Descriptive Modelling

Rita P. Ribeiro

Machine Learning - 2022/2023

Today

References

- Moreira, João, et al. 2018.
  Data Analytics: A General Introduction. Ch 5
- Gama, João, et al. 2015.
  Data Mining 3rd Ed. Ch 12, Ch 13.1 - 13.3, 15.2.1
- Aggarwal, Charu C. 2015.
  Data Mining, the Textbook. Ch 6.3, 6.4, 6.6.2, 6.9.1

Today

- Descriptive Analytics
- Descriptive Modelling
  - Partitional Clustering
  - Hierarchical Clustering

Descriptive Analytics

Descriptive Analytics

- Main Goal: Describe/summarize or finding structure on what we have observed

- Data summarization and visualization (e.g. PCA) can be seen as simple forms of descriptive analytics

- However, most frequently descriptive modeling is associated with clustering

Similarity Measures

- How to measure similarity between objects?
- The notion of similarity is strongly related with the notion of distance between observations
- It can be measured as the ooposite of the distance

ID	Income	Position	Age
1	2500	manager	35
2	2750	manager	30
3	4550	director	50

- Which cases are more similar?

Similarity Measures

- Similarity measure
  - Numerical measure of how alike two data objects are.
  - Is higher when objects are more alike.
  - Often falls in the range [0,1]

- Dissimilarity measure
  - Numerical measure of how different two data objects are
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0 Upper limit varies

Proximity refers to a similarity or dissimilarity

Similarity Measures

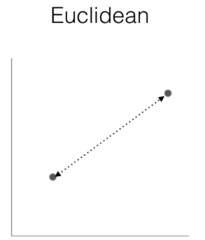
- Dissimilarity measure can be expressed by a distance metric
- Distance metrics d have some well-known properties
- Triangle Inequality
- Given two data points x_i and x_j
  - $d(\mathrm{x}_i, \mathrm{x}_j) \geq 0$
  - $d(\mathrm{x}_i, \mathrm{x}_j) = 0$ only if $\mathrm{x}_i = \mathrm{x}_j$
  - $d(\mathrm{x}_i, \mathrm{x}_j) = d(\mathrm{x}_j, \mathrm{x}_i)$
  - $d(\mathrm{x}_i, \mathrm{x}_j) \leq d(\mathrm{x}_i, \mathrm{x}_k) + d(\mathrm{x}_k, \mathrm{x}_j)$ for any point x_i, x_j and x_k

Similarity Measures

Euclidean Distance

$$d(\mathrm{x}_i, \mathrm{x}_j) = \sqrt{\sum_{a=1}^{m} (x_i^a - x_j^a)^2}$$

where m is the number of attributes and x_i^a and x_j^a are the a^{th} attribute value for the data points x_i and x_j, respectively

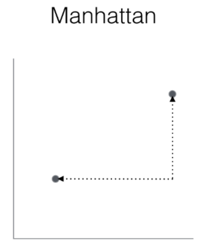


Similarity Measures

Manhattan Distance

$$d(\mathrm{x}_i, \mathrm{x}_j) = \sum_{a=1}^{m} |x_i^a - x_j^a|$$

where m is the number of attributes and x_i^a and x_j^a are the a^{th} attribute value for the data points x_i and x_j, respectively



Similarity Measures

A Generalization: Minkowski Distance

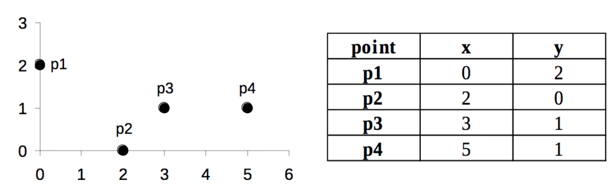
$$d(\mathrm{x}_i, \mathrm{x}_j) = \sqrt[p]{\sum_{a=1}^{m} |x_i^a - x_j^a)^p}$$

where if

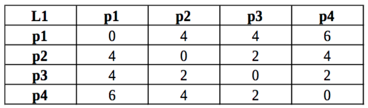
- $p = 1$, we have the Manhattan Distance (or L_1-norm)
- $p = 2$, we have the Euclidean Distance (or L_2-norm)
- $\ldots$
- $p = \infty$, we have Chebyschev or *supremum* distance (or L_∞-norm):
  - maximum difference between any of the attributes of the data points.

Similarity Measures

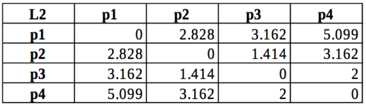
Example of Minkowski Distances: L_1-norm, L_2-norm and L_∞-norm



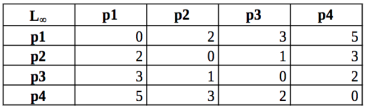
point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1



L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0



L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0



L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Similarity Measures

- More examples of similarity/distance measures
  - Canberra distance
  - Jaccard Coefficients
  - Cosine similarity
- Several problems may arise that may distort the notion of distance:
  - different scales of variables
  - different importance of variables
  - different types of data (e.g. both numeric and categorical variables)

Similarity Measures

Heterogeneous Distance Functions

$$d(\mathrm{x}_i, \mathrm{x}_j) = \sum_{a=1}^{m} \delta_a(x_i^a, y_i^a)$$

where

- if *a* is a categorical variable

$$\delta_a(x_i^a, x_j^a) = \begin{cases} 0 & \text{if } x_i^a == x_j^a \\ 1 & \text{otherwise} \end{cases}$$

- if *a* is a numeric variable

$$\delta_a(x_i^a, x_j^a) = \frac{|x_i^a - x_j^a|}{|max_a - min_a|}$$

Similarity Measures

General Coefficient of Similarity

$$s(\mathrm{x}_i, \mathrm{x}_j) = \sum_{a=1}^{m} w_a s(x_i^a, y_i^a) / \sum_{a=1}^{m} w_a$$

- $s()$ is a similarity measure, m is the number of attributes,
- x_i^a and x_j^a are the a^{th} attribute value for x_i and x_j, respectively,
- $w_a \in [0, 1]$ is the weight for the attribute a.

- Given any two data points x_i and x_j
  - $s(\mathrm{x}_i, \mathrm{x}_j) = 1$, only if $\mathrm{x}_i == \mathrm{x}_j$
  - $s(\mathrm{x}_i, \mathrm{x}_j) = s(\mathrm{x}_j, \mathrm{x}_i)$

Clustering

Clustering

Goals:

- Obtain the “natural” grouping of a set of data, i.e. find some structure on the data set
  - The key issue on clustering is the notion of similarity
  - Observations on the same group are supposed to share some properties, i.e. being similar
  - Most clustering methods use the information on the distances among observations in a data set
- Provide some abstraction of the found groups (e.g. a representation of their main features; a prototype for each group; etc.), gain novel insights of data

Clustering: Some Applications

- Biology
  - describe spatial and temporal communities of organisms
  - group genes or proteins that have similar functionality
- Business and Marketing
  - describe different market segments from a set of potetential clients
  - group stocks with similar price fluctuations
- Web Mining
  - find groups of related documents for information retrieval
  - find communities in social networks
  - build recommender systems
- . . .

Clustering: Main Types of Methods

- Partitional: divide the observations in *k* partitions according to some criterion
- Hierarchical: generate a hierarchy of groups, from 1 to n groups, where n is the number of lines in the data set
  - Agglomerative: create a hierarchy bottom up (from n to 1 group)
  - Divisive: create a hierarchy top down (from 1 to n groups)

Clustering Partitional Methods

Goal: Partition the given set of data into k groups by either minimizing or maximizing a pre-specified criterion

- Some key issues:
  - The choice of the number of groups
  - The nr of possible divisions of n cases into k groups can grow fast!

$$N(n,k) = \frac{1}{k!}\sum_{i=1}^{k}(-1)^{k-i}\binom{k}{i}i^n$$

e.g. for $n = 100$ and $k = 5$, $N(100,5) \approx 6.6 \cdot 10^{67}$

Clustering Partitional Methods

Some important properties

- Cluster compactness
  - how similar are cases within the same cluster
- Cluster separation
  - how far is the cluster from the other clusters

Goal:

minimize intra-cluster distance and maximize inter-cluster distances

- A clustering solution assigns all the objects to a cluster
  - *hard clustering*: an object belongs to a single cluster
  - *fuzzy clustering*: each object has a probability associated to belong to each cluster

Clustering Partitional Methods

Consider the cluster $C_k = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_{n_k}\}$, the centroid of C_k is given by

$$\bar{\mathrm{x}}^{(k)} = \frac{1}{n_k} \sum_{\mathrm{x}_i \in C_k} \mathrm{x}_i$$

the centroid of C_k can also be the median of its data objects, i.e. $\tilde{\mathrm{x}}^{(k)}$

Goal: obtain a set of clusters C that minimize

$$h(\mathrm{C}) = \sum_{j=1}^{k} \sum_{\mathrm{x}_i \in C_j} d(\mathrm{x}_i, \bar{\mathrm{x}}^{(j)})$$

(Some) Criteria for numeric data

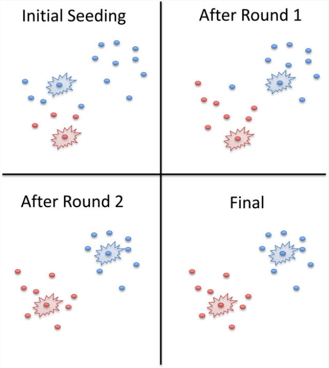
- Sum of Squared Errors (SSE): $d(\mathrm{x}_i, \bar{\mathrm{x}}^{(j)}) = (\mathrm{x}_i - \bar{\mathrm{x}}^{(j)})^2$
- L_1 measure: $d(\mathrm{x}_i, \bar{\mathrm{x}}^{(j)}) = |\mathrm{x}_i - \tilde{\mathrm{x}}^{(j)}|$

Clustering Partitional Methods: *k*-Means

It is a partition-based method that obtains *k* groups of a data set

k-means algorithm

- Initialize the centers of the *k* groups to a set of randomly chosen observations
- Repeat
  - Allocate each observation to the group whose center is nearest
  - Re-calculate the center of each group
- Until the groups are stable, i.e. there is no significant decrease or there is an increase on the minimize criterion $h(\mathrm{C})$



Clustering Partitional Methods: *k*-Means

Some observations:

- Clusters have always a convex shape
    - line connecting any two instances in the cluster lies within the cluster
- The shape depends on the distance function
    - hypercube for Manhattan distance
    - hypersphere for Euclidean distance
    - hyperellipsoid for Mahalanobis distance
- Typically, it uses the Euclidean distance as criterion
- Maximizes inter-cluster dissimilarity

Clustering Partitional Methods: *k*-Means

Advantages:

- Fast algorithm that scales well
- Stochastic approach that frequently works well. It tends to identify local minima.

Disadvantages:

- It does not ensure an optimal clustering
- We may obtain different solutions with different starting points
- The initial guess of k for the number of clusters, maybe away from the real optimal value of k.

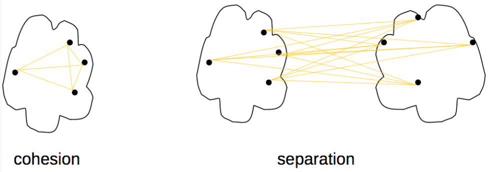
Clustering Validation

How to validate/evaluate/compare the results obtained by some clustering method?

- Is the found group structure random?
- What is the “correct” number of groups?
- How to evaluate the result of a clustering algorithm when we do not have information on the number of groups in the data set?
- How to compare the results obtained by different methods when outside information on the number of groups exists?
- How to compare alternative solutions (e.g. obtained using different clustering algorithms)?

Clustering Validation: Types of Evaluation Measures

- Supervised - compare the obtained clustering (grouping) with the external information that we have available
- Unsupervised - try to measure the quality of the clustering without any information on the “ideal” structure of the data
  - Cohesion coefficients - determine how compacts/cohesive are the members of a group
  - Separation coefficients - determine how different are the members of different groups



Clustering Validation: Silhouette Coefficient

Silhouette Coefficient (unsupervised measure)

- Popular coefficient that incorporates both the notions of cohesion and separation
- For each object x_i:
  - obtain the average distance to all objects in the same group (a_i)
  - to any other group to which x_i does not belong, calculate the average distance to the members of these other groups; obtain the minimum value of these distances (b_i)
  - The silhouette coefficient, s_i is equal to

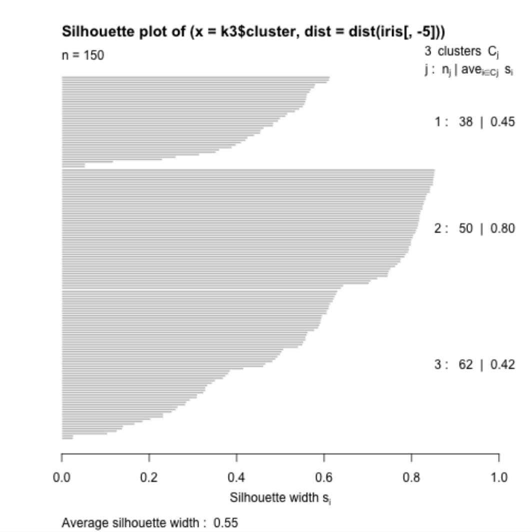
$$s_i = \frac{b_i - a_i}{max(a_i, b_i)}$$

- The coefficient takes values between -1 and 1.

Clustering Validation: Silhouette Coefficient (cont.)

Example: iris data set silhouette coefficients s_i with $k = 3$ clusters

- Large s_i (almost 1) means that they are very well clustered.
- Small s_i (around 0) means that they lie between two clusters.
- Negative s_i means that they are probably placed in the wrong cluster.
- The closer average silhouette to 1, the better.



Clustering Validation: Best Number of Clusters

How to select the right *k* for k-means?

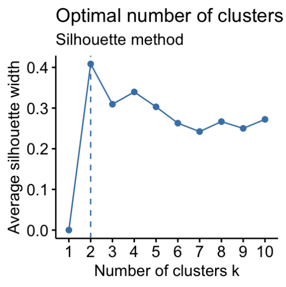
- An inappropriate choice of k can result in a clustering with poor performance.
- What happens if we select a k that is too high?
- What if the k is too low?
- Ideally, you should have some a priori knowledge on the real structure of the data.
- If no a priori value is known start with $\sqrt{n/2}$ as a rule of thumb, where n is the number of attributes.

Clustering Validation: Best Number of Clusters

Silhouette-based method

For several possible number of clusters k:

- Calculate the average silhouette coefficient value and choose the k that yields to the highest value

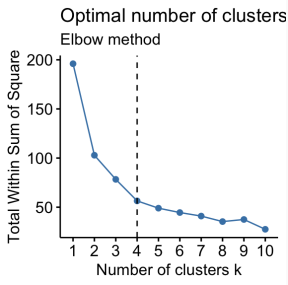


Clustering Validation: Best Number of Clusters (cont.)

Elbow method

For several possible number of clusters *k*:

- Calculate the within-cluster SSE, also called distortion, and choose the *k* so that adding another cluster doesn't yield to a much smaller SSE.



Other, more sophisticated methods exist (e.g. intracluster to intercluster distance ratio)

Other Clustering Partitional Methods

PAM (Partitioning Around Medoids)

- It searches for the *k* representative objects (the medoids) among the cases in the given data set.
- As with k-means each observation is allocated to the nearest medoid.
- Is more robust to the presence of outliers because it uses original objects as centroids instead of averages that may be subject to the effects of outliers.
- Moreover, it uses a more robust measure of the clustering quality: $L_1 - norm$, which is based on absolute error instead of the squared error used in k-means,

Other Clustering Partitional Methods

CLARA (Clustering Large Applications)

- The PAM algorithm has several advantages in terms of robustness when compared to k-means.
- However, these advantages come at the price of aditional computational complexity that may be critical for large data sets
- CLARA tries to solve these efficiency problems
- It does that by using sampling, i.e. working on parts of the data set instead of the full data set

Other Clustering Partitional Methods

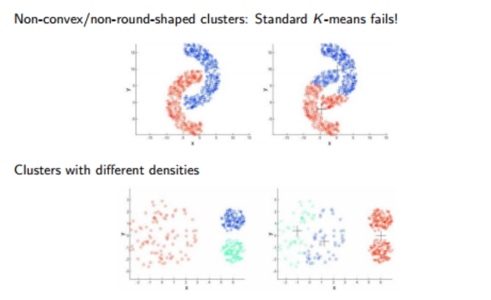
CLARA Algorithm

- Repeat *n* times the following:
  - Draw a random sample of size *m*
  - Apply PAM to this random sample to obtain *k* centroids
  - Allocate the full set of observations to one of these centroids
  - Calculate sum of dissimilarities of the resulting clustering (as in PAM)
- Return as result the clustering of the n repetitions that got lowest sum of dissimilarities

Other Clustering Partitional Methods

These “k-means like” methods have problems with:

- clusters of different sizes, densities and with non-globular shape



Non-convex/non-round-shaped clusters: Standard K-means fails!

Clusters with different densities

- data that contains outliers/noise

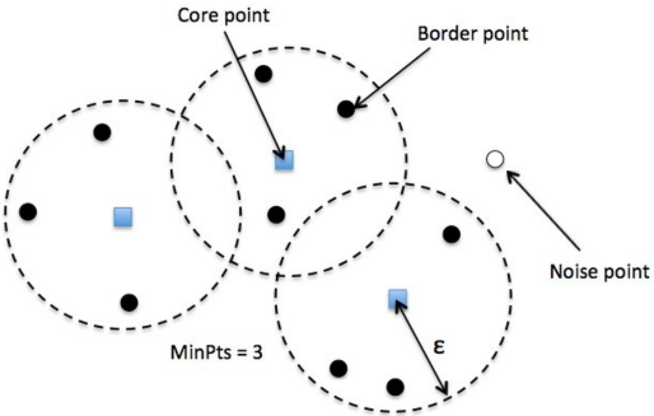
Other Clustering Partitional Methods

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

- The density of an observation is estimated by the nr. of observations within a certain radius (a parameter of the method)
- Based on this idea observations are classified as:
  - core points: if the nr. of observations within its radius are above a certain threshold
  - border points: if the nr. of observations within their radius does not reach the threshold but they are within the radius of a core point
  - noise points: they do not have enough observations within their radius, nor are they sufficiently close to any core point

Other Clustering Partitional Methods

DBSCAN: Core, Border and Noise Points



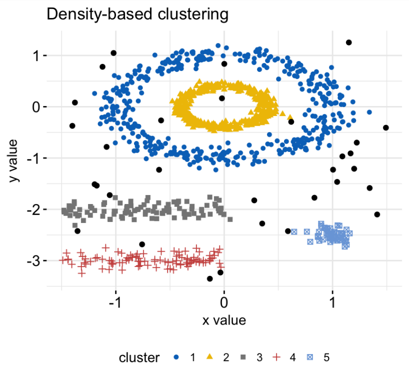
Other Clustering Partitional Methods

- DBSCAN Algorithm
  - Classify each observation in one of the three possible alternatives
  - Eliminate the noise points from the formation of the groups
  - All core points that are within a certain distance of each other are allocated to the same group
  - Each border point is allocated to the group of the nearest core point
- Note that this method does not require the user to specify the number of groups.
- But, you need to specify the radius (ε) and the minimum number of points (MinPts)

Other Clustering Partitional Methods (cont.)

DBSCAN

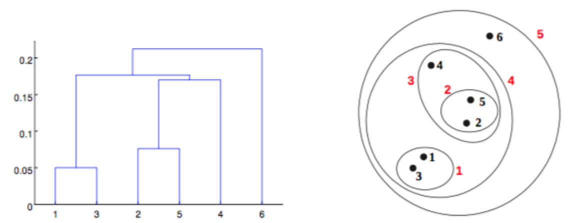
- Advantages:
  - Can detect clusters of an arbitrary shape
  - Resistant to noise
- Disadvantages:
  - Computationally more complex than k-means
  - Difficulty in setting the hyper-parameter values



Hierarchical Clustering

Goal:

- Obtain a hierarchy of groups, where each level represents a possible solution with *k* groups.
- It is up to the user to select the solution he wants.
- A dendogram can be used for visualization



Hierarchical Clustering

- Agglomerative Methods - *bottom-up*
  - Start with as many groups as there are cases
  - On each upper level a pair of groups is merged into a single group
  - The chosen pair is formed by the groups that are more similar

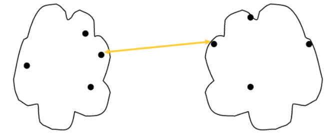
- Divisive Methods - *top-down* (much less used)
  - Start with a single group
  - On each level select a group to be split in two
  - The selected group is the one with smallest uniformity

Hierarchical Clustering

Some proximity measures for the merging/splitting step

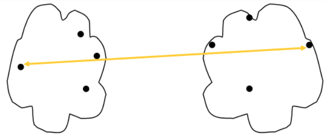
- single link

$$d(C_1, C_2) = \min_{\mathrm{x}_i \in C_1, \mathrm{x}_j \in C_2} d(\mathrm{x}_i, \mathrm{x}_j)$$



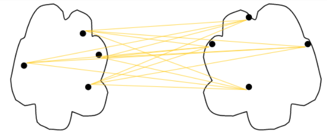
- complete link

$$d(C_1, C_2) = \max_{\mathrm{x}_i \in C_1, \mathrm{x}_j \in C_2} d(\mathrm{x}_i, \mathrm{x}_j)$$



- average link

$$d(C_1, C_2) = \frac{1}{n_1 n_2} \sum_{\mathrm{x}_i \in C_1, \mathrm{x}_j \in C_2} d(\mathrm{x}_i, \mathrm{x}_j)$$



Other methods: distance between the centroids, Ward's method (uses SSE).

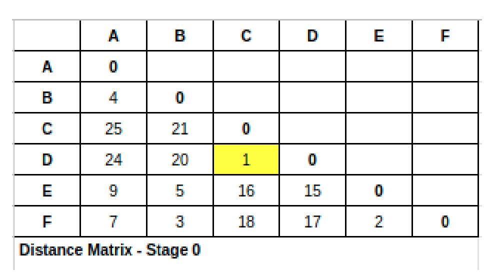
Hierarchical Clustering: Agglomerative Methods

Hierarchical Agglomerative Algorithm

- Compute the proximity matrix
- Let each data point be a cluster
- Repeat
  - Merge the two closest clusters
  - Update the proximity matrix
- Until only a single cluster remains

Hierarchical Clustering: Agglomerative Methods

Example: Consider the following distance matrix



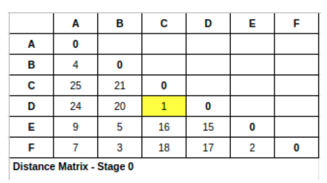
	A	B	C	D	E	F
A	0					
B	4	0				
C	25	21	0			
D	24	20	1	0		
E	9	5	16	15	0	
F	7	3	18	17	2	0

Distance Matrix - Stage 0

- Use agglomerative hierarchical clustering with single-link method

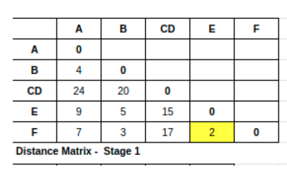
Hierarchical Clustering: Agglomerative Methods

Example: Agglomerative Hierarchical Clustering, single-link method.



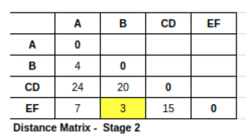
	A	B	C	D	E	F
A	0					
B	4	0				
C	25	21	0			
D	24	20	1	0		
E	9	5	16	15	0	
F	7	3	18	17	2	0

Distance Matrix - Stage 0



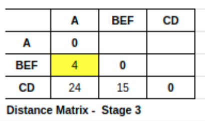
	A	B	CD	E	F
A	0				
B	4	0			
CD	24	20	0		
E	9	5	15	0	
F	7	3	17	2	0

Distance Matrix - Stage 1



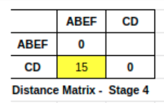
	A	B	CD	EF
A	0			
B	4	0		
CD	24	20	0	
EF	7	3	15	0

Distance Matrix - Stage 2



	A	BEF	CD
A	0		
BEF	4	0	
CD	24	15	0

Distance Matrix - Stage 3

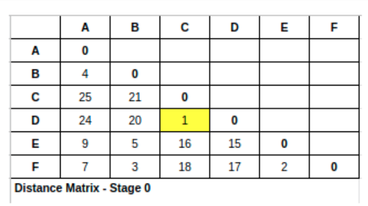


	ABEF	CD
ABEF	0	
CD	15	0

Distance Matrix - Stage 4

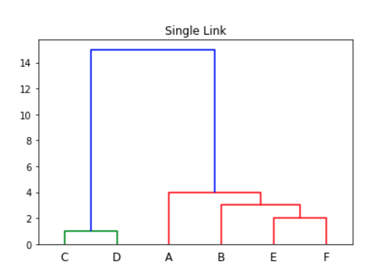
Hierarchical Clustering: Agglomerative Methods

Example: Agglomerative Hierarchical Clustering, single-link method.



	A	B	C	D	E	F
A	**0**					
B	4	**0**				
C	25	21	**0**			
D	24	20	1	**0**		
E	9	5	16	15	**0**	
F	7	3	18	17	2	**0**

Distance Matrix - Stage 0



Hierarchical Clustering: Agglomerative Methods

Different proximity measures yield to different types of clusters.

- single-link
  - can handle non-elliptical shapes
  - uses a local merge citerion
  - distant parts of the cluster and the clusters' overall structure are not taken into account

Hierarchical Clustering: Agglomerative Methods

Different proximity measures yield to different types of clusters.

- complete-link
    - biased towards globular clusters
    - uses a non-local merge citerion
    - chooses the pair of clusters whose merge has the smallest diameter
    - the similarity of two clusters is the similarity of their most dissimilar members
    - sensitive to noise/outliers
- average-link
    - it is a compromise between single and complete link

Hierarchical Clustering: Divisive Methods

Hierarchical Divisive Algorithm

- Compute the proximity matrix
- Start with a single cluster that contains all data points
- Repeat
    - choose the cluster with the largest diameter, i.e. largest dissimilarity between any two of its points
    - select the data point with largest average dissimilarity to the other members in that cluster
    - re-allocate the data points to either the cluster of this selected point or the “old” cluster (represented by its center), depending on which one is nearest
- Until each data point constitutes a cluster

Clustering Methods: Wrap-up

Overall, we can compare clustering methods w.r.t

- Algorithm:
  - complexity and scalability
  - similarity measures that can be employed
  - robustness to noise
  - it is able to find clusters on sub-spaces
  - different runs lead to different results
  - it is incremental

Clustering Methods: Wrap-up

Overall, we can compare clustering methods w.r.t

- Data:
    - it is able to handle different types of data?
        - continuous, categorical, binary
    - is there dependency on the order of data points?
- Domain:
    - does the algorithm finds the number of clusters, or needs it as input?
    - how many parameters are necessary?
    - what is the required domain knowledge for that?
- Results:
    - shape of clusters that is able to find
    - interpretability

References

References

Aggarwal, Charu C. 2015. *Data Mining, the Texbook*. Springer.

Gama, João, André Carlos Ponce de Leon Ferreira de Carvalho, Katti Faceli, Ana Carolina Lorena, and Márcia Oliveira. 2015. *Extração de Conhecimento de Dados: Data Mining -3rd Edition*. Edições Sílabo.

Gandomi, Amir, and Murtaza Haider. 2015. "Beyond the Hype: Big Data Concepts, Methods, and Analytics." *International Journal of Information Management* 35 (2): 137–44. https://doi.org/https://doi.org/10.1016/j.ijinfomgt.2014.10.007.

Han, Jiawei, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques*. 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

"R Project." 2021. https://www.r-project.org/.

Tan, Pang-Ning, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. 2018. *Introduction to Data Mining*. 2nd ed. Pearson.