CS559 Lecture 14

Clustering

Reading: Chapter 9, Bishop book

Non-parametric unsupervised learning

- **Parametric unsupervised learning**
  - Equivalent to density estimation with a mixture of (Gaussian) components
- **Non-parametric unsupervised learning**
  - No density functions are considered in these methods
  - Instead, we are concerned with finding natural groupings (clusters) in a dataset

Clustering

Groups together "similar" instances in the data sample

Basic clustering problem:

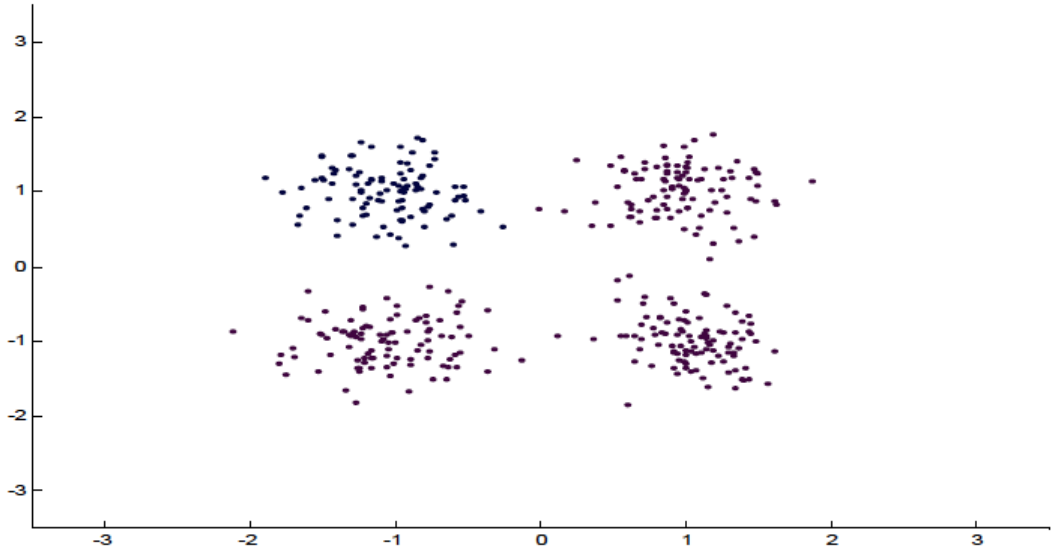
- distribute data into *k* different groups such that data points similar to each other are in the same group
- Similarity between data points is defined in terms of some distance metric (can be chosen)

Clustering is useful for:

- **Similarity/Dissimilarity analysis**
  Analyze what data points in the sample are close to each other
- **Dimensionality reduction**
  High dimensional data replaced with a group (cluster) label

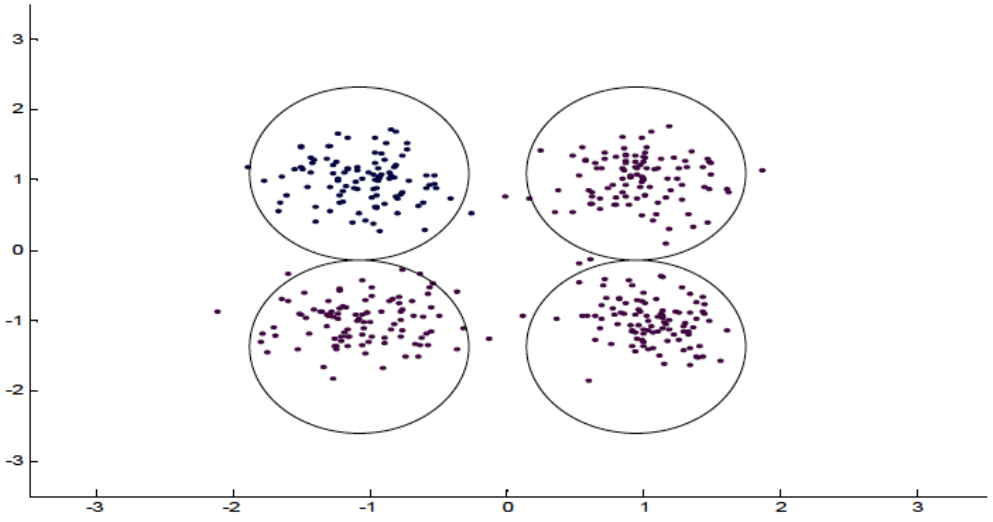
Clustering example

- We see data points and want to partition them into groups
- Which data points belong together?



Clustering example

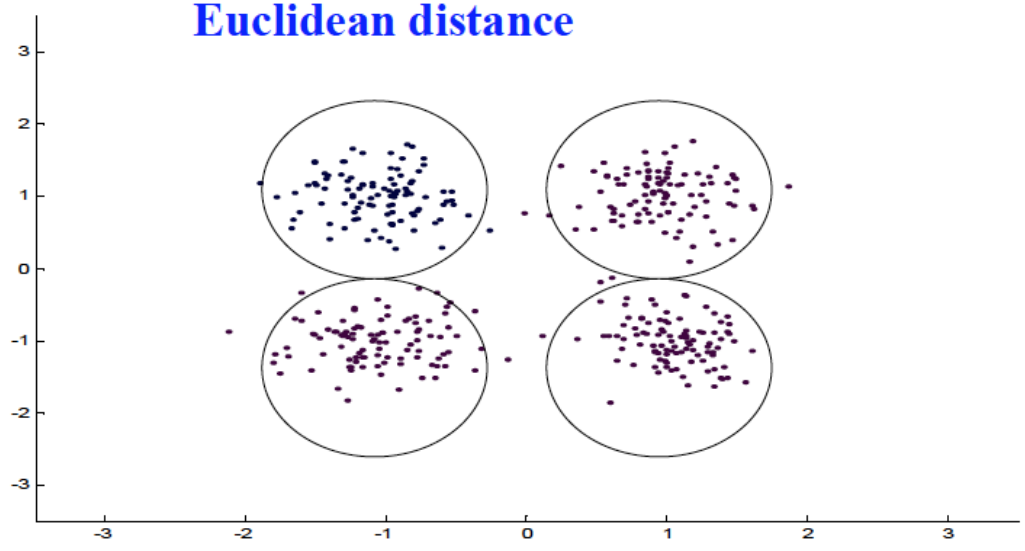
- We see data points and want to partition them into the groups
- Which data points belong together?



Clustering example

- We see data points and want to partition them into the groups
- Requires a distance measure to tell us what points are close to each other and are in the same group

Euclidean distance



Clustering example

- A set of patient cases
- We want to partition them into groups based on similarities

Patient #	Age	Sex	Heart Rate	Blood pressure …
Patient 1	55	M	85	125/80
Patient 2	62	M	87	130/85
Patient 3	67	F	80	126/86
Patient 4	65	F	90	130/90
Patient 5	70	M	84	135/85

Clustering example

- A set of patient cases
- We want to partition them into the groups based on similarities

Patient #	Age	Sex	Heart Rate	Blood pressure …
Patient 1	55	M	85	125/80
Patient 2	62	M	87	130/85
Patient 3	67	F	80	126/86
Patient 4	65	F	90	130/90
Patient 5	70	M	84	135/85

How to design the distance metric to quantify similarities?

Distance measures.

Assume pure real-valued data-points:

12	34.5	78.5	89.2	19.2
23.5	41.4	66.3	78.8	8.9
33.6	36.7	78.3	90.3	21.4
17.2	30.1	71.6	88.5	12.5
…				

What distance metric to use?

Distance measures.

Assume pure binary values data:

0	1	1	0	1
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1
…				

What distance metric to use?

Proximity measures (1)

- **Definition of metric**
  - A measuring rule d(x,y) for the distance between two vectors x and y is considered a metric if it satisfies the following properties

$$\begin{aligned} &d(x,y) \geq d_0 \\ &d(x,y) = d_0 \ \text{ iff } \ x = y \\ &d(x,y) = d(y,x) \\ &d(x,y) \leq d(x,z) + d(z,y) \end{aligned}$$

  - If the metric has the property

$$d(ax,ay) = |a| \cdot d(x,y)$$

    - then it is called a norm and denoted d(x,y)=||x-y||

- **The most general form of distance metric is the power norm**

$$\|x-y\|_{p/r} = \left(\sum_{i=1}^{D} |x_i - y_i|^p \right)^{1/r}$$

Proximity measures (2)

- **Most of the commonly used metrics are derived from the power norm**
  - **Minkowski metric (L_k norm)**

$$\|x - y\|_k = \left(\sum_{i=1}^{D} |x_i - y_i|^k \right)^{1/k}$$

    - The choice of an appropriate value of k depends on the amount of emphasis that you would like to give to the larger differences between dimensions
  - **Manhattan or city-block distance (L_1 norm)**

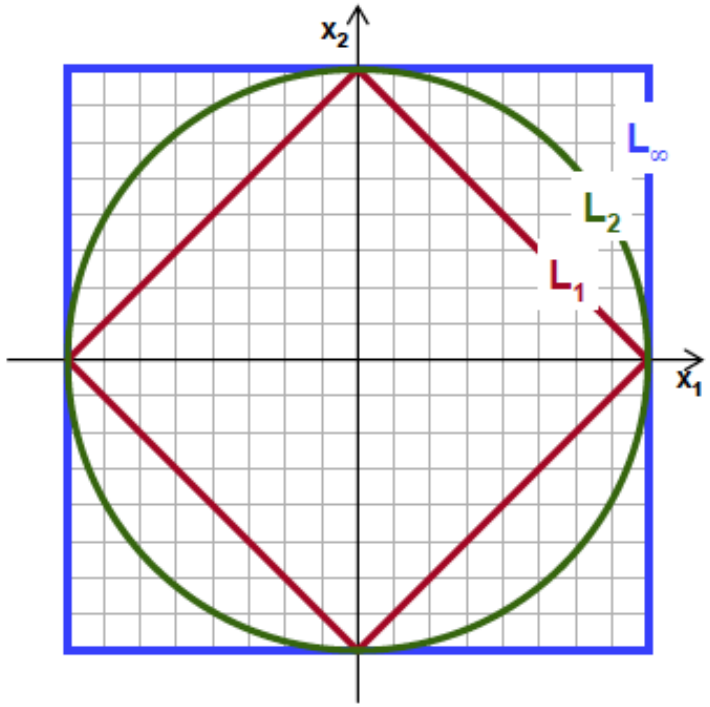
$$\|x - y\|_{c\text{-}b} = \sum_{k=1}^{D} |x_k - y_k|$$

    - When used with binary vectors, the L_1 norm is known as the Hamming distance
  - **Euclidean norm (L_2 norm)**

$$\|x - y\|_e = \left(\sum_{k=1}^{D} |x_k - y_k|^2 \right)^{1/2}$$

  - **Chebyshev distance (L_∞ norm)**

$$\|x - y\|_c = \max_{1 \le i \le D} |x_i - y_i|$$



Contours of equal distance

Proximity measures (3)

- **Other metrics are also popular**
  - **Quadratic distance**

$$d(x,y)=\sqrt{(x-y)^T B(x-y)}$$

    - The Mahalanobis distance is a particular case of this distance
  - **Canberra metric** (for non-negative features)

$$d_{ca}(x,y)=\sum_{i=1}^{D}\frac{|x_i-y_i|}{x_i+y_i}$$

  - **Non-linear distance**

$$d_N(x,y)=\begin{cases}0 & \text{if } d_e(x,y)<T\\ H & \text{if } d_e(x,y)\geq T\end{cases}$$

    - where T is a threshold and H is a distance. An appropriate choice for H and T **for feature selection** is that they should satisfy

$$H=\frac{\Gamma(p/2)}{T^p 2\sqrt{\pi^p}}$$

    - and that T satisfies the unbiasedness and consistency conditions of the Parzen estimator: $T^pN\to\infty$, $T\to 0$ as $N\to\infty$

Distance measures

Generalized distance metric:

$$d^2(\mathbf{a}, \mathbf{b}) = (\mathbf{a} - \mathbf{b})\boldsymbol{\Gamma}^{-1}(\mathbf{a} - \mathbf{b})^T$$

Γ semi-definite positive matrix

Γ^{-1} is a matrix that weights attributes proportionally to their importance. Different weights lead to a different distance metric.

If $\Gamma = I$ we get squared Euclidean

$\Gamma = \Sigma$ (covariance matrix) – we get the Mahalanobis distance that takes into account correlations among attributes

Proximity measures (4)

- **Notice that the above distance metrics are measures of DISSIMILARITY**
- **Some measures OF SIMILARITY also exist**
  - **Inner product**

$$s_{INNER}(x,y) = x^T y$$

    - The inner product is used when the vectors x and y are normalized, so that they have the same length
  - **Correlation coefficient**

$$s_{CC}(x,y) = \frac{\sum_{i=1}^{D}(x-\bar{x})(y-\bar{y})}{\left[\sum_{i=1}^{D}(x-\bar{x})^2 \sum_{i=1}^{D}(y-\bar{y})^2\right]^{1/2}}$$

  - **Tanimoto measure** (for binary-valued vectors)

$$s_T(x,y) = \frac{x^T y}{\|x\|^2 + \|y\|^2 - x^T y}$$

Distance measures.

Combination of real-valued and categorical attributes

Patient #	Age	Sex	Heart Rate	Blood pressure …
Patient 1	55	M	85	125/80
Patient 2	62	M	87	130/85
Patient 3	67	F	80	126/86
Patient 4	65	F	90	130/90
Patient 5	70	M	84	135/85

What distance metric to use?

Distance measures.

Combination of real-valued and categorical attributes

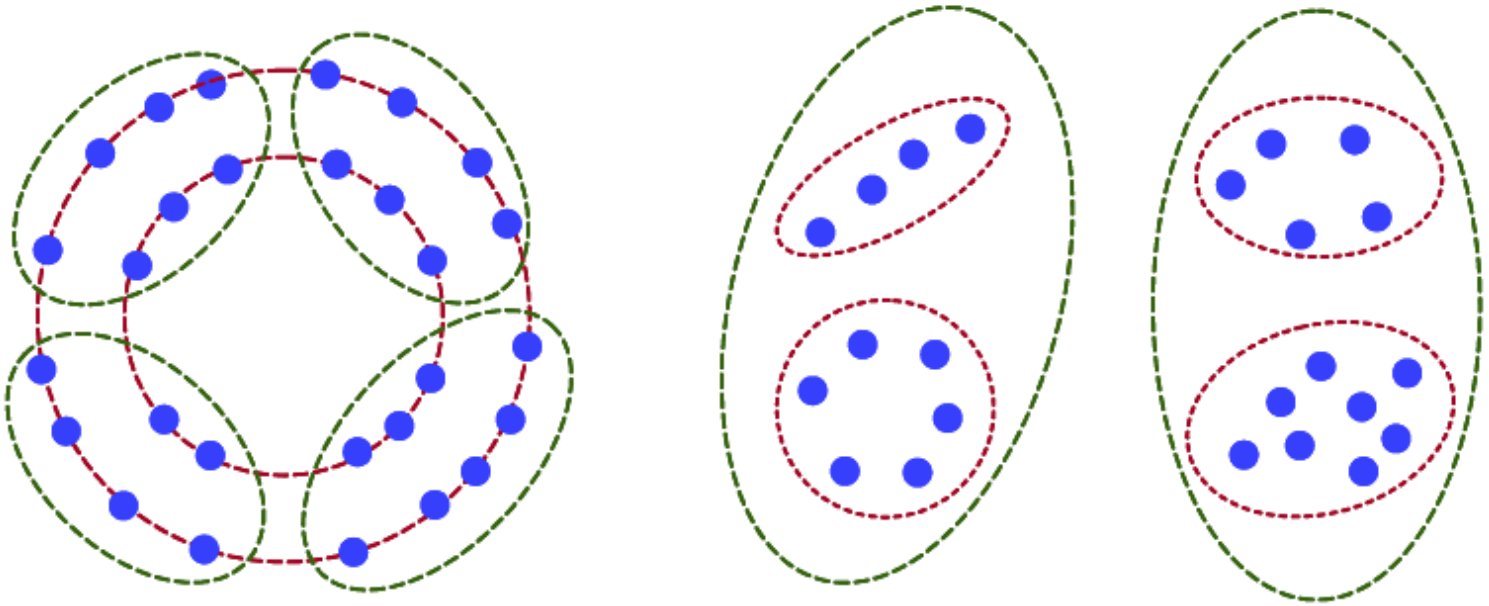
Patient #	Age	Sex	Heart Rate	Blood pressure …
Patient 1	55	M	85	125/80
Patient 2	62	M	87	130/85
Patient 3	67	F	80	126/86
Patient 4	65	F	90	130/90
Patient 5	70	M	84	135/85

What distance metric to use?

A weighted sum approach: e.g. a mix of Euclidian and Hamming distances for subsets of attributes

Cluster validity

- **The choice of (dis)similarity measure and criterion function will have a major impact on the final clustering produced by the algorithms**
  - Notice that the validity of the final cluster solution is highly subjective
    - This is in contrast with supervised training, where a clear objective function is known: Bayes risk.
  - Example
    - Which are the meaningful clusters in these cases?
    - How many clusters should be considered?



- **A number of quantitative methods for cluster validity are proposed in [Theodoridis and Koutrombas, 1999]**

Clustering

Clustering is useful for:

- **Similarity/Dissimilarity analysis**

  Analyze what data points in the sample are close to each other
- **Dimensionality reduction**

  High dimensional data replaced with a group (cluster) label
- **Data reduction:** Replaces many datapoints with the point representing the group mean

Problems:

- Pick the correct similarity measure (problem specific)
- Choose the correct number of groups
  - Many clustering algorithms require us to provide the number of groups ahead of time

Criterion function for clustering

- **Once a (dis)similarity measure has been determined, we need to define a criterion function to be optimized**
  - The most widely used criterion function for clustering is the sum-of-square-error

$$J_{MSE} = \sum_{i=1}^{C} \sum_{x \in \omega_i} |x - \mu_i|^2 \qquad \text{where } \mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x$$

    - This criterion measures how well the data set $X=\{x^{(1}, x^{(2}, \ldots, x^{(N}\}$ is represented by the cluster centers $\mu=\{\mu^{(1}, \mu^{(2}, \ldots, \mu^{(C}\}$ $(C<N)$
    - Clustering methods that use this criterion are called minimum variance methods
  - Other criterion functions exist, based on the scatter matrices used in Linear Discriminant Analysis
    - For details, refer to [Duda, Hart and Stork, 2001]

Iterative optimization

- **Once a criterion function has been defined, we must find a partition of the data set that minimizes the criterion**
  - Exhaustive enumeration of all partitions, which guarantees the optimal solution, is unfeasible
    - For example, a problem with 5 clusters and 100 examples yields 10^{67} partitionings
- **The common approach is to proceed in an iterative fashion**
  1. Find some reasonable initial partition and then
  2. Move samples from one cluster to another in order to reduce the criterion function
- **These iterative methods produce sub-optimal solutions but are computationally tractable**

Clustering algorithms

- **K-means algorithm**
  - **suitable** only when data points have continuous values; groups are defined in terms of cluster centers (also called **means**). Refinement of the method to categorical values: **K-medoids**
- **Probabilistic methods (with EM)**
  - **Latent variable models**: class (cluster) is represented by a latent (hidden) variable value
  - Every point goes to the class with the highest posterior
  - **Examples:** mixture of Gaussians, Naïve Bayes with a hidden class
- **Hierarchical methods**
  - **Agglomerative**
  - **Divisive**

K-means

K-Means algorithm:

Initialize randomly *k* values of means (centers)

Repeat two steps until no change in the means:

- Partition the data according to the current set of means (using the similarity measure)
- Move the means to the center of the data in the current partition

Stop when no change in the means

Properties:

- Minimizes the sum of **squared center-point distances** for all clusters
- The algorithm always converges (to the local optima).

The k-means algorithm

- **The k-means algorithm is a simple clustering procedure that attempts to minimize the criterion function J_{MSE} in an iterative fashion**

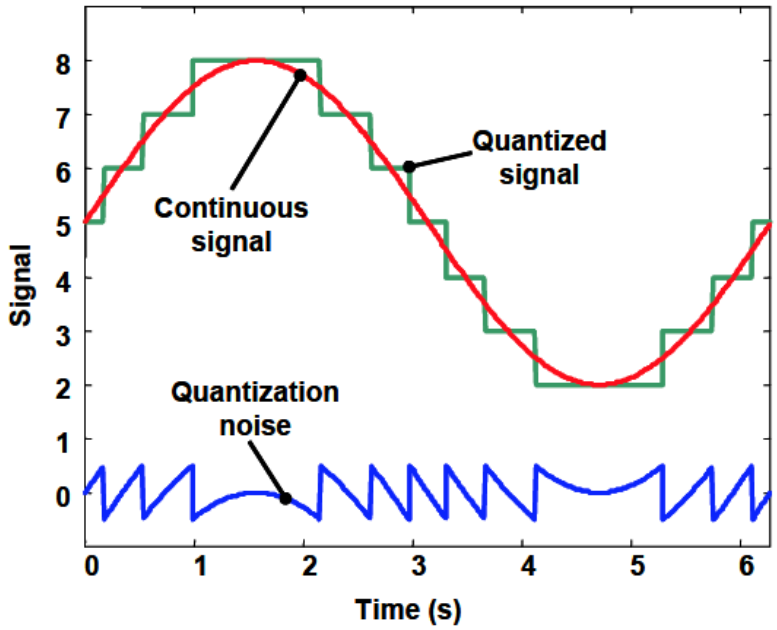
$$J_{MSE} = \sum_{i=1}^{C} \sum_{x \approx \omega_i} |x - \mu_i|^2 \qquad \text{where } \mu_i = \frac{1}{N_i} \sum_{x \approx \omega_i} x$$

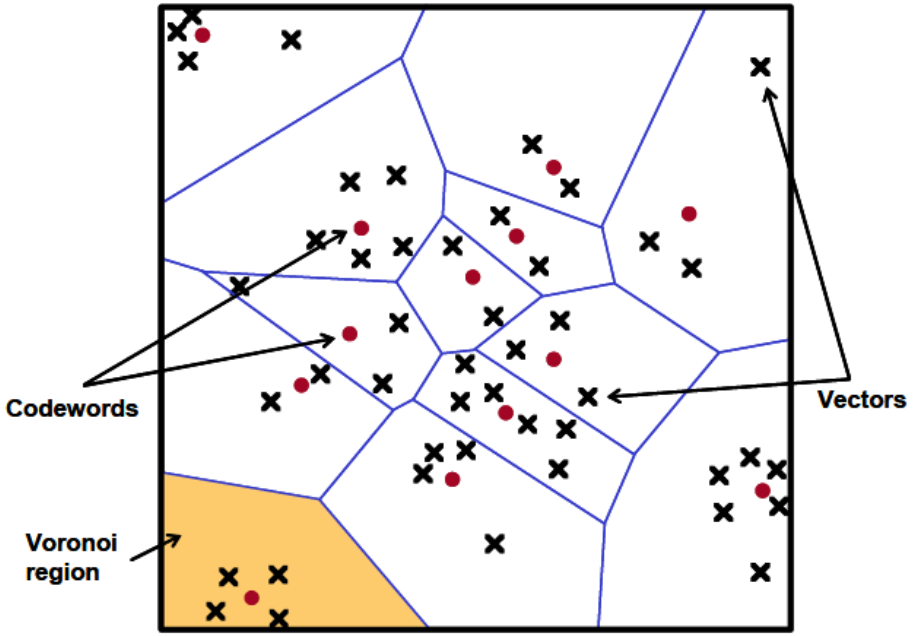
1. Define the number of clusters
2. Initialize clusters by
   - an arbitrary assignment of examples to clusters or
   - an arbitrary set of cluster centers (some examples used as centers)
3. Compute the sample mean of each cluster
4. Reassign each example to the cluster with the nearest mean
5. If the classification of all samples has not changed, stop, else go to step 3

- **It can be shown (Lecture 14) that k-means is a particular case of the EM algorithm for mixture models**

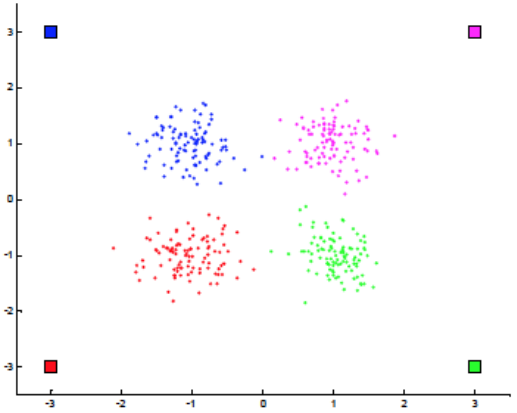
The k-means algorithm

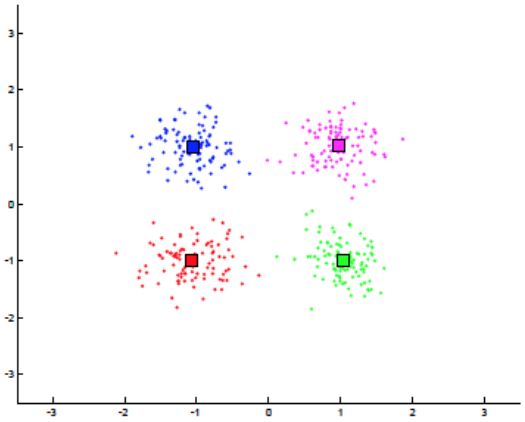
- **The k-means algorithm is widely used in the fields of signal processing and communication for Vector Quantization**
  - Unidimensional signal values are usually quantized into a number of levels (typically a power of 2 so the signal can be transmitted or stored in binary format)
  - The same idea can be extended for multiple channels
    - However, rather than quantizing each separate channel, we can obtain a more efficient signal coding if we quantize the overall multidimensional vector by finding a number of multidimensional prototypes (cluster centers)
  - The set of cluster centers is called a "codebook", and the problem of finding this codebook is normally solved using the k-means algorithm

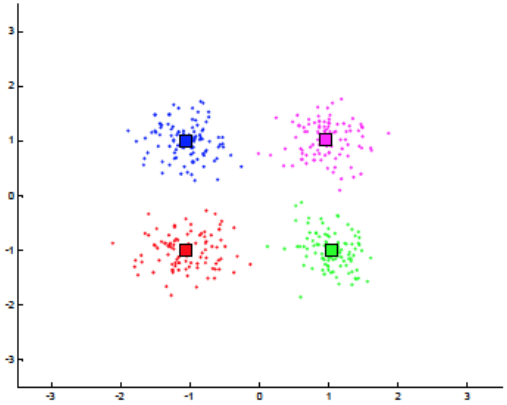




K-Means example







K-means algorithm

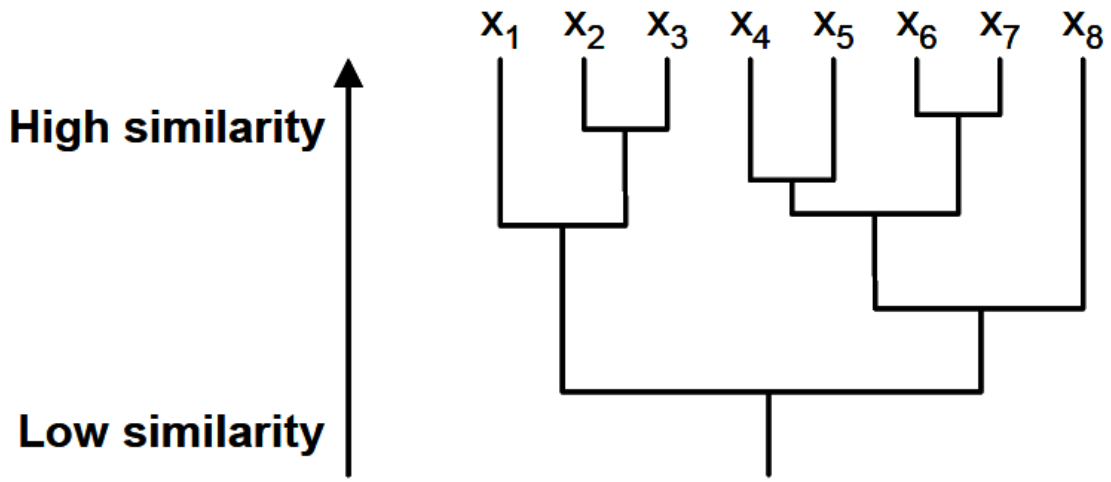
- **Properties:**
  - converges to centers minimizing the sum of squared center-point distances (still local optima)
  - The result is sensitive to the initial means' values
- **Advantages:**
  - Simplicity
  - Generality – can work for more than one distance measure
- **Drawbacks:**
  - Can perform poorly with overlapping regions
  - Lack of robustness to outliers
  - Good for attributes (features) with continuous values
    - Allows us to compute cluster means
    - k-medoid algorithm used for discrete data

Hierarchical clustering

- **k-means and ISODATA create disjoint clusters, resulting in a "flat" data representation**
  - However, sometimes it is desirable to obtain a hierarchical representation of data, with clusters and sub-clusters arranged in a tree-structured fashion
    - Hierarchical representations are commonly used in the sciences (i.e., biological taxonomy)
- **Hierarchical clustering methods can be grouped in two general classes**
  - Agglomerative
    - Also known as bottom-up or merging
    - Starting with N singleton clusters, successively merge clusters until one cluster is left
  - Divisive
    - Also known as top-down or splitting
    - Starting with a unique cluster, successively split the clusters until N singleton examples are left

Dendrograms

- **The preferred representation for hierarchical clusters is the dendrogram**
  - The dendrogram is a binary tree that shows the structure of the clusters
    - In addition to the binary tree, the dendrogram provides the similarity measure between clusters (the vertical axis)
  - An alternative representation is based on sets
    - $\{\{x_1, \{x_2, x_3\}\}, \{\{\{x_4, x_5\}, \{x_6, x_7\}\}, x_8\}\}$
    - However, unlike the dendrogram, sets cannot express quantitative information



Hierarchical clustering.

Uses an arbitrary similarity/dissimilarity measure.

Typical similarity measures *d(a,b)* :

Pure real-valued data-points:

- Euclidean, Manhattan, Minkowski distances

Pure binary values data:

- Hamming distance - Number of matching values

Pure categorical data:

- Number of matching values

Combination of real-valued and categorical attributes

- Weighted approaches

Hierarchical clustering

Approach:

- **Compute dissimilarity matrix for all pairs of points**
  - uses standard or other distance measures
- **Construct clusters greedily:**
  - **Agglomerative approach**
    - Merge pair of clusters in a bottom-up fashion, starting from singleton clusters
  - **Divisive approach:**
    - Splits clusters in top-down fashion, starting from one complete cluster
- **Stop the greedy construction** when some criterion is satisfied
  - E.g. fixed number of clusters

Agglomerative clustering (1)

- **Outline**
  - Define
    - N_C: Number of clusters
    - N_{EX}: Number of examples

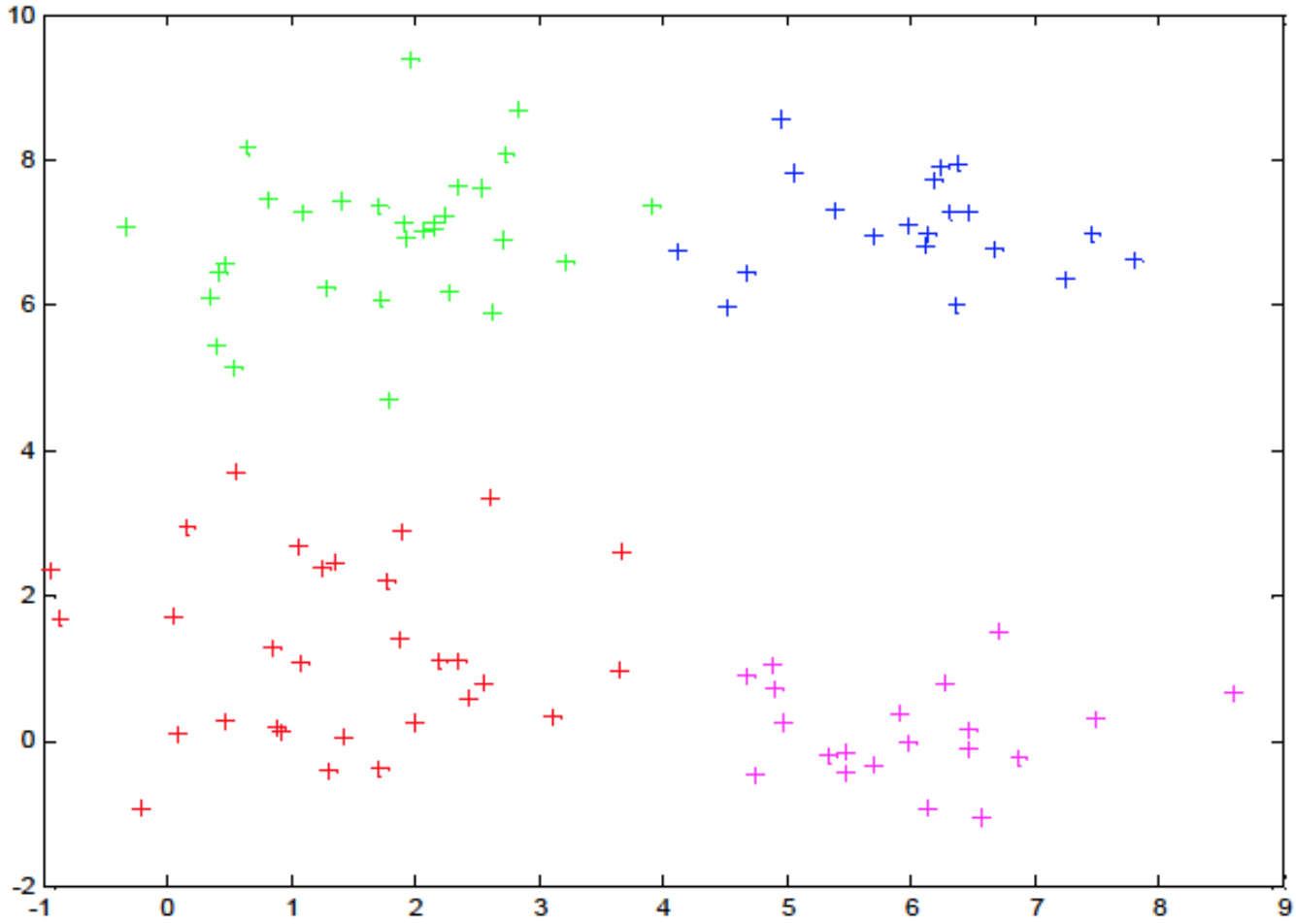
1. Start with N_{EX} singleton clusters
2. Find nearest clusters
3. Merge them
4. If N_C>1 go to 2

- **How to find the "nearest" pair of clusters**
  - Minimum distance $d_{min}(\omega_i,\omega_j)=\min_{\substack{x\in\omega_i\\y\in\omega_j}}\|x-y\|$
  - Maximum distance $d_{max}(\omega_i,\omega_j)=\max_{\substack{x\in\omega_i\\y\in\omega_j}}\|x-y\|$
  - Average distance $d_{avg}(\omega_i,\omega_j)=\frac{1}{N_iN_j}\sum_{x\in\omega_i}\sum_{y\in\omega_j}\|x-y\|$
  - Mean distance $d_{mean}(\omega_i,\omega_j)=\|\mu_i-\mu_j\|$

Agglomerative clustering (2)

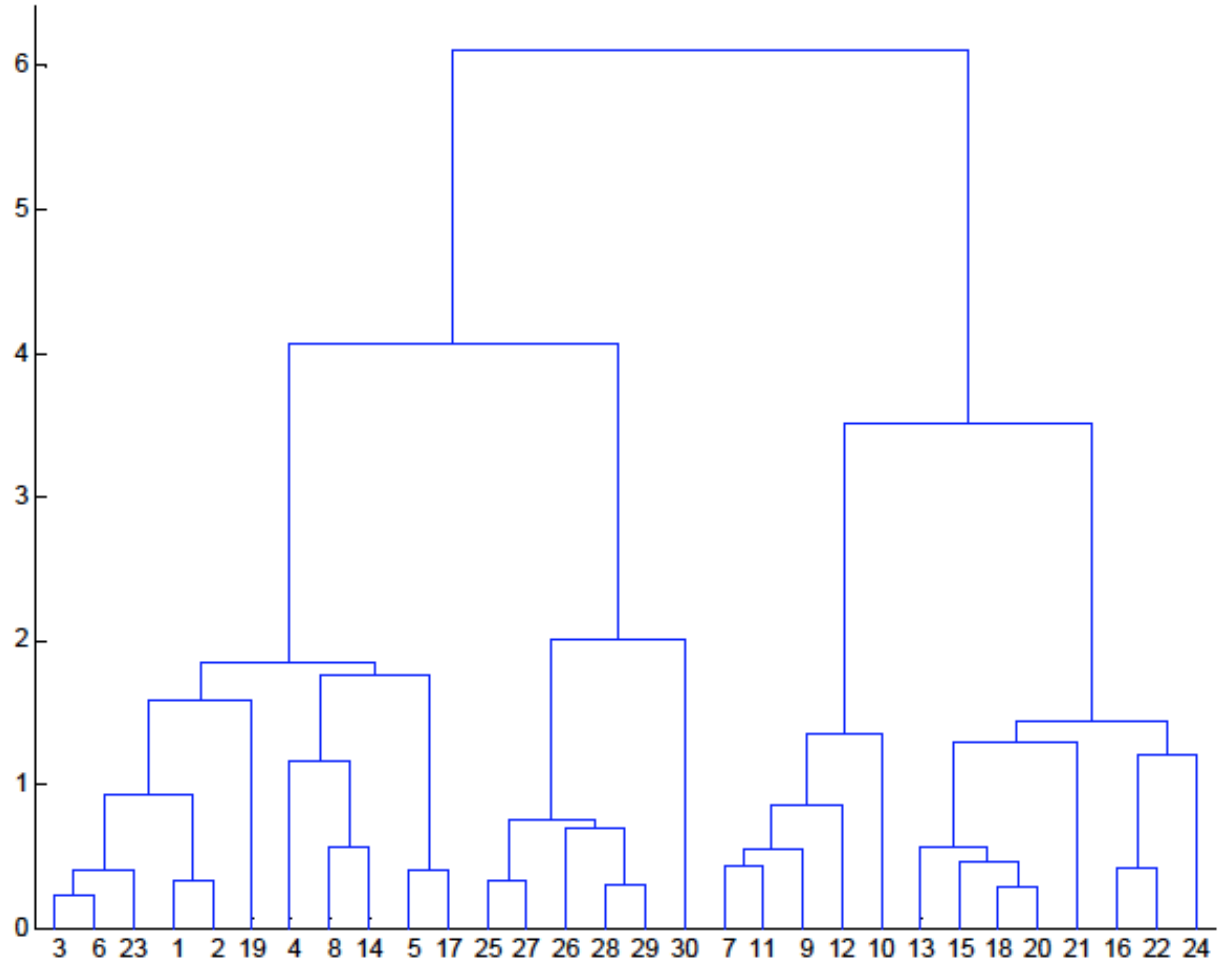
- **Minimum distance**
  - When d_{min} is used to measure distance between clusters, the algorithm is called the nearest-neighbor or single-linkage clustering algorithm
  - If the algorithm is allowed to run until only one cluster remains, the result is a minimum spanning tree (MST)
  - This algorithm favors elongated classes
- **Maximum distance**
  - When d_{max} is used to measure distance between clusters, the algorithm is called the farthest-neighbor or complete-linkage clustering algorithm
  - From a graph-theoretic point of view, each cluster constitutes a complete sub-graph
  - This algorithm favors compact classes
- **Average and mean distance**
  - The minimum and maximum distance are extremely sensitive to outliers since their measurement of between-cluster distance involves minima or maxima
  - The average and mean distance approaches are more robust to outliers
  - Of the two, the mean distance is computationally more attractive
    - Notice that the average distance approach involves the computation of N_iN_j distances for each pair of clusters

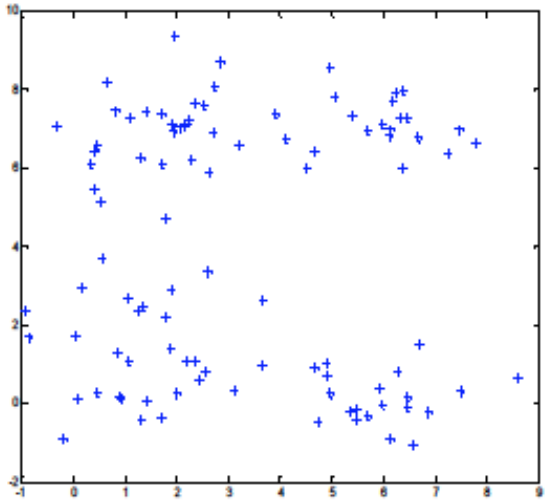
Hierarchical clustering example



Hierarchical clustering example

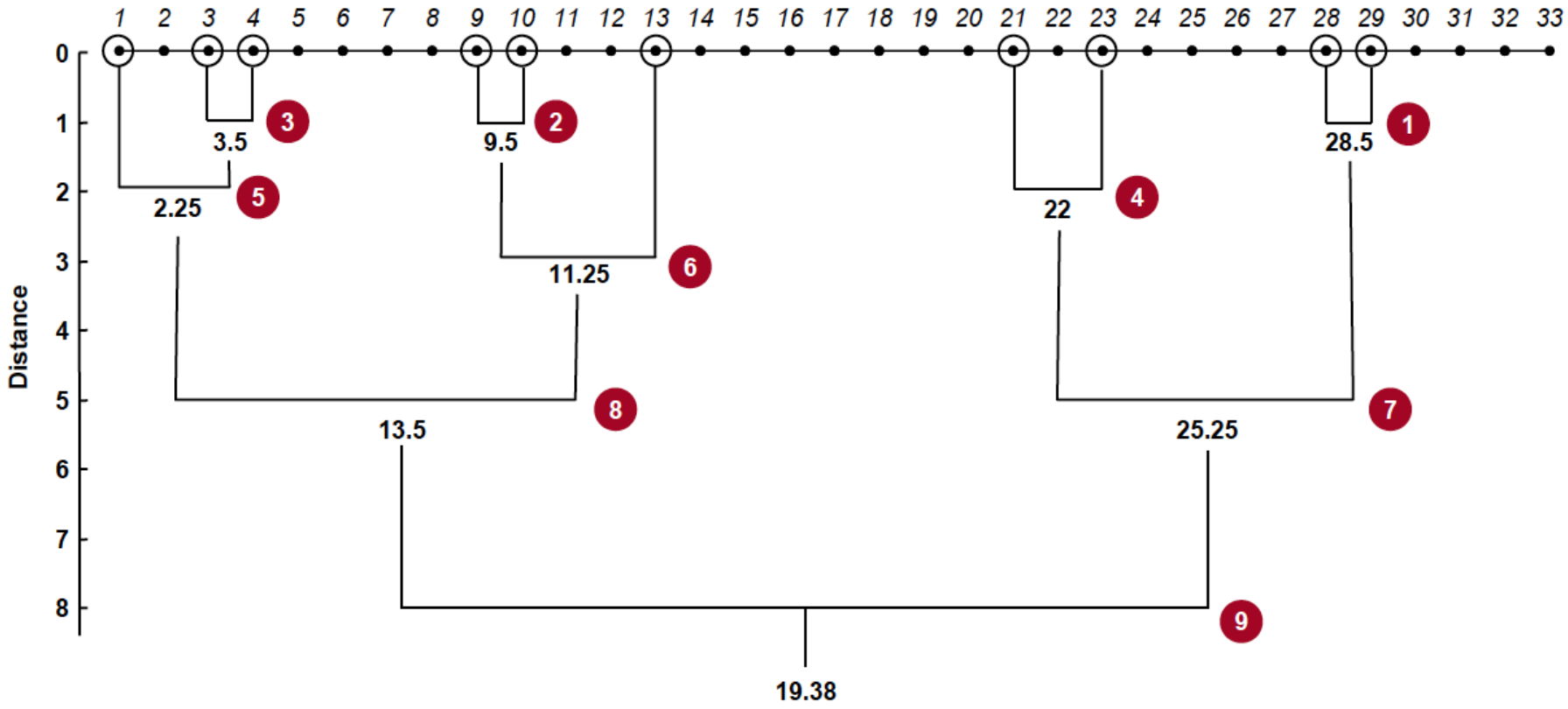
- **dendogram**





Agglomerative clustering example

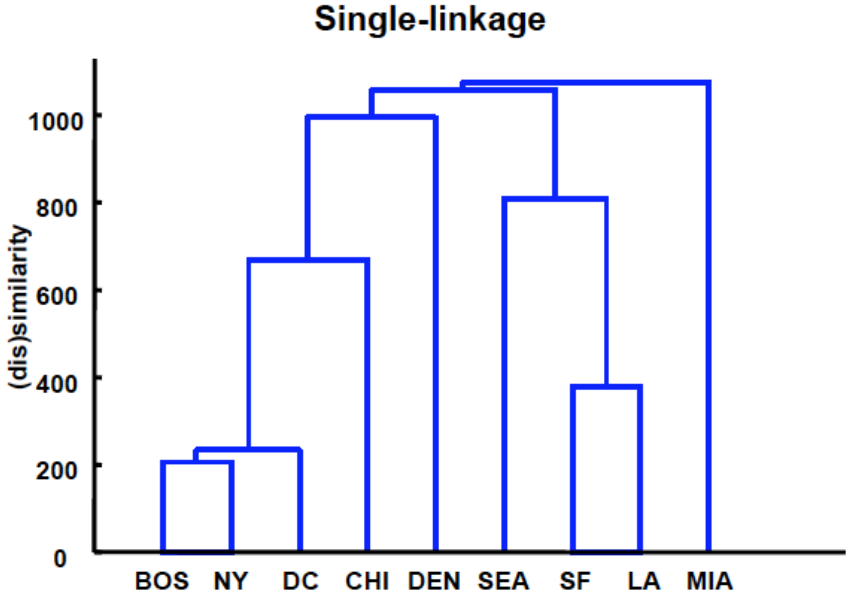
- **Perform agglomerative clustering on the following dataset using the single-linkage metric**
  - X = {1, 3, 4, 9, 10, 13, 21, 23, 28, 29}
  - In case of ties, always merge the pair of clusters with the largest mean
  - Indicate the order in which the merging operations occur

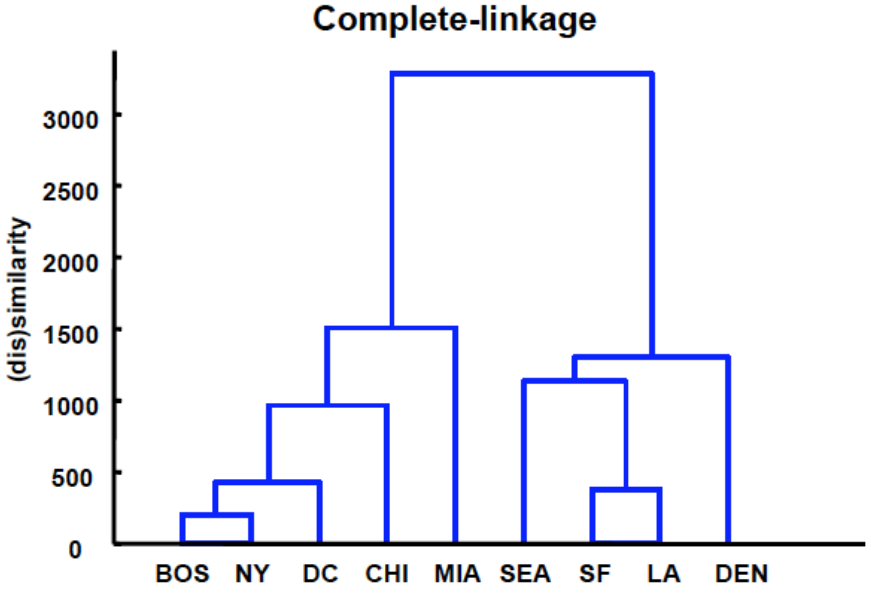


Agglomerative clustering, minimum Vs. maximum distance

- **Consider the problem of clustering nine major cities in the United States**

	BOS	NY	DC	MIA	CHI	SEA	SF	LA	DEN
BOS	0	206	429	1504	963	2976	3095	2979	1949
NY	206	0	233	1308	802	2815	2934	2786	1771
DC	429	233	0	1075	671	2684	2799	2631	1616
MIA	1504	1308	1075	0	1329	3273	3053	2687	2037
CHI	963	802	671	1329	0	2013	2142	2054	996
SEA	2976	2815	2684	3273	2013	0	808	1131	1307
SF	3095	2934	2799	3053	2142	808	0	379	1235
LA	2979	2786	2631	2687	2054	1131	379	0	1059
DEN	1949	1771	1616	2037	996	1307	1235	1059	0





Divisive clustering

- **Outline**
  - Define
    - N_C: Number of clusters
    - N_{EX}: Number of examples

1. Start with one large cluster
2. Find "worst" cluster
3. Split it
4. If $N_C < N_{EX}$ go to 2

- **How to choose the "worst" cluster**
  - Largest number of examples
  - Largest variance
  - Largest sum-squared-error
  - ...
- **How to split clusters**
  - Mean-median in one feature direction
  - Perpendicular to the direction of largest variance
  - …
- **The computations required by divisive clustering are more intensive than for agglomerative clustering methods**
  - For this reason, agglomerative approaches are more popular

Hierarchical clustering

- **Advantage:**
  - Smaller computational cost; avoids scanning all possible clusterings
- **Disadvantage:**
  - Greedy choice fixes the order in which clusters are merged; cannot be repaired
- **Partial solution:**
    - combine hierarchical clustering with iterative algorithms like k-means