CSCI 556 Data Analysis & Visualization

Unsupervised Learning

Instructor: Dr. Jinoh Kim

Unsupervised learning

- Refers to statistical methods that extract meaning from data without training a model on labeled data
  - Recall that the goal of classification is to build a model (set of rules) to predict a response from a set of predictor variables (i.e., classification is based on supervised learning)
  - Unsupervised learning also constructs a model of the data, but does not distinguish between a response variable and predictor variables.
- Can be used:
  - To create a predictive rule in the absence of a labeled response
  - To reduce the dimension of data
  - To perform exploratory data analysis

Principal components analysis

- PCA is a technique to discover the way in which numeric variables covary (vary together)
- PCA combines multiple numeric predictor variables into a smaller set of variables, which are weighted linear combinations of the original set
- Principal components: smaller set of variables that explain most of the variability of the full set of variables, reducing the dimension of the data

Covariance

- Covariance measures the relationship between two variables
- Covariance of two variables of *x* and *z* with the mean for each variable $\bar{x}$ and $\bar{z}$ is defined:

$$s_{x,z} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(z_i - \bar{z})}{n - 1}$$

where *n* is the number of records

- Positive values indicate a positive relationship and negative values indicate a negative relationship
- Covariance matrix (∑) consists of the individual variable variances s_x^2 and s_z^2 and the covariances between variable pairs

$$\hat{\Sigma} = \begin{bmatrix} s_x^2 & s_{x,z} \\ s_{x,z} & s_z^2 \end{bmatrix}$$

Covariance vs. correlation

- As with the correlation coefficient, positive values indicate a positive relationship and negative values indicate a negative relationship
- Correlation is constrained to be between –1 and 1, whereas covariance is on the same scale as the variables
- Covariance:

$$s_{x,z} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(z_i - \bar{z})}{n-1}$$

- Correlation: $$r = \frac{\sum_{i=1}^{n}(xi - \bar{x})(z_i - \bar{z})}{(n-1)s_x s_z}$$

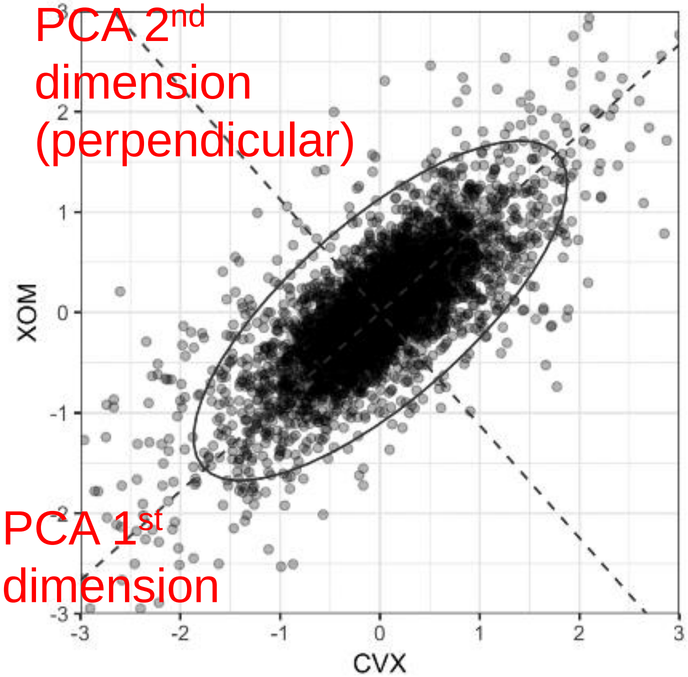
- Thus, $r = \frac{s_{x,z}}{s_x s_y}$

Simple example: PCA

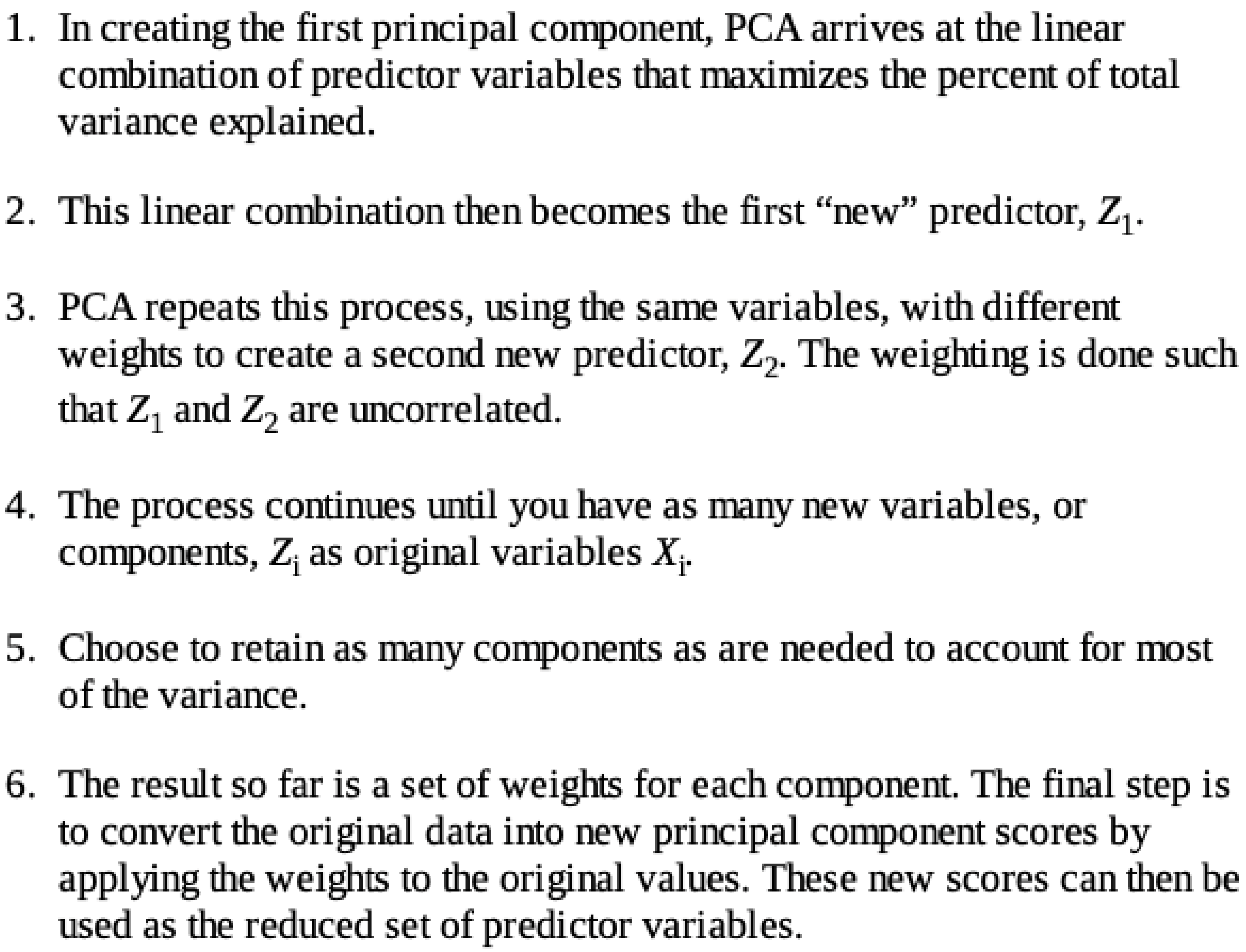
- For two variables, X_1 and X_2, there are two principal components Z_i (i=1 or 2):

$$Z_i = w_{i,1}X_1 + w_{i,2}X_2$$

- The weights ($w_{i,1}$, $w_{i,2}$) are also known as component loadings
- The first principal component Z_1 is the linear combination that best explains the total variation
- The second principal component Z_2 is the linear combination that explains the remaining variation and Z_2 is perpendicular to Z_1 (i.e., unrelated)



Computing principal components

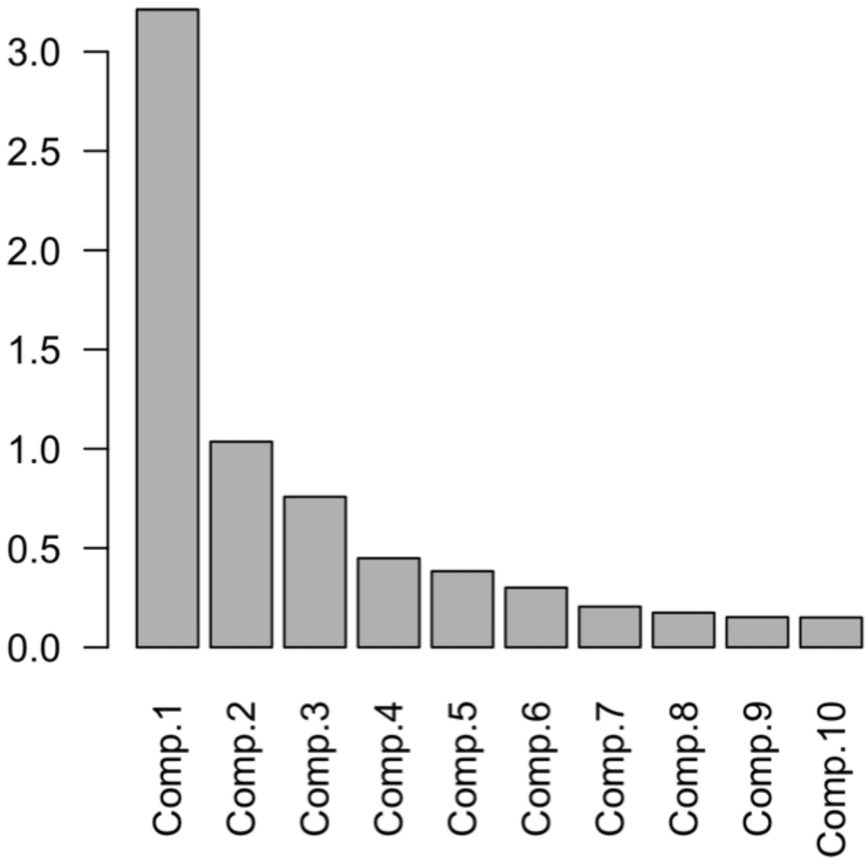


1. In creating the first principal component, PCA arrives at the linear combination of predictor variables that maximizes the percent of total variance explained.
2. This linear combination then becomes the first "new" predictor, Z_1.
3. PCA repeats this process, using the same variables, with different weights to create a second new predictor, Z_2. The weighting is done such that Z_1 and Z_2 are uncorrelated.
4. The process continues until you have as many new variables, or components, Z_i as original variables X_i.
5. Choose to retain as many components as are needed to account for most of the variance.
6. The result so far is a set of weights for each component. The final step is to convert the original data into new principal component scores by applying the weights to the original values. These new scores can then be used as the reduced set of predictor variables.

Screeplot

- Visualizes the relative importance of principal components

- Example of screeplot showing 10 principal components with their importance



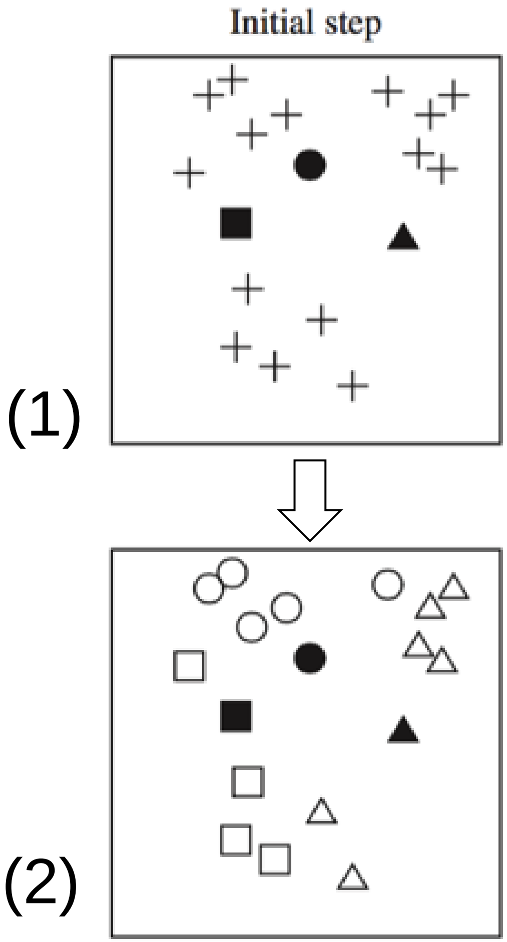
Clustering

- A technique to divide data into different groups, where the records in each group are similar to one another
- Goal: to identify significant and meaningful groups of data
- Cluster: A group of records that are similar
- Cluster mean: The vector of variable means for the records in a cluster.

K-means clustering

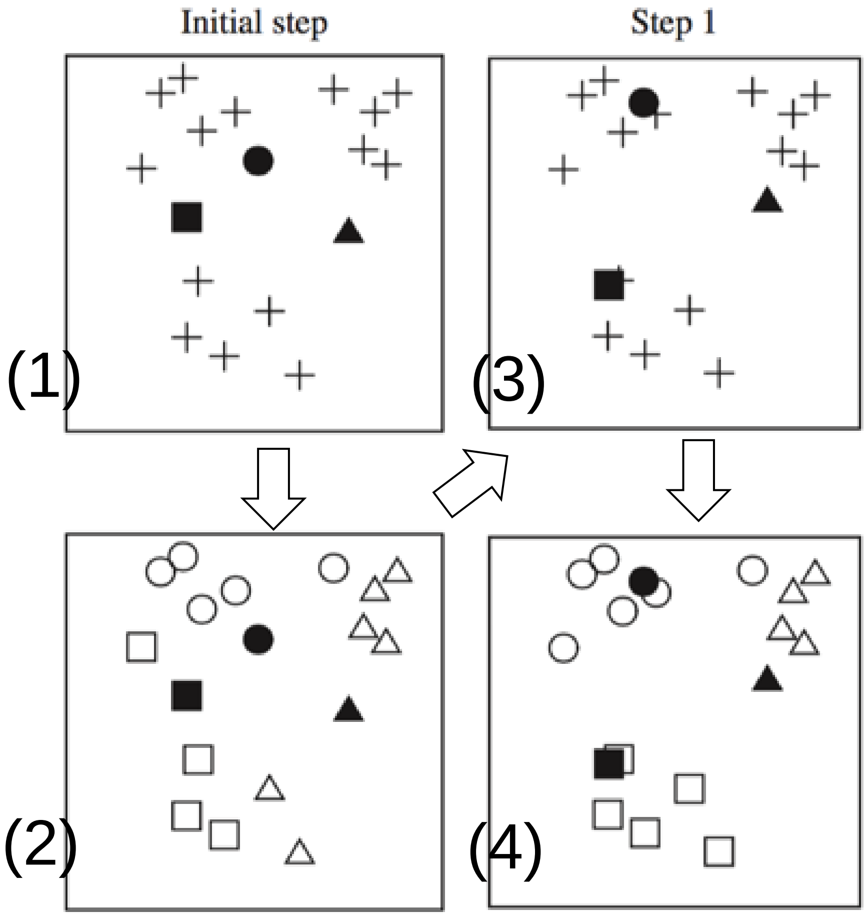
- First clustering method widely used, owing its popularity to the relative simplicity of the algorithm and its ability to scale to large data sets
- *K*-means divides the data into *K* clusters by minimizing the sum of the squared distances of each record to the mean of its assigned cluster, also known as within-cluster sum of squares (SS)
- *K*-means does not ensure the clusters will have the same size, but finds the clusters that are the best separated

K-means algorithm: example



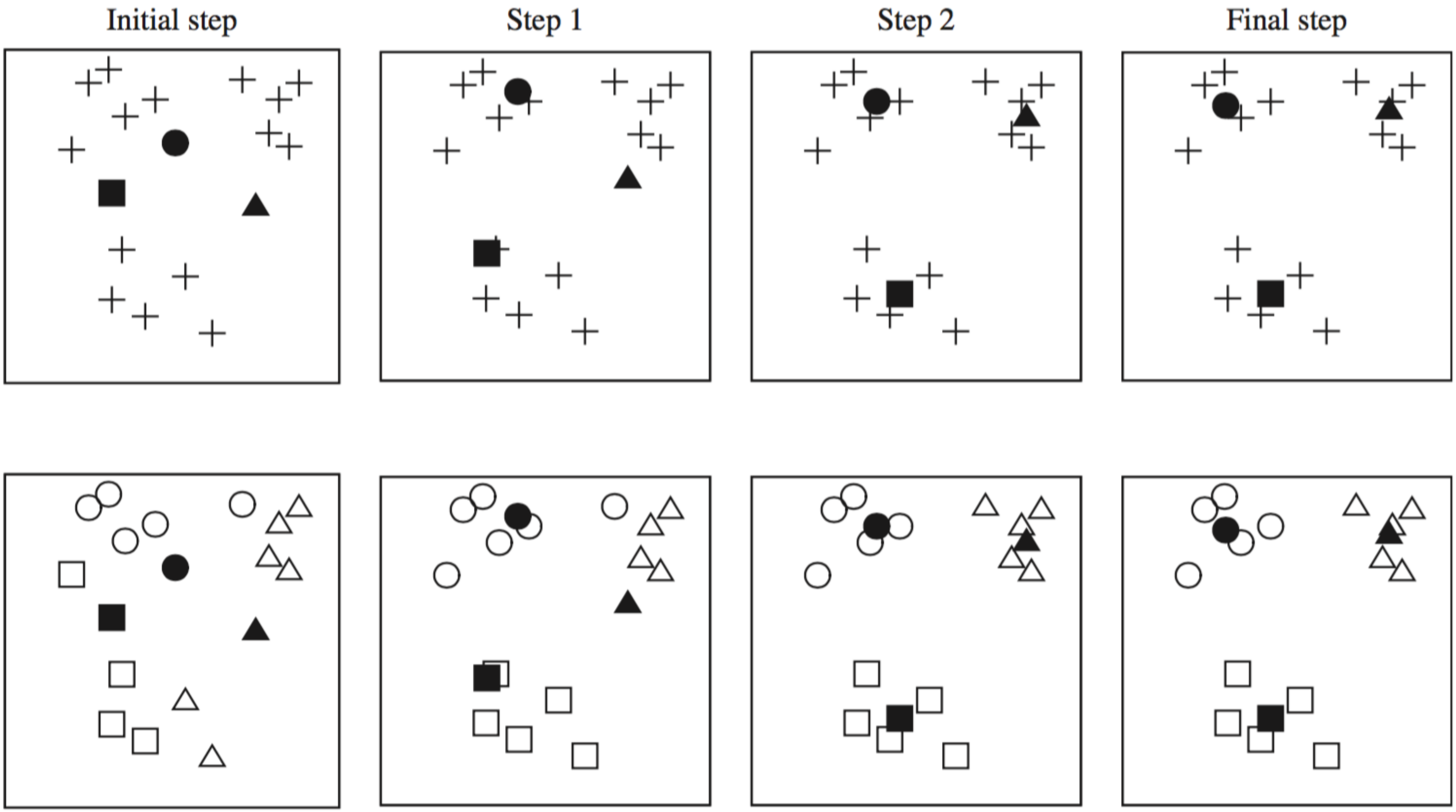
- Data instances ('+')
- Number of clusters (k) = 3
- Seed points (centers): square, circle, triangle
- Assign each instance to its closest cluster center based on Euclidean distance

K-means algorithm: example (2)



- At every step:
  - Recompute cluster centers by computing the average (aka *centroid*) of the instances pertaining to each cluster
  - Assign each instance to its closest cluster center based on Euclidean distance
- Stopping condition:
  - Stop when assignment of instances to cluster centers has not changed

K-means algorithm: example (3)



Initial step

Step 1

Step 2

Final step

Simple example: K-means clustering

- Consider a data set with *n* records and just two variables, *x* and *y*
- Suppose we want to split the data into K=4 clusters
- For cluster *k* with n_k elements, the center of the cluster $(\overline{x}_k, \overline{y}_k)$ is simply the mean of the points in the cluster:

$$\overline{x}_k = \frac{1}{n_k} \sum_{i \in \text{Cluster } k} x_i$$

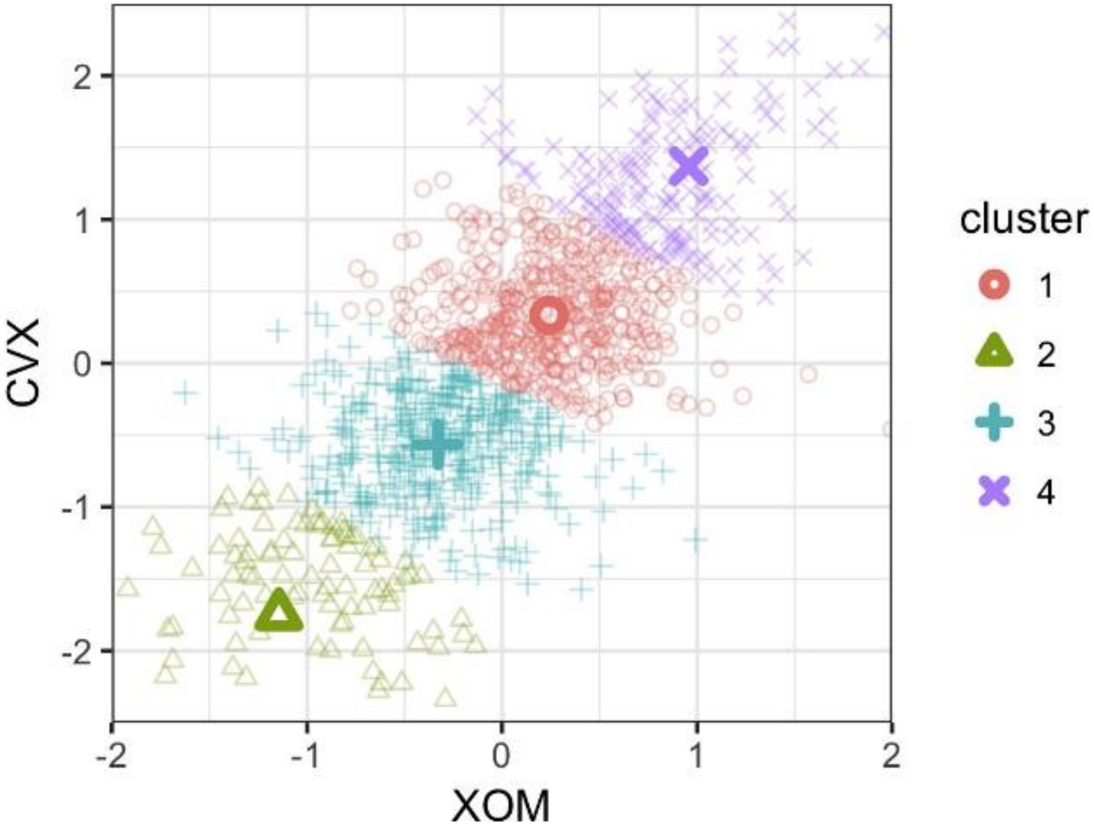
$$\overline{y}_k = \frac{1}{n_k} \sum_{i \in \text{Cluster } k} y_i$$

- The sum of squares within a cluster is given by:

$$\text{SS}_k = \sum_{i \in \text{Cluster } k} (x_i - \overline{x}_k)^2 + (y_i - \overline{y}_k)^2$$

- K-means finds the assignment of records that minimizes within-cluster sum of squares across all four clusters, i.e., $SS_1 + SS_2 + SS_3 + SS_4$

K-means clustering example



- Shows 4 different clusters with their center position

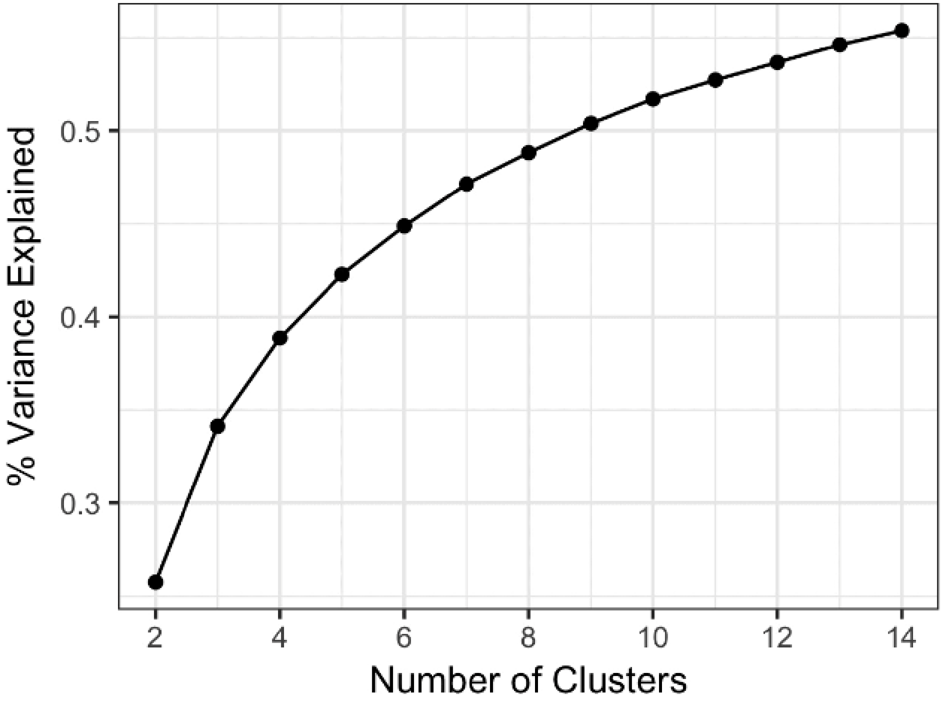
K-means algorithm

- With given k and an initial set of cluster means, iterate the following steps:
  1. Assign each record to the nearest cluster mean as measured by squared distance
  2. Compute the new cluster means based on the assignment of records
  3. Stops if the assignment of records to clusters does not change
- Euclidean distance is used to locate the nearest cluster

Selecting the number of clusters

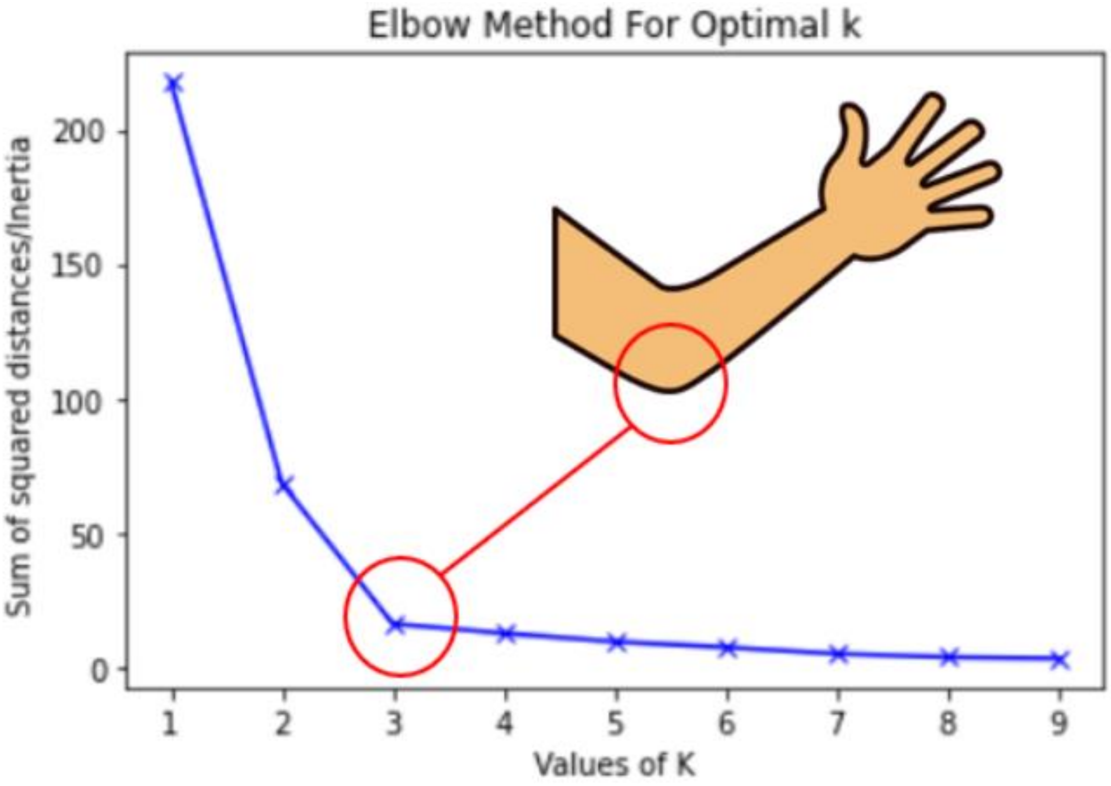
- Elbow method is a common approach that identifies when the set of clusters explains "most" of the variance in the data
- Elbow: the point where the cumulative variance explained flattens out after rising steeply

- Where is the elbow in this example?
- There is no obvious candidate, since the incremental increase in variance explained drops gradually.



Selecting the number of clusters

- Can use elbow method with the within-cluster sum of squares (across all clusters)



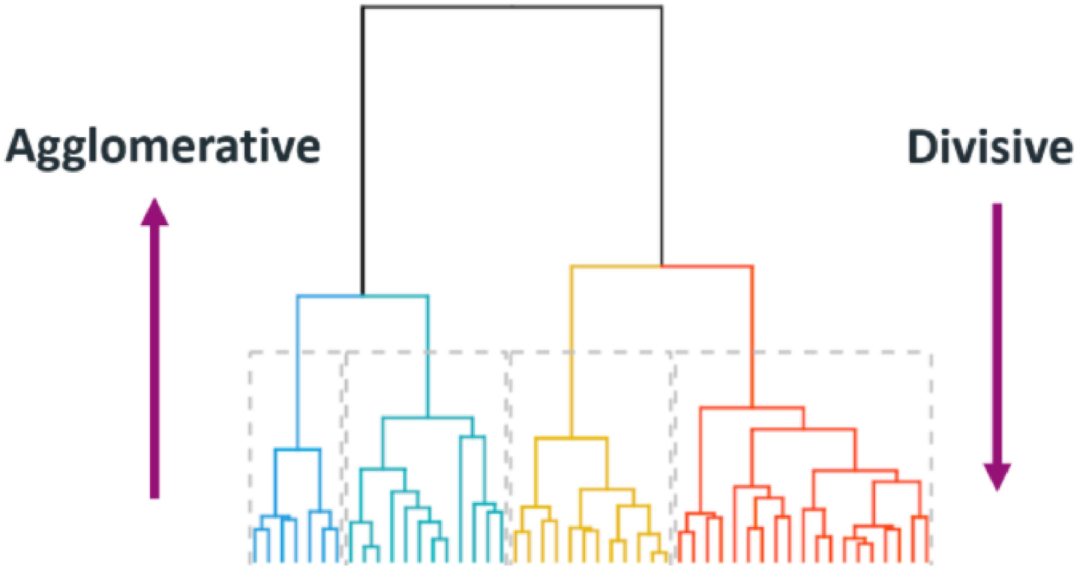
Source: https://www.analyticsvidhya.com/blog/2021/05/k-mean-getting-the-optimal-number-of-clusters/

Hierarchical clustering

- Builds a hierarchy of clusters
- Not necessary to prespecify the number of clusters
- Provides an intuitive graphical display, leading to easier interpretation of the clusters
- But does not scale well to large data sets with intensive computing requirements

Agglomerative clustering

- Standard hierarchical clustering performs clustering in a bottom-up manner; it performs *agglomerative* clustering:
  - First, make each instance in the dataset into a trivial mini-cluster
  - Then, find the two closest clusters and merge them; repeat
  - Clustering stops when all clusters have been merged into a single cluster



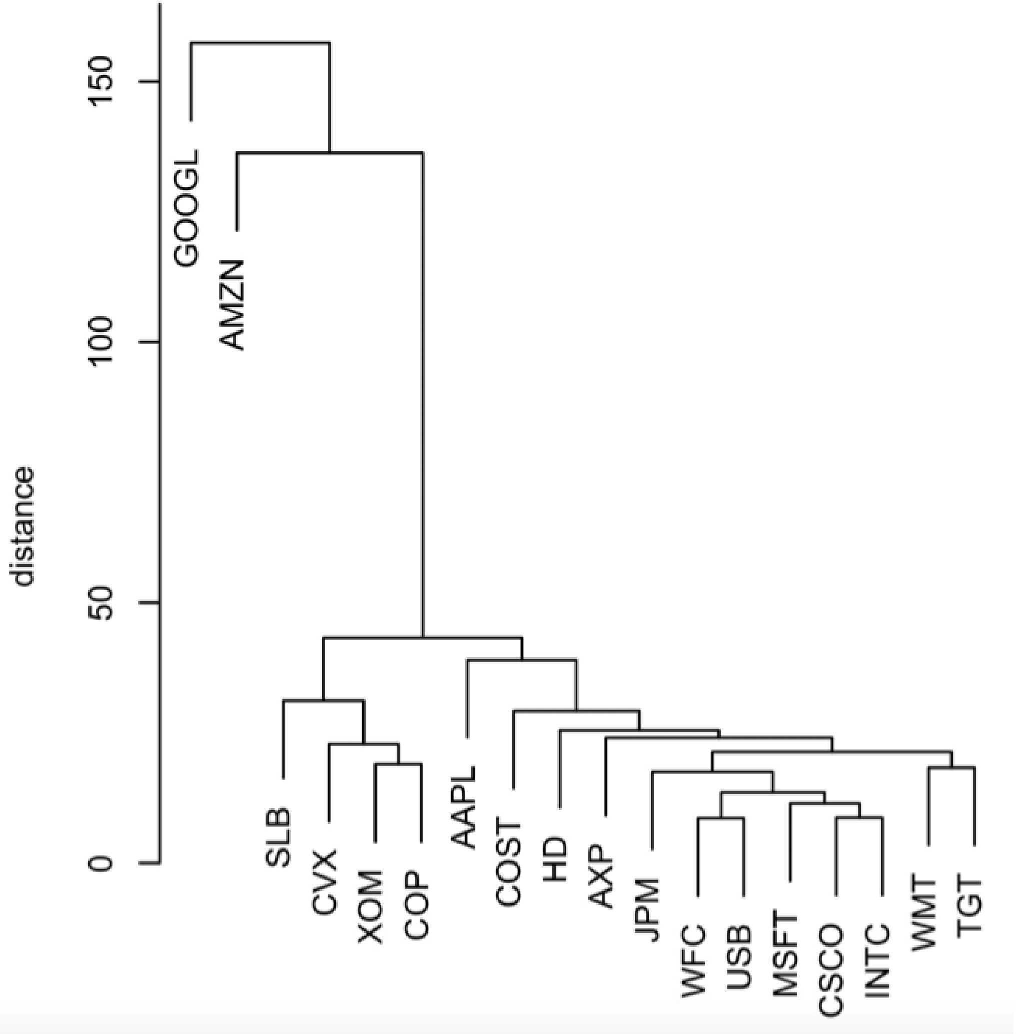
Source: https://www.mygreatlearning.com/blog/hierarchical-clustering/

Simple example: hierarchical clustering

- Hierarchical clustering works on a data set with *n* records and *p* variables and is based on two basic building blocks:
  - $d_{i,j}$: Distance metric to measure the distance between two records *i* and *j* (e.g., Euclidean distance)
  - $D_{A,B}$: Dissimilarity metric to measure the difference between two clusters A and B based on the distances between the members of each cluster.
- Hierarchical clustering starts by setting each record as its own cluster and iterates to combine the least dissimilar clusters

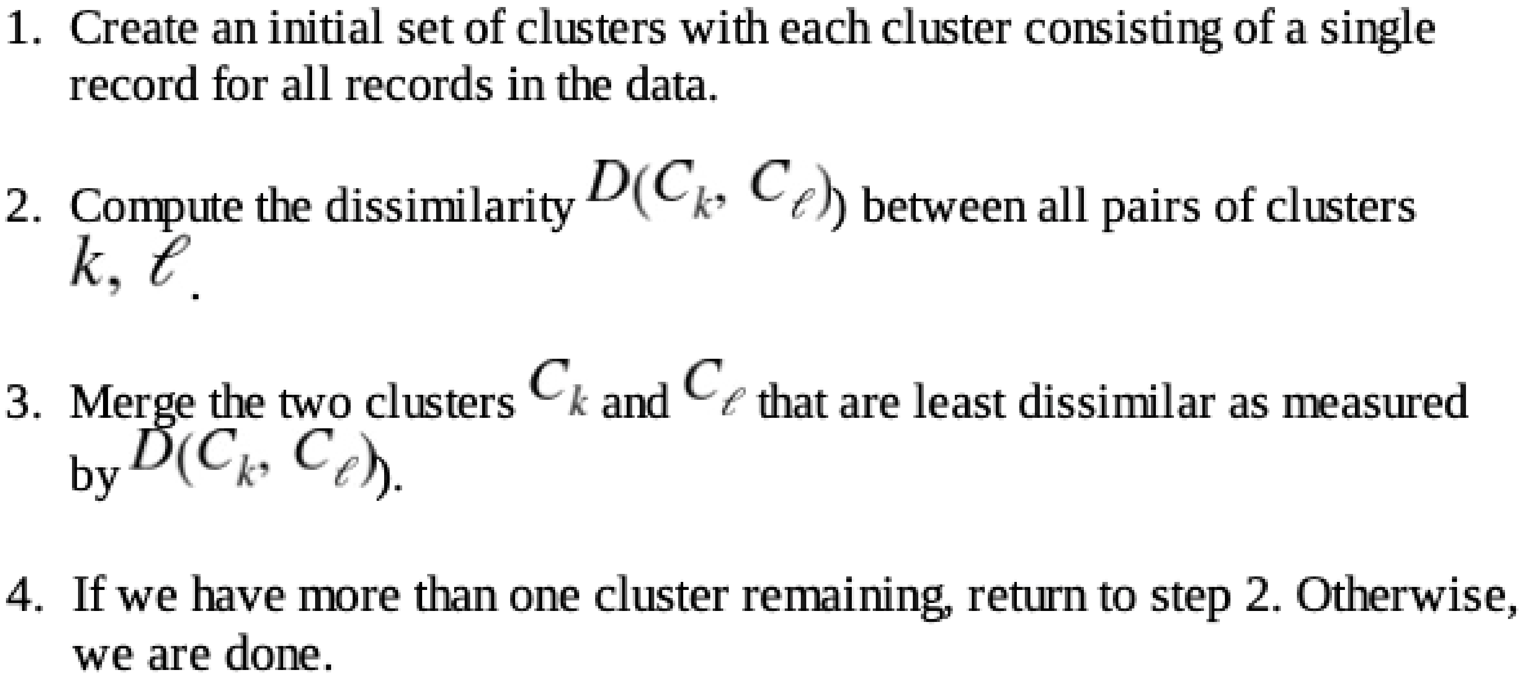
Dendrogram

- Hierarchical clustering lends itself to a natural graphical display as a tree
- Leaves of the tree correspond to the records
- Length of the branch in the tree indicates the degree of dissimilarity between clusters
- Example: The returns for Google and Amazon are quite dissimilar to the returns for the other stocks



Agglomerative algorithm

- Agglomerative algorithm merges similar clusters iteratively
- Begins with each record constituting its own single-record cluster, then builds up larger and larger clusters
- Algorithm:

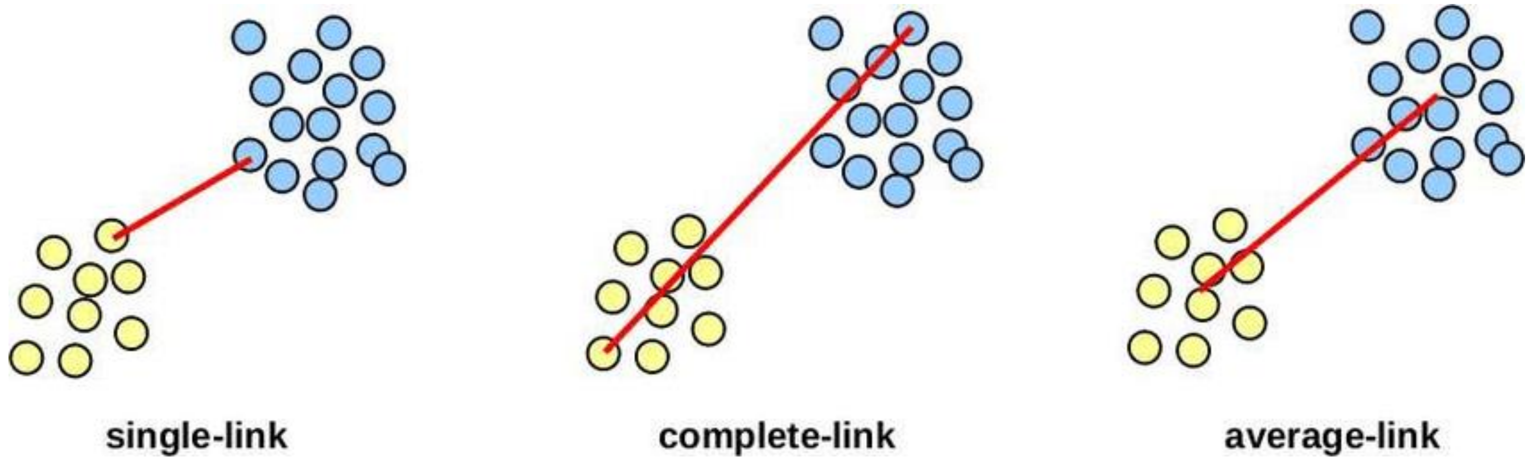


1. Create an initial set of clusters with each cluster consisting of a single record for all records in the data.
2. Compute the dissimilarity $D(C_k, C_\ell)$ between all pairs of clusters k, ℓ.
3. Merge the two clusters C_k and C_ℓ that are least dissimilar as measured by $D(C_k, C_\ell)$.
4. If we have more than one cluster remaining, return to step 2. Otherwise, we are done.

Measures of dissimilarity

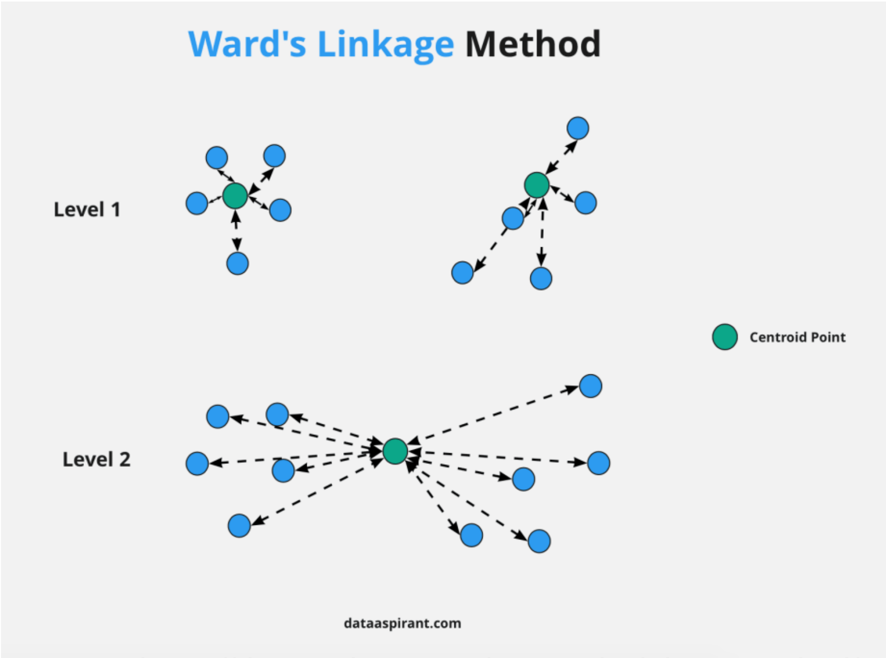
- *D(A, B)*: measures the dissimilarity between two clusters, A and B, by using the distances between the members of A and the members of B
- Four common dissimilarity measures:
    - Complete-linkage: the maximum distance across all pairs of records between A and B
    - Single-linkage: the minimum distance between the records in A and B
    - Average-linkage: the average of all distance pairs (i.e., a compromise between the single and complete linkage methods)
    - Minimum variance method (Ward's linkage): this minimizes the within-cluster sum of squares (like K-means)

Measures of dissimilarity (example)



Source: https://www.researchgate.net/figure/The-three-linkage-types-of-hierarchical-clustering-single-link-complete-link-and_fig57_281014334

Ward-linkage example: calculates the increase in the within-cluster sum of squares before and after fusing two clusters



Source: https://dataaspirant.com/14-wards-linkage-method/

Model-based clustering

- K-means and hierarchical clustering: distance measured directly with the data (no probability model involved)
- Model-based clustering methods are grounded in statistical theory (and provide more rigorous ways to determine the nature and number of clusters)
- (Generally) relies on multivariate normal distribution
- Use case: There might be a case that one group of records that are similar to one another but not necessarily close to one another

Multivariate normal distribution

- Generalization of the normal distribution to set of *p* variables (X_1, X_2, .., X_p)
- Distribution is defined by a set of means μ = μ_1, μ_2, .., μ_p and a covariance matrix ∑, consisting of *p* variances σ_1, σ_2, .., σ_p, and covariances $\sigma_{i,j}$ for all pairs of variables

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{1,2} & \cdots & \sigma_{1,p} \\ \sigma_{2,1} & \sigma_2^2 & \cdots & \sigma_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p,1} & \sigma_{p,2}^2 & \cdots & \sigma_p^2 \end{bmatrix}$$

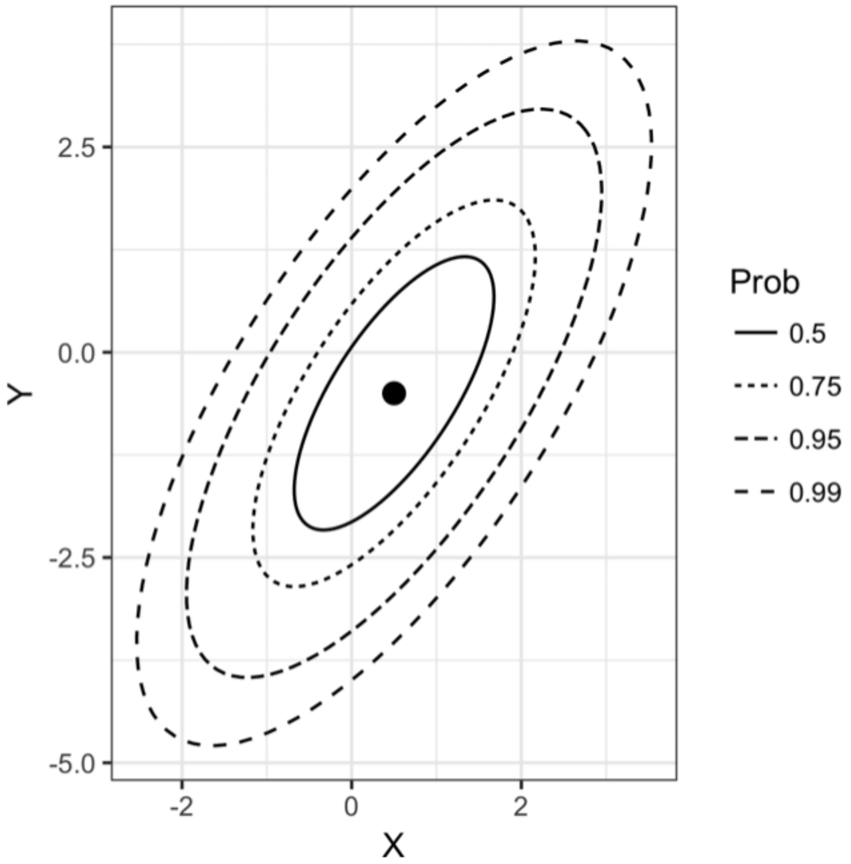
- The distribution is denoted by:

$$(X_1,\ X_2,\ \ldots,\ X_p)\widetilde{N}_p(\mu,\ \Sigma)$$

Example: 2D normal distribution

- Probability contours for a multivariate normal distribution for two variables *X* and *Y*
- For example, *t*he 0.5 probability contour contains 50% of the distribution
- Means μ_x=0.5 and μ_y=-0.5
- Covariance matrix:

$$\Sigma = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$



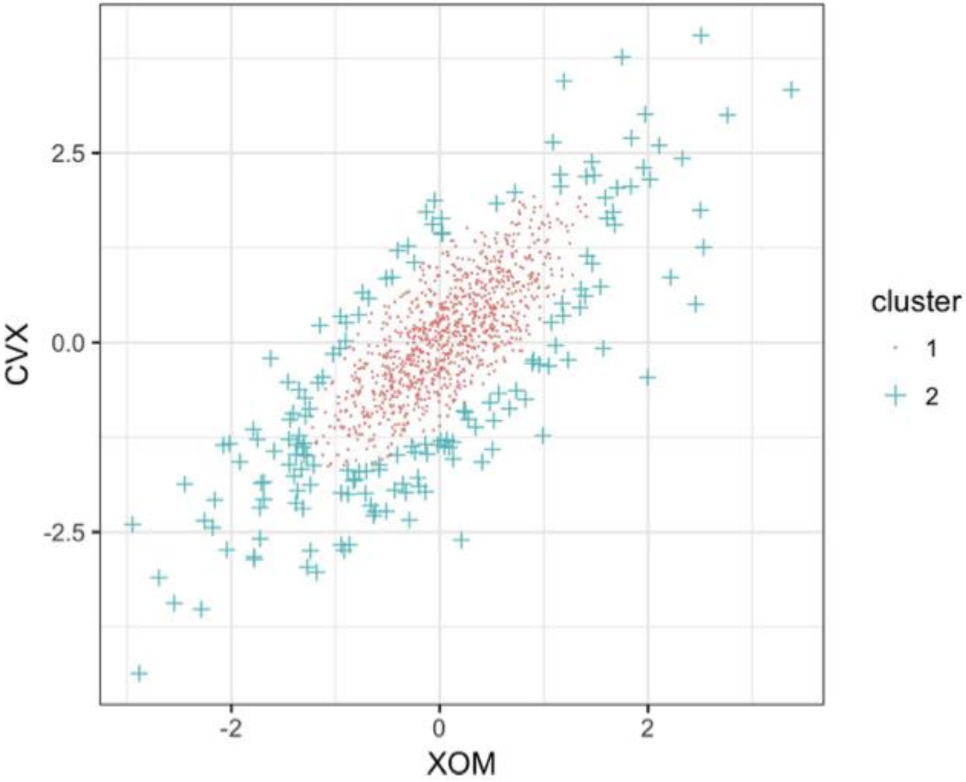
Mixtures of Normals

- Key idea behind model-based clustering is that each record is assumed to be distributed as one of *K* multivariate-normal distributions (*K*=number of clusters)
- Each distribution has a different mean μ and covariance matrix ∑
- Example: For two variables, X and Y, each row (X_i, Y_i) is modeled as having been sampled from one of *K* distributions:

$$N_1(\mu_1, \Sigma_1),\ N_2(\mu_2, \Sigma_2),\ \ldots,\ N_k(\mu_k, \Sigma_k)$$

Example: model-based clustering

- The distributions show similar means and correlations, but the second distribution has much larger variances and covariances
- Unlike *K*-means and hierarchical clustering, model-based clustering internally determines the number of clusters (e.g., using AIC or BIC)



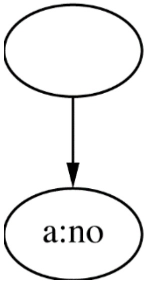
Incremental clustering

- Heuristic approach (COBWEB/CLASSIT)
- Forms a hierarchy of clusters incrementally
- Start:
  - tree consists of empty root node
- Then:
  - add instances one by one
  - update tree appropriately at each stage
  - to update, find the right leaf for an instance
  - may involve restructuring the tree using *merging* or *splitting* of nodes
- Update decisions are based on a goodness measure called *category utility*

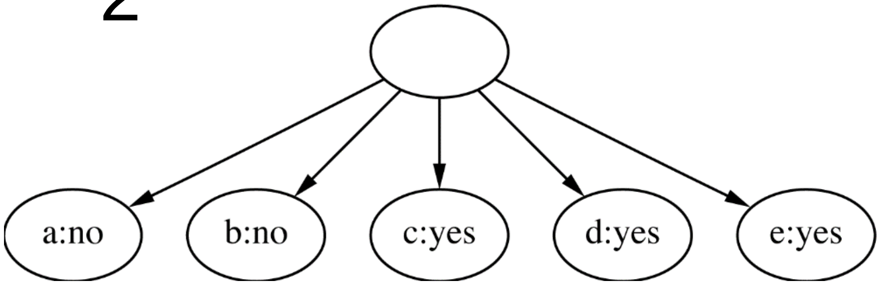
Example: incremental culstering

ID	Outlook	Temp.	Humidity	Windy
A	Sunny	Hot	High	False
B	Sunny	Hot	High	True
C	Overcast	Hot	High	False
D	Rainy	Mild	High	False
E	Rainy	Cool	Normal	False
F	Rainy	Cool	Normal	True
G	Overcast	Cool	Normal	True
H	Sunny	Mild	High	False
I	Sunny	Cool	Normal	False
J	Rainy	Mild	Normal	False
K	Sunny	Mild	Normal	True
L	Overcast	Mild	High	True
M	Overcast	Hot	Normal	False
N	Rainy	Mild	High	True

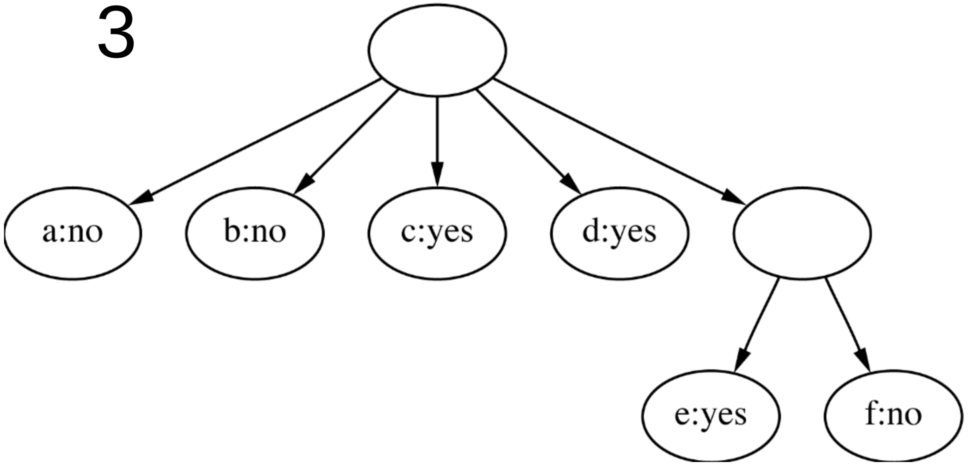
1



2



3



Summary

- Unsupervised learning
- Principal component analysis
- K-means: finds k clusters (using the Euclidean distance metric)
- Agglomerative clustering forming a hierarchical tree for clustering
- Model-based clustering based on probabilistic density