

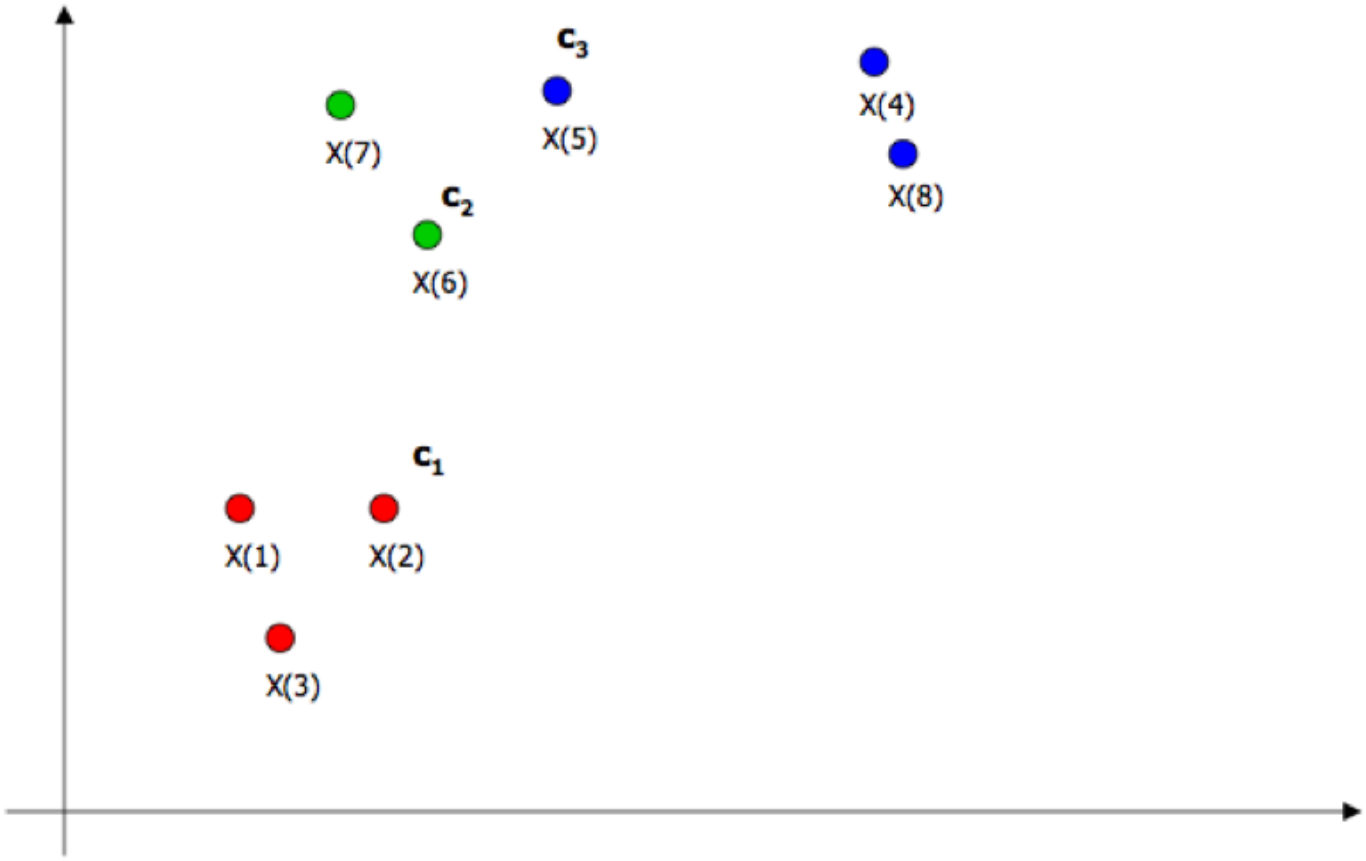
- **Input**: data D={**x**(1),**x**(2),...,**x**(n)}
- **Output**: k clusters C={C_1,...,C_k} such that each x(i) is assigned to a unique C_j
- **Evaluation**: Score(C,D) is maximized/minimized
- Combinatorial optimization: search among n^k allocations of n objects into k classes to maximize score function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

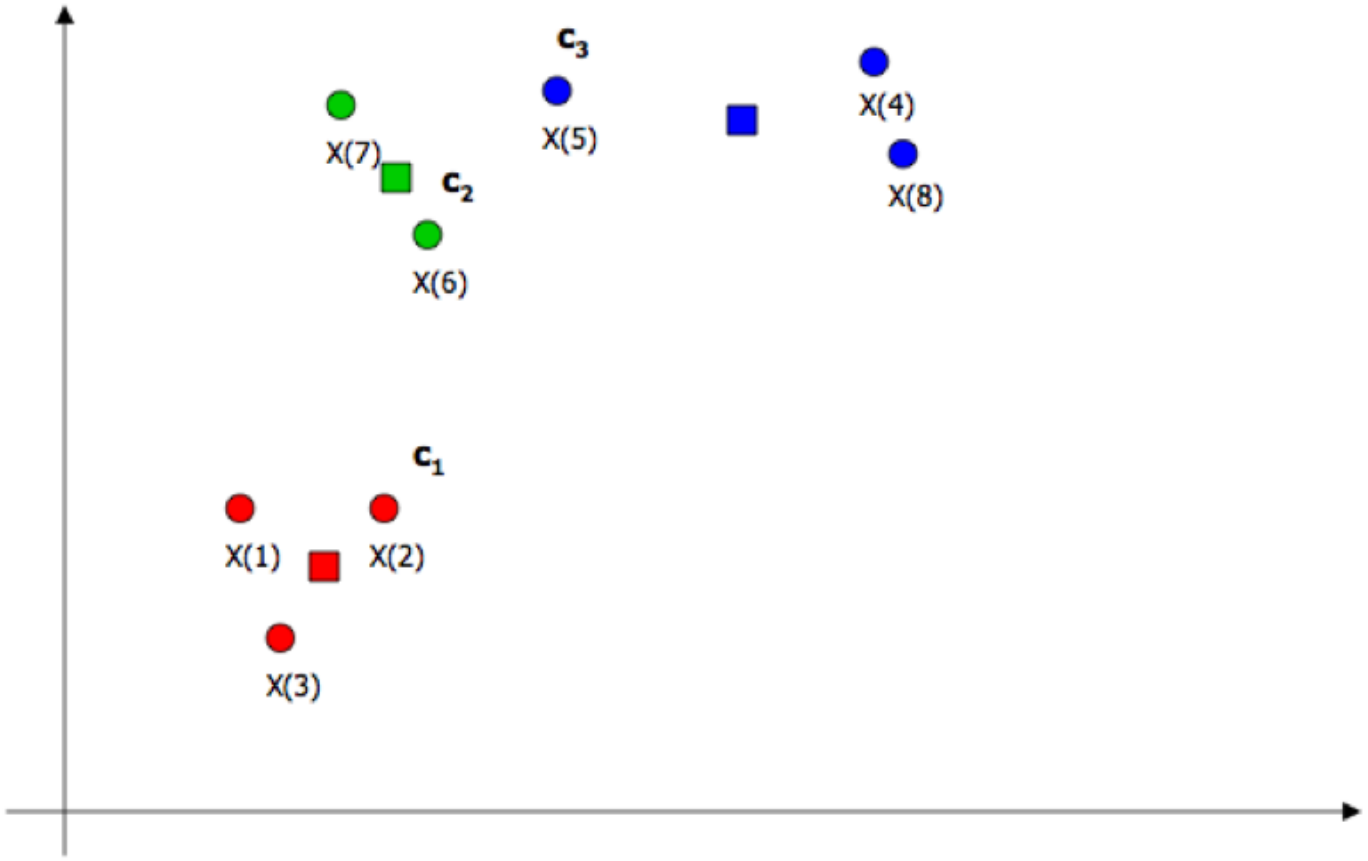
Example: K-means

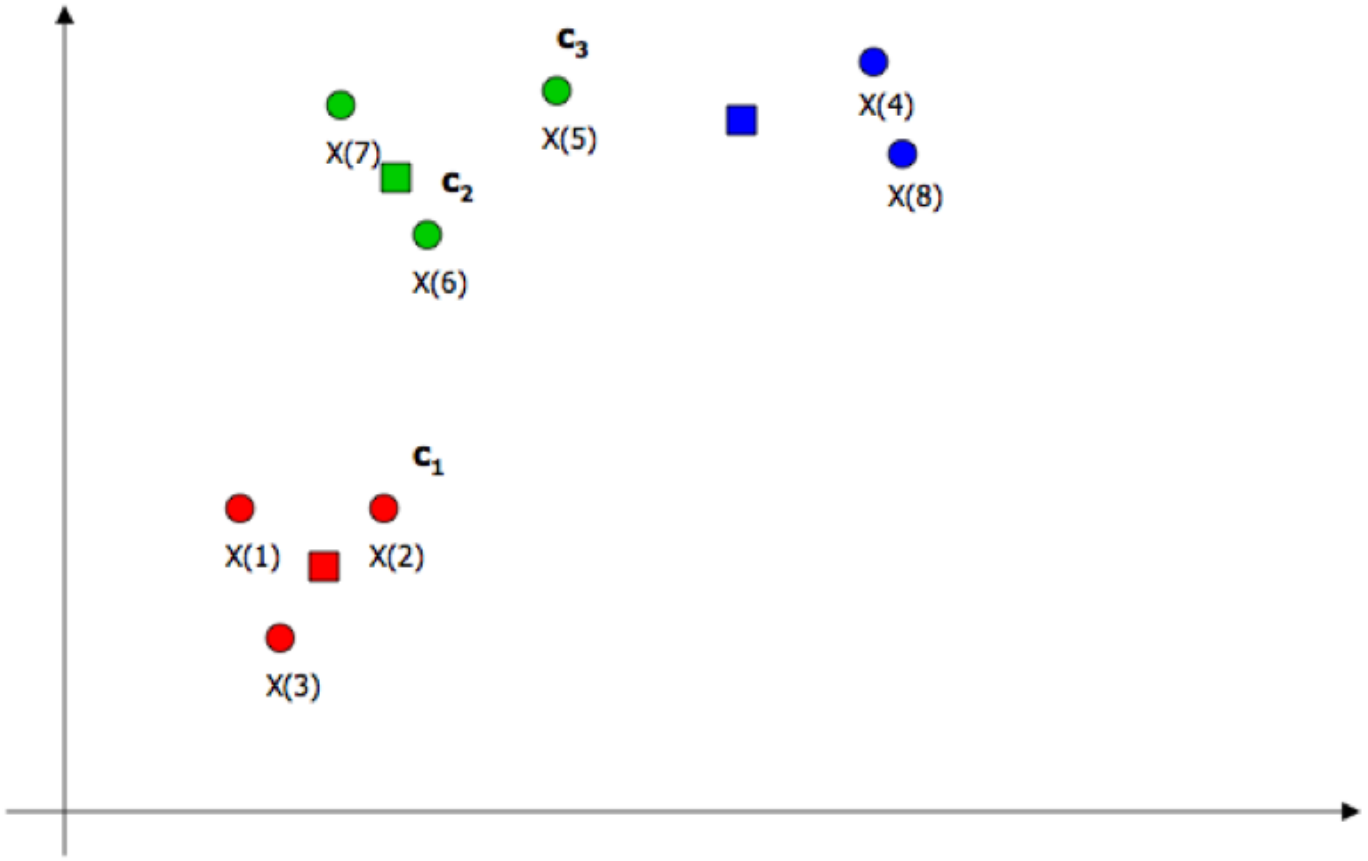
- **Algorithm idea:**
  - Start with k randomly chosen centroids
  - Repeat until no changes in assignments
    - Assign instances to closest centroid
    - Recompute cluster centroids

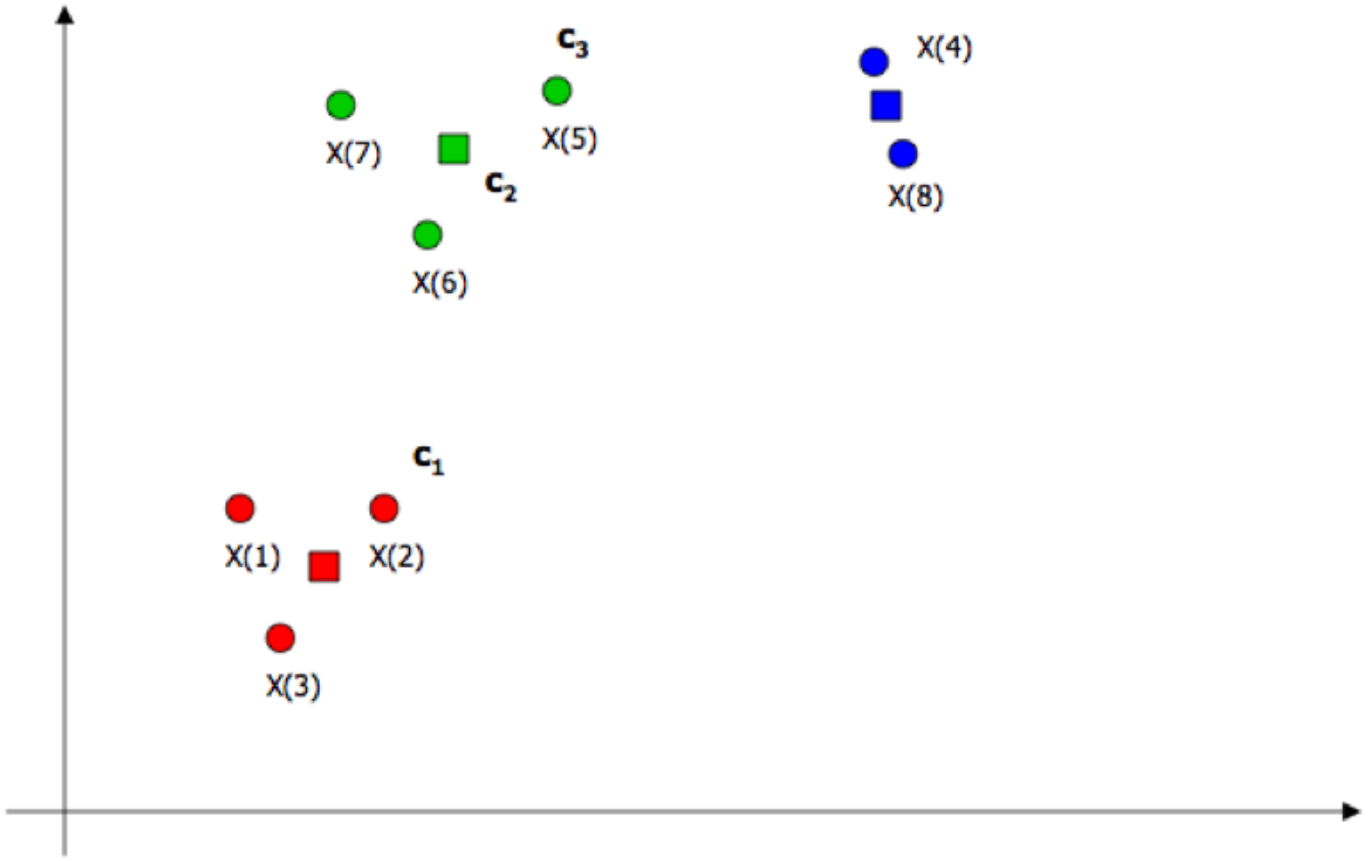

X(7)
X(5)
X(4)
X(8)
X(6)
X(1)
X(2)
X(3)

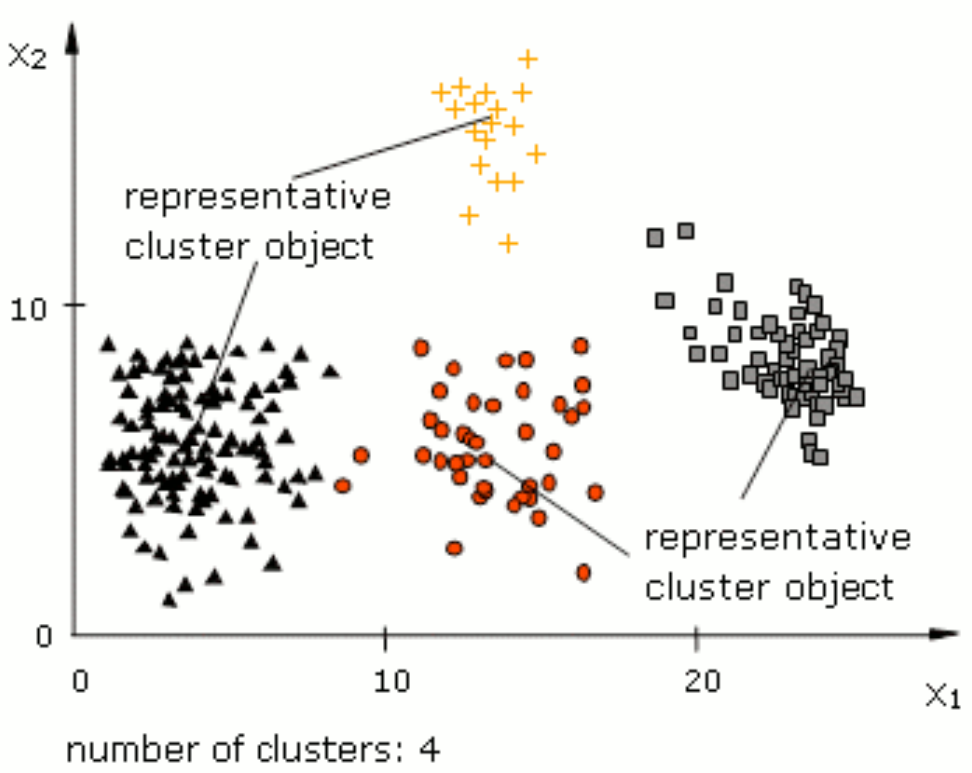

c3
X(7)
X(5)
X(4)
X(8)
c2
X(6)
c1
X(1)
X(2)
X(3)


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)



Groups represented by *canonical* item description(s)

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

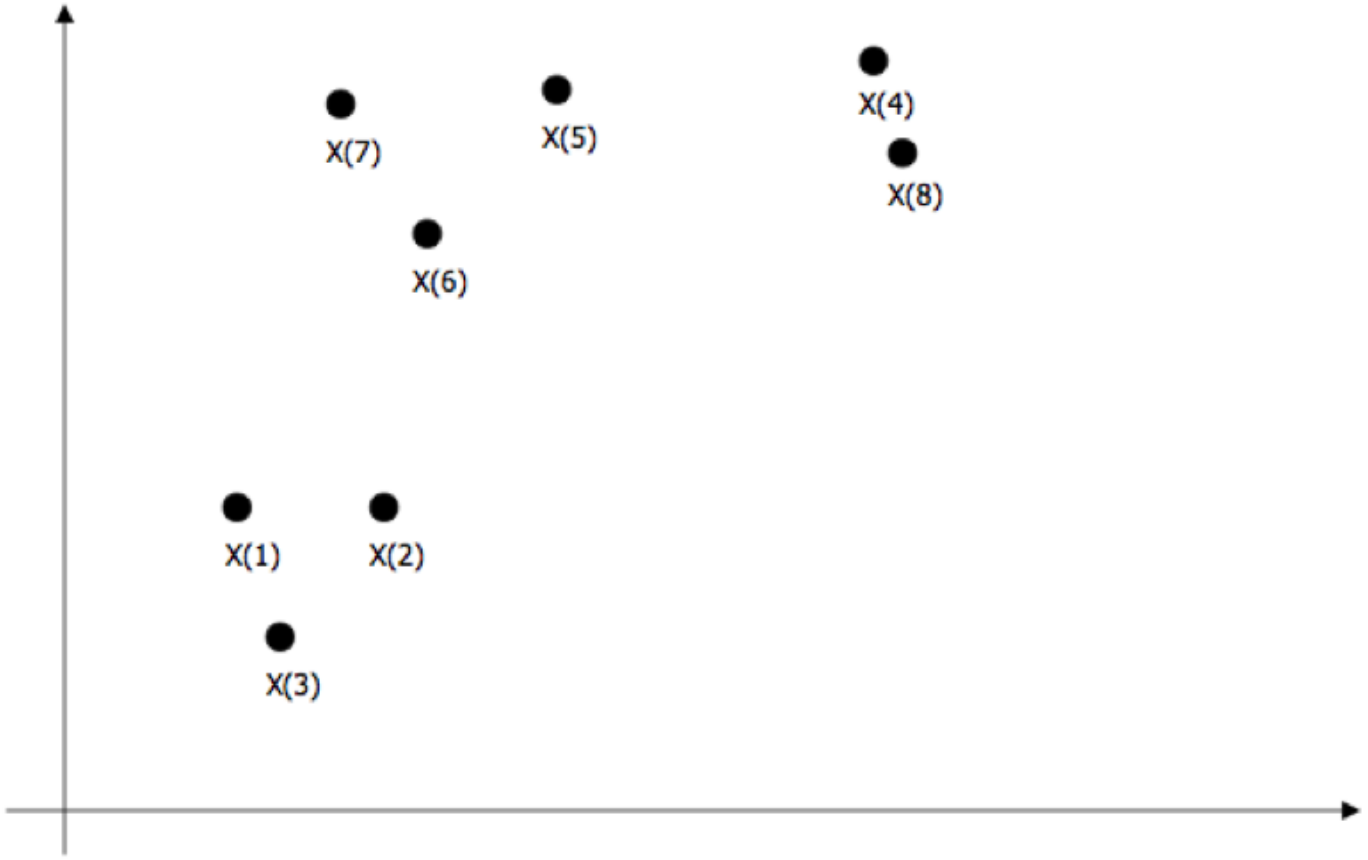
Input: Dataset D, number clusters k

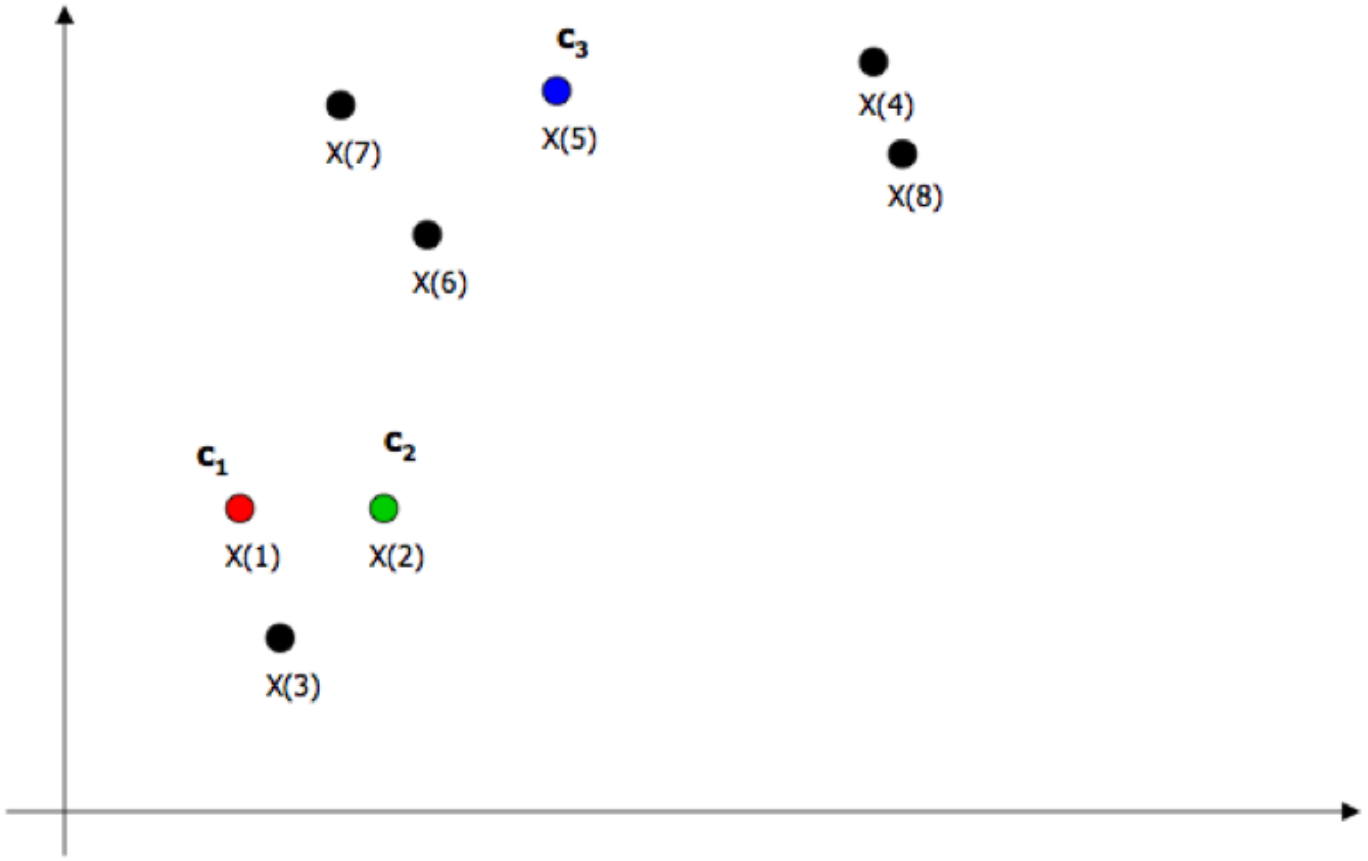
Output: Set of cluster representatives C, cluster membership vector $\mathbf{m}$

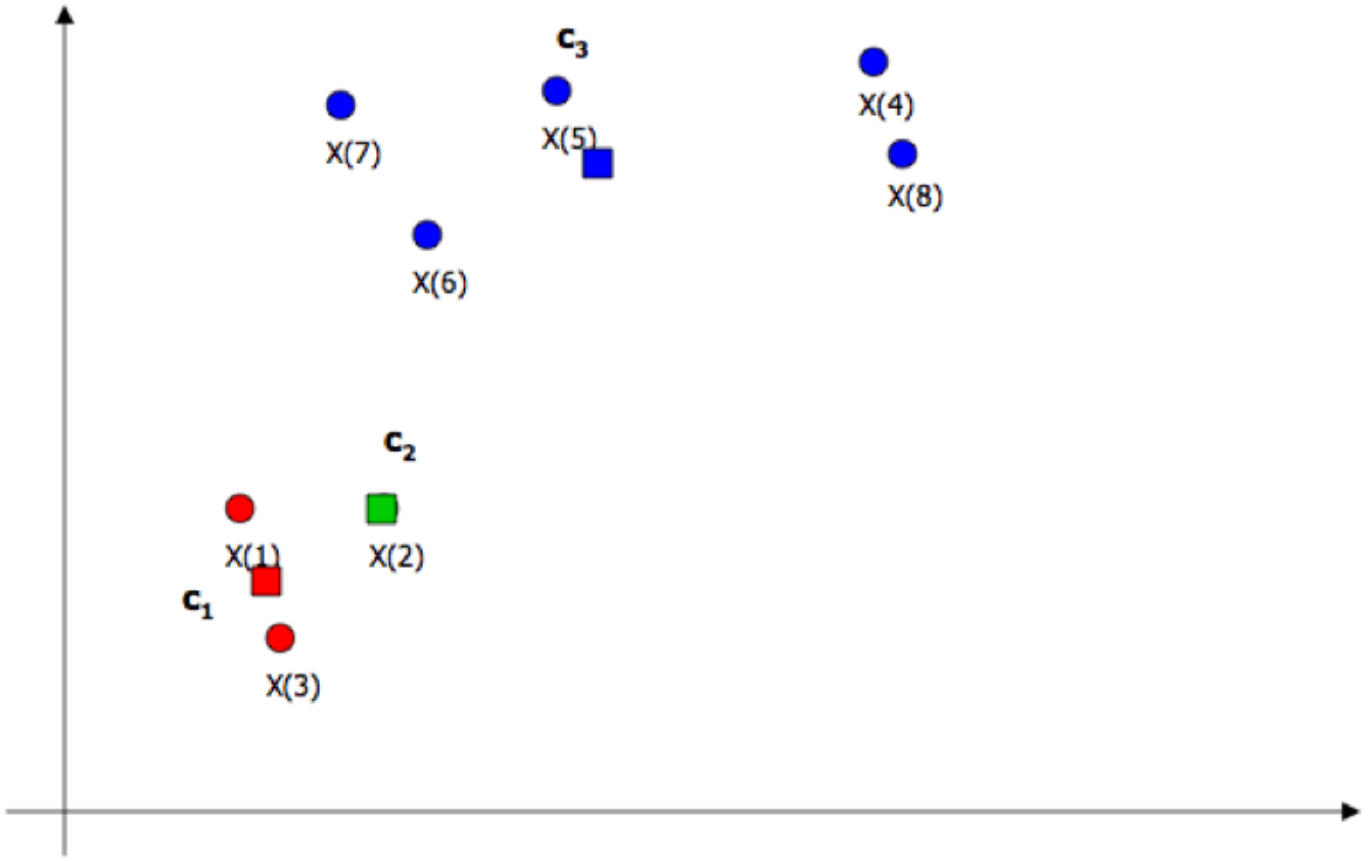
```
    /* Initialize cluster representatives C */
    Randomly choose k data points from D
    Use these k points as initial set of cluster representatives C
    repeat
        /* Data Assignment */
        Reassign points in D to closest cluster mean
        Update m such that m_i is cluster ID of 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

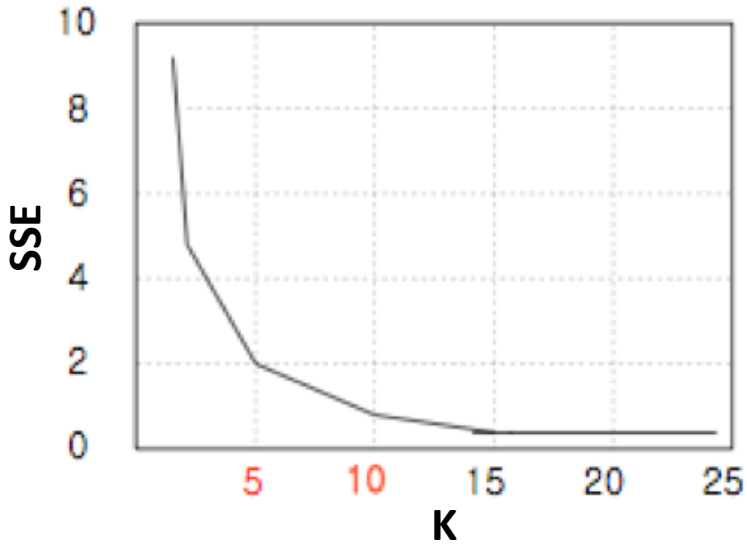
Variations

- **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*

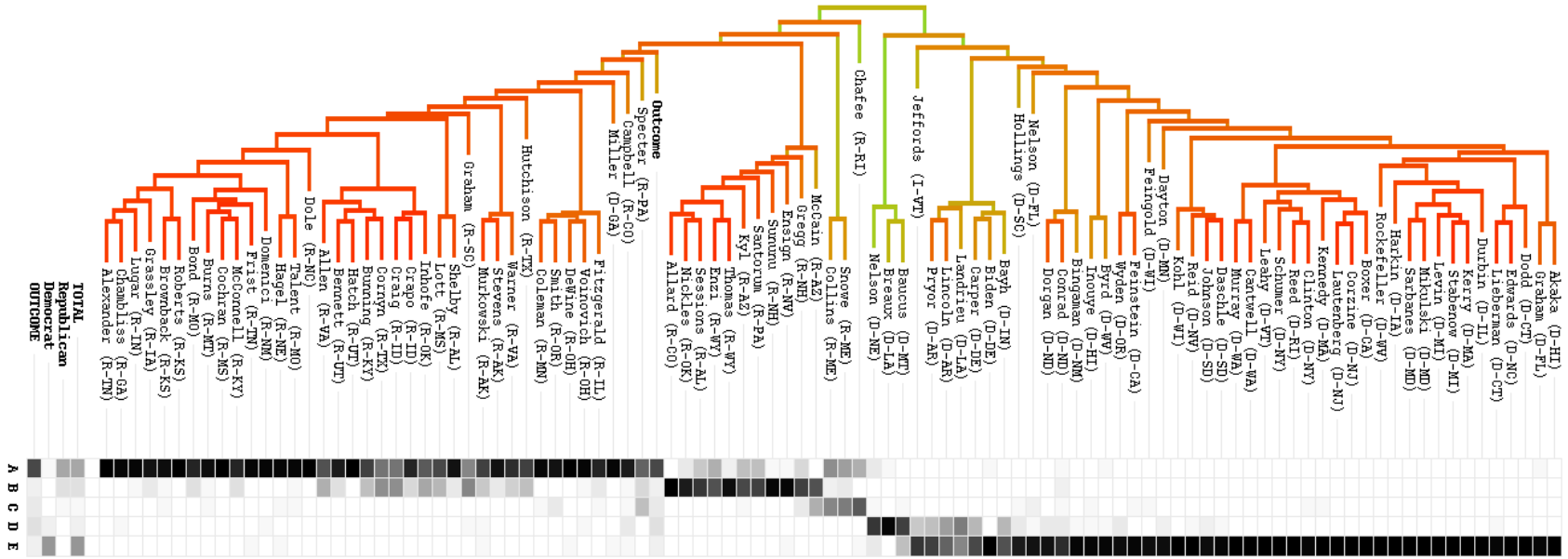
Hierarchical clustering

Hierarchical methods

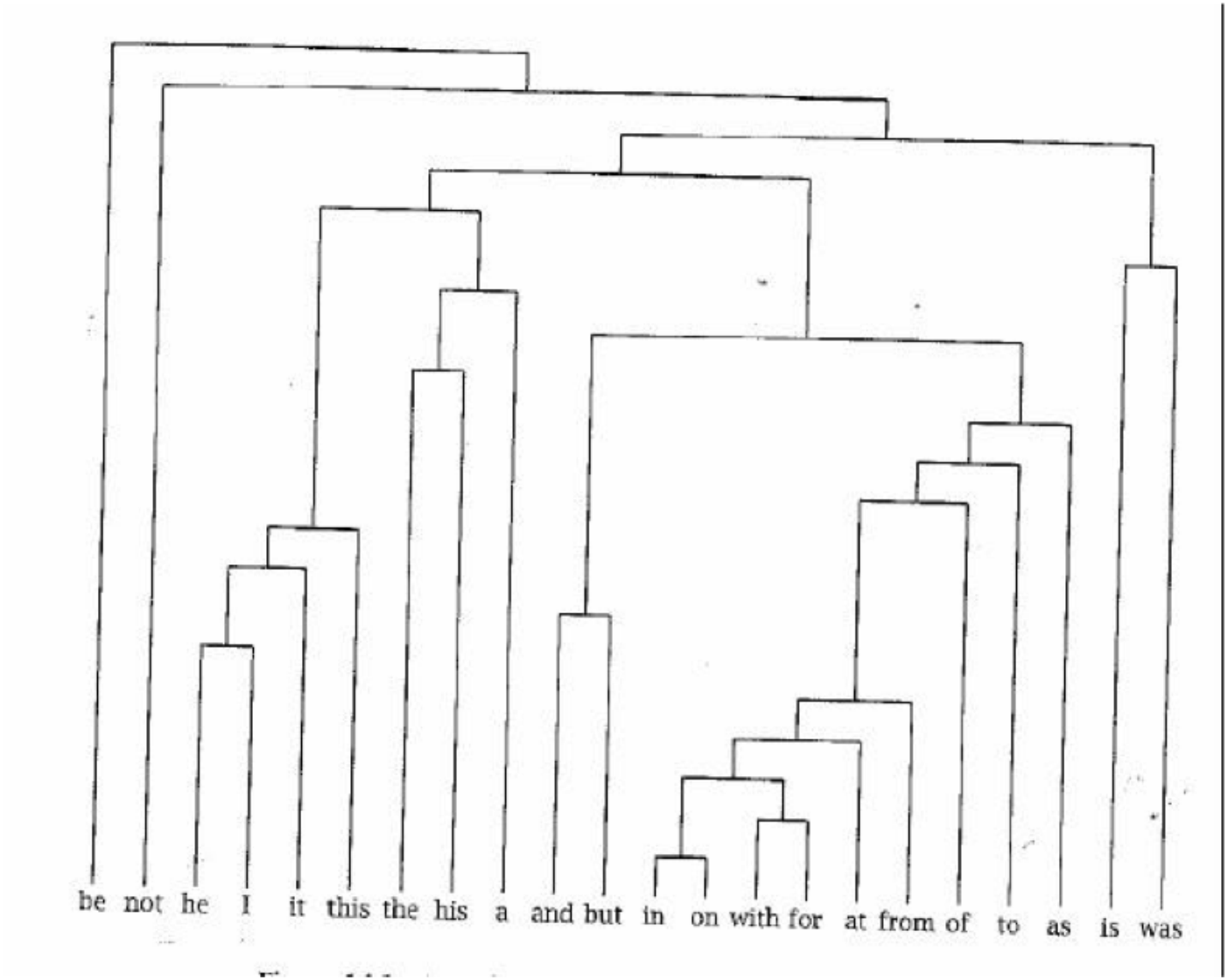
- Construct a *hierarchy* of nested clusters rather than picking k beforehand
- **Approaches**:
  - Agglomerative: merge clusters successively
  - Divisive: divided clusters successively
- Dendrogram (tree diagram) depicts sequences of merges or splits and height indicates distance

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 ∪ C_j
  - Remove C_j



Clustering represented with dendogram

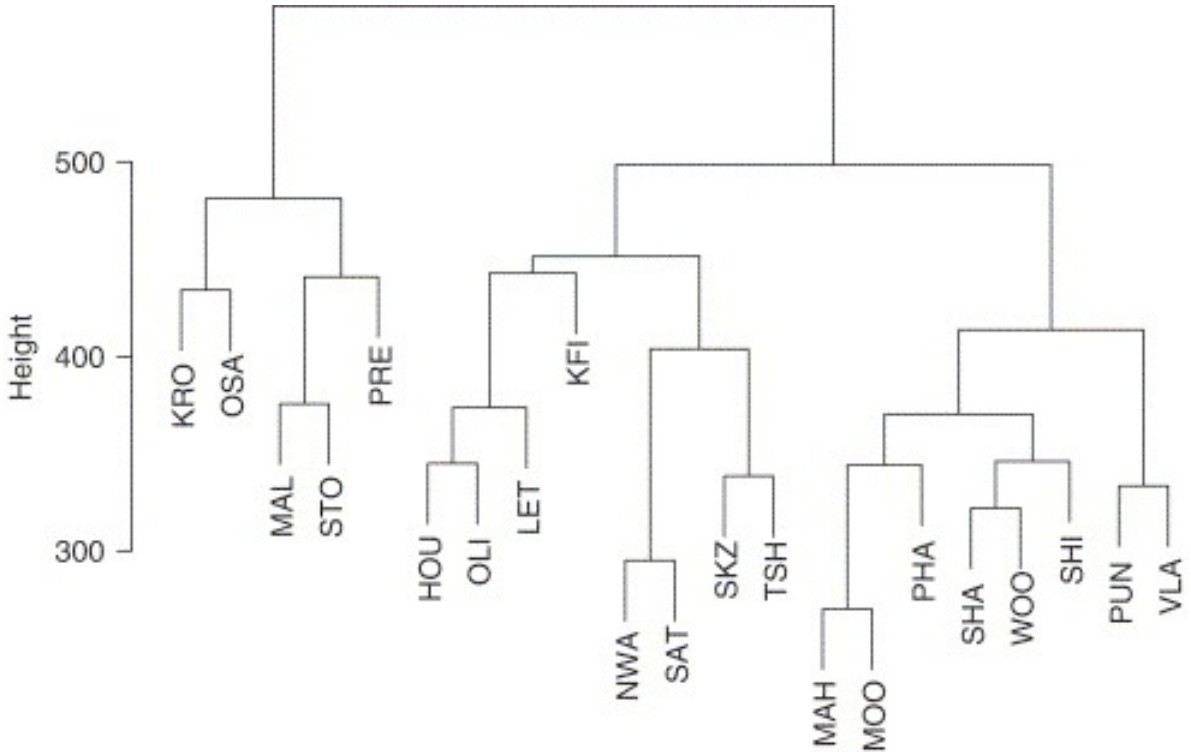


A hierarchical clustering of 22 frequent English words represented as a dendrogram.

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

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.



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*