LECTURE 15: Statistical clustering

- **Similarity measures**
- **Criterion functions**
- **Cluster validity**
- **Flat clustering algorithms**
  - k-means
  - ISODATA
- **Hierarchical clustering algorithms**
  - Divisive
  - Agglomerative

Non-parametric unsupervised learning

- **In lecture 14 we introduced the concept of unsupervised learning**
  - A collection of pattern recognition methods that "learn without a teacher"
  - Two types of clustering methods were mentioned: parametric and non-parametric
- **Parametric unsupervised learning**
  - Equivalent to density estimation with a mixture of (Gaussian) components
  - Through the use of EM, the identity of the component that originated each data point was treated as a missing feature
- **Non-parametric unsupervised learning**
  - No density functions are considered in these methods
  - Instead, we are concerned with finding natural groupings (clusters) in a dataset
- **Non-parametric clustering involves three steps**
  - Defining a measure of (dis)similarity between examples
  - Defining a criterion function for clustering
  - Defining an algorithm to minimize (or maximize) the criterion function

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**

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

  - Parameter *p* controls the weight placed on any dimension dissimilarity, whereas parameter *r* controls the distance growth of patterns that are further apart
  - Notice that the definition of norm must be relaxed, allowing a power factor for |a|

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)**

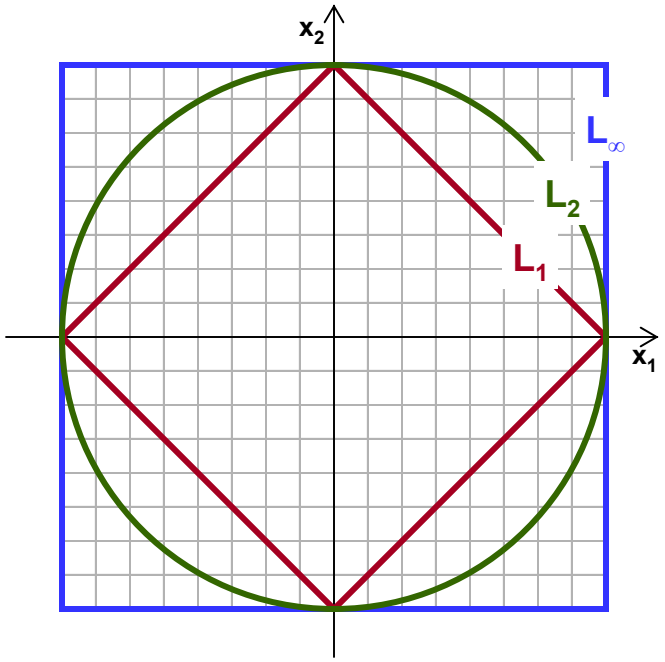
$$\|x - y\|_{c\text{-}b} = \sum_{k=1}^{D} |x_k - y_k|$$

    - 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 $N\to\infty$

From [Webb, 1999]

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)

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

Criterion function for clustering

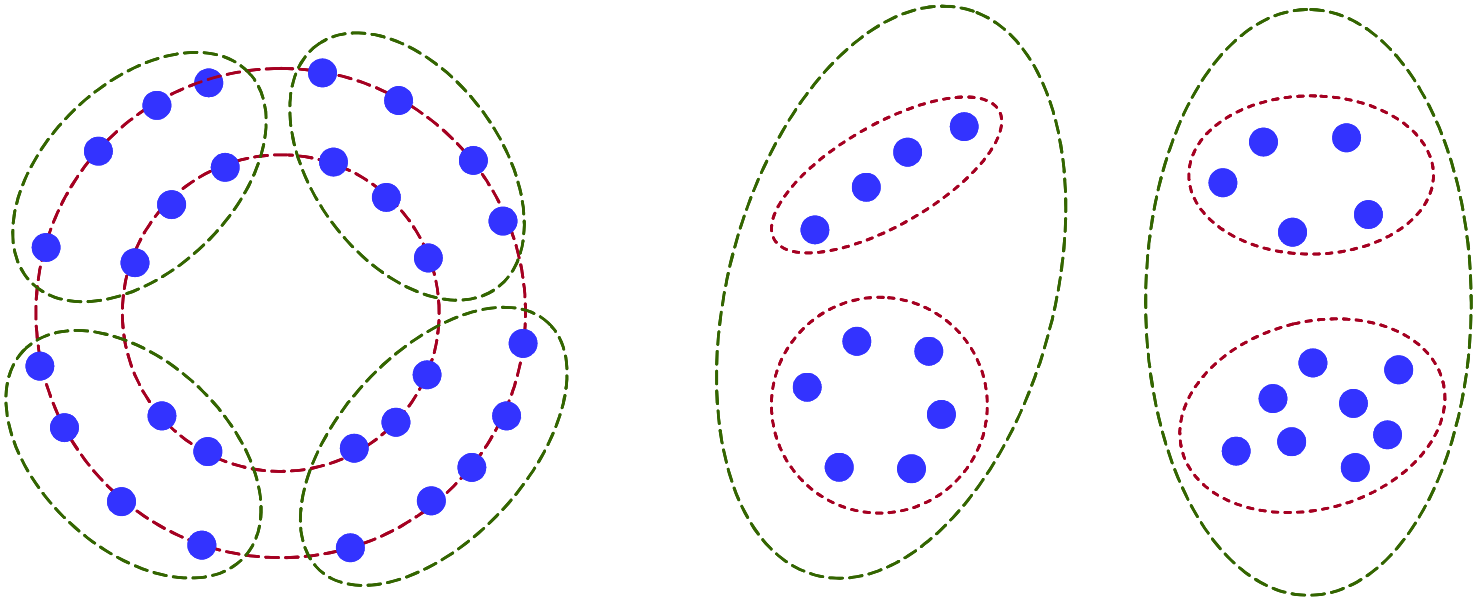
- **Once a (dis)similarity measure has been determined, we need to define a criterion function to be optimized**
  - The most widely used criterion function for clustering is the sum-of-square-error

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

    - 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]

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]**

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**
- **We will consider two groups of iterative methods**
  - Flat clustering algorithms
    - These algorithms produce a set of disjoint clusters
    - Two algorithms are widely used: k-means and ISODATA
  - Hierarchical clustering algorithms:
    - The result is a hierarchy of nested clusters
    - These algorithms can be broadly divided into agglomerative and divisive approaches

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**

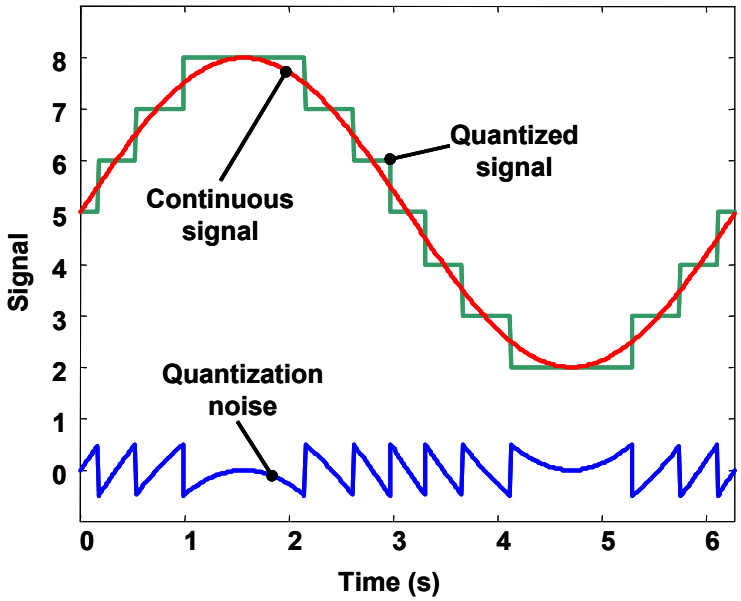
$$J_{MSE} = \sum_{i=1}^{C} \sum_{x \approx \omega_i} |x - \mu_i|^2 \qquad \text{where } \mu_i = \frac{1}{N_i} \sum_{x \approx \omega_i} x$$

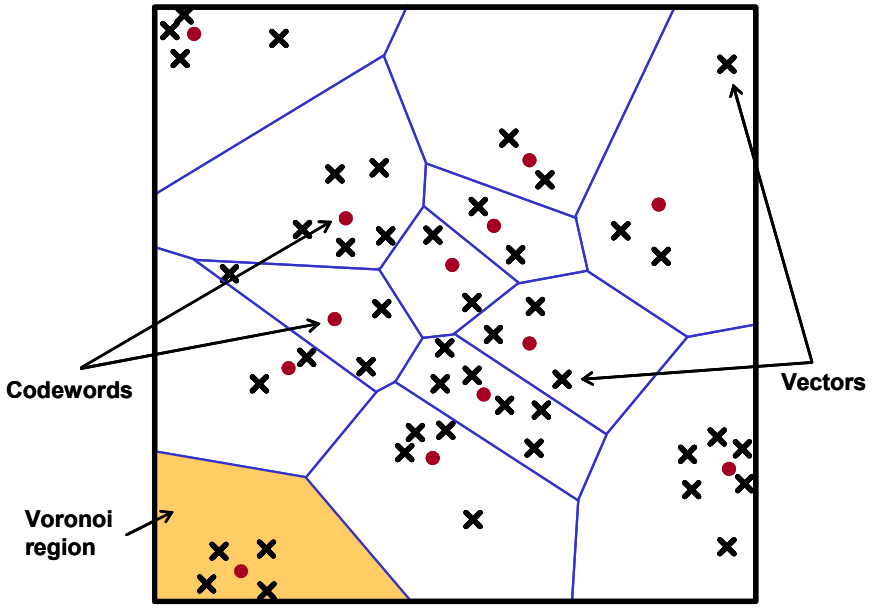
1. Define the number of clusters
2. Initialize clusters by
   - an arbitrary assignment of examples to clusters or
   - an arbitrary set of cluster centers (some examples used as centers)
3. Compute the sample mean of each cluster
4. Reassign each example to the cluster with the nearest mean
5. If the 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





ISODATA (1)

- **ISODATA, which stands for Iterative Self-Organizing Data Analysis Technique (Algorithm) is an extension to the k-means algorithm with some heuristics to automatically select the number of clusters**
- **ISODATA requires the user to select a number of parameters**
  - N_{MIN_EX} minimum number of examples per cluster
  - N_D desired (approximate) number of clusters
  - σ_S^2 maximum spread parameter for splitting
  - D_{MERGE} maximum distance separation for merging
  - N_{MERGE} maximum number of clusters that can be merged
- **The algorithm works in an iterative fashion**
  (1) Perform k-means clustering
  (2) Split any clusters whose samples are sufficiently dissimilar
  (3) Merge any two clusters sufficiently close
  (4) Go to (1)
- **A more detailed description of the algorithm follows**

ISODATA (2)

1. Select an initial number of clusters N_C and use the first N_C examples as cluster centers μ_k, k=1..N_C
2. Assign each example to the closest cluster
   a. Exit the algorithm if the classification of examples has not changed
3. Eliminate clusters that contain less than N_{MIN_EX} examples and
   a. Assign their examples to the remaining clusters based on minimum distance
   b. Decrease N_C accordingly
4. For each cluster k,
   a. Compute the center μ_k as the sample mean of all the examples assigned to that cluster
   b. Compute the average distance between examples and cluster centers $d_{AVG} = \frac{1}{N}\sum_{k=1}^{N_C} N_k d_k$ and $d_k = \frac{1}{N_k}\sum_{x \approx \omega_k} |x - \mu_k|$
   c. Compute the variance of each axis and find the axis n* with maximum variance $\sigma_k^2(n^*)$
6. For each cluster k with $\sigma_k^2(n^*) > \sigma_S^2$, if $\{d_k > d_{AVG}$ and $N_k > 2N_{MIN\_EX}+1\}$ or $\{N_C < N_D/2\}$
   a. Split that cluster into two clusters where the two centers μ_{k1} and μ_{k2} differ only in the coordinate n*
      i. $\mu_{k1}(n^*) = \mu_k(n^*) + \varepsilon\sigma_k(n^*)$ (all other coordinates remain the same, $0<\varepsilon<1$)
      ii. $\mu_{k2}(n^*) = \mu_k(n^*) - \varepsilon\sigma_k(n^*)$ (all other coordinates remain the same, $0<\varepsilon<1$)
   b. Increment N_C accordingly
   c. Reassign the cluster's examples to one of the two new clusters based on minimum distance to cluster centers
7. If $N_C > 2N_D$ then
   a. Compute all distances $D_{ij}=d(\mu_i, \mu_j)$
   b. Sort D_{ij} in decreasing order
   b. For each pair of clusters sorted by D_{ij}, if (1) neither cluster has been already merged, (2) the distance D_{ij} satisfies $D_{ij} < D_{MERGE}$ and (3) not more than N_{MERGE} pairs of clusters have been merged in this loop, then
      i. Merge i^{th} and j^{th} clusters
      ii. Compute the cluster center $\mu' = \frac{N_i\mu_i + N_j\mu_j}{N_i + N_j}$
      iii. Decrement N_C accordingly
8. Go to step 1

From [Therrien, 1989]

ISODATA (3)

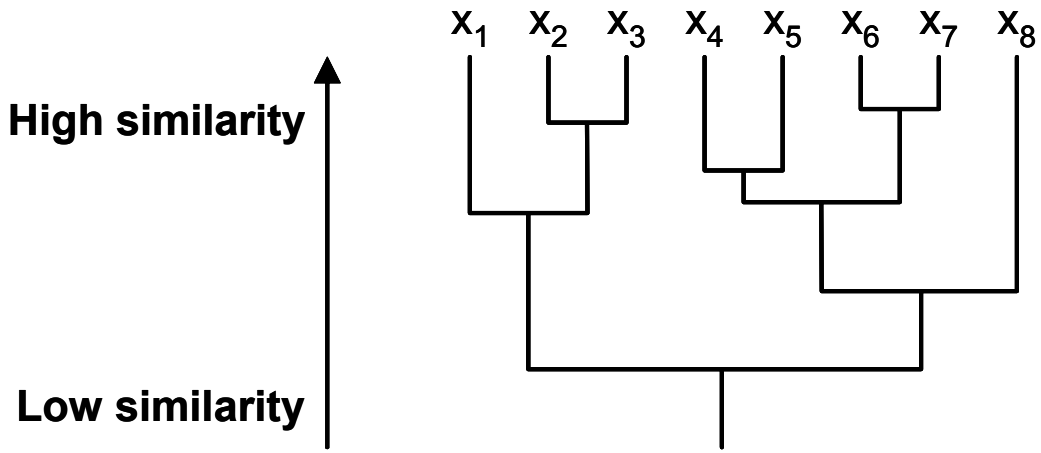
- **ISODATA has been shown to be an extremely powerful heuristic**
- **Some of its advantages are**
  - Self-organizing capabilities
  - Flexibility in eliminating clusters that have very few examples
  - Ability to divide clusters that are too dissimilar
  - Ability to merge clusters that are sufficiently similar
- **However, it suffers from the following limitations**
  - Data must be linearly separable (long narrow or curved clusters are not handled properly)
  - It is difficult to know a priori the “optimal” parameters
  - Performance is highly dependent on these parameters
  - For large datasets and large number of clusters, ISODATA is less efficient than other linear methods
  - Convergence is unknown, although it appears to work well for non-overlapping clusters
- **In practice, ISODATA is run multiple times with different values of the parameters and the clustering with minimum sum-squared error is selected**

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



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

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

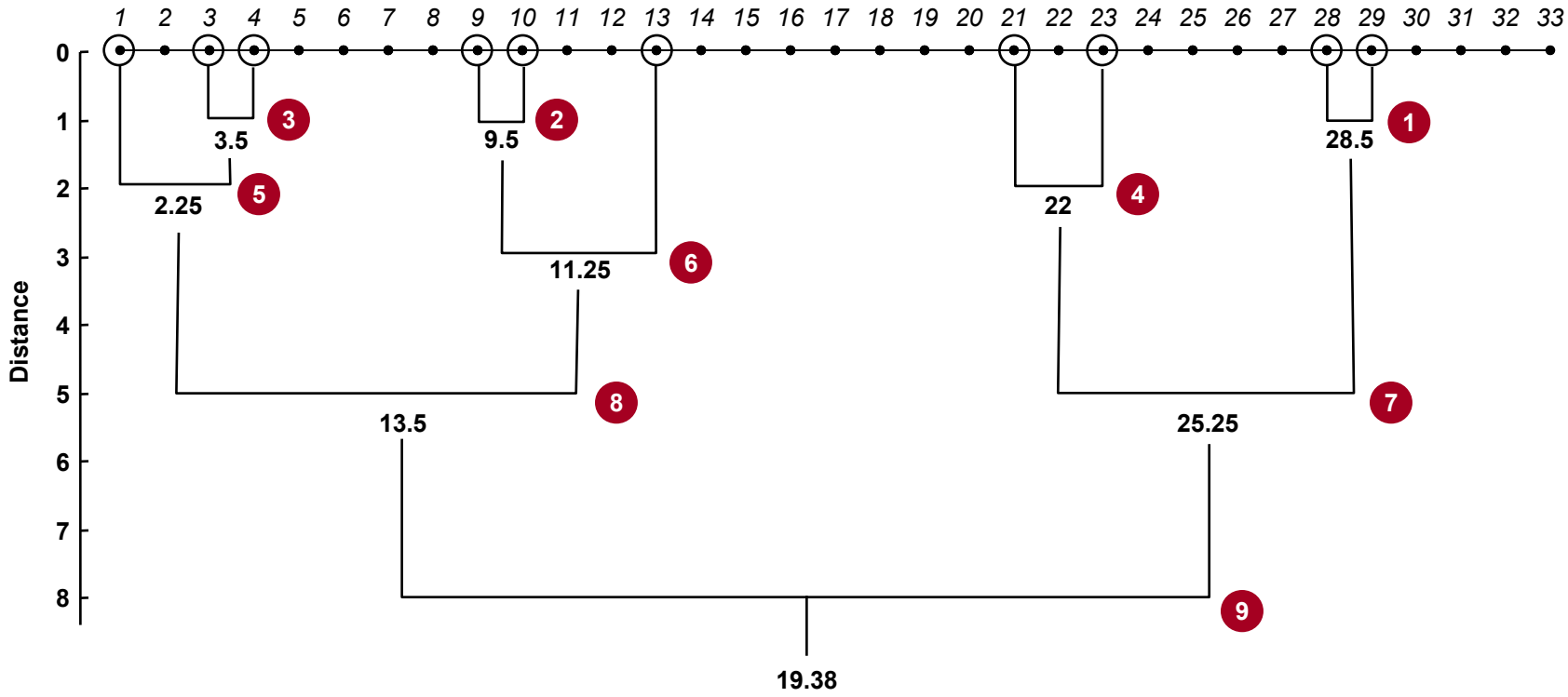
- **How to find the "nearest" pair of clusters**
  - Minimum distance $d_{min}(\omega_i,\omega_j)=\min_{\substack{x\in\omega_i\\y\in\omega_j}}\|x-y\|$
  - Maximum distance $d_{max}(\omega_i,\omega_j)=\max_{\substack{x\in\omega_i\\y\in\omega_j}}\|x-y\|$
  - Average distance $d_{avg}(\omega_i,\omega_j)=\frac{1}{N_iN_j}\sum_{x\in\omega_i}\sum_{y\in\omega_j}\|x-y\|$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

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

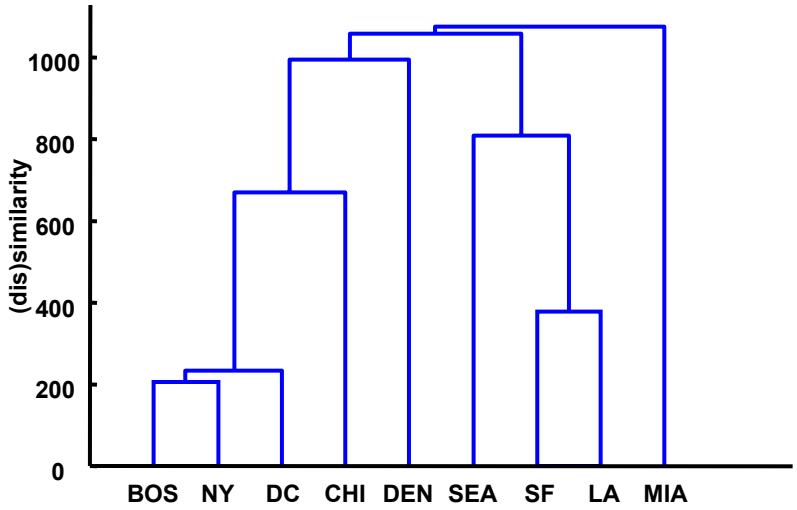


Agglomerative clustering, minimum Vs. maximum distance

- **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

Single-linkage



Complete-linkage