2.1 Descriptive Modelling

Descriptive Analytics

Goals

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

- Data summarization and visualization can be seen as simple forms of descriptive analytics
- However, most frequently descriptive modeling is associated with clustering

Similarity Measures

- Notion of similarity is strongly related with the notion of **distance between observations**
- Can be measured as the opposite of the distance
- Proximity refers to a similarity or dissimilarity

Similarity Measure

- Numerical measure of how alike 2 data objects are
- Higher when objects are more alike
- Often falls in the range [0,1]

Dissimilarity Measure

- Numerical measure of how different 2 data objects are
- Lower when objects are more alike
- Minimum dissimilarity is often 0
- Upper limit varies

Dissimilarity measure can be expressed by a distance metric. Distance metrics d have some well-known properties

Euclidean Distance

$$d(x_i, x_j) = \sqrt{\sum_{a=1}^{n} (x_i^a - x_j^a)^2}$$

Manhattan Distance

$$d(x_i, x_j) = \sum_{a=1}^{n} |x_i^a - x_j^a|$$

Minkowski Distance

$$d(x_i, x_j) = \sqrt[p]{\sum_{a=1}^{n} |x_i^a - x_j^a)^p}$$

where if

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

More examples:

- Canberra distance
- Jaccard Coefficients
- Cosine similarity

Problems:

- different scales of variables
- different importance of variables
- different types of data

Heterogeneous Distance Functions

$$d(x_i, x_j) = \sum_{a=1}^{n} \delta_a(x_i^a, y_i^a)$$

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

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

General Coefficient of Similarity

$$s(x_i, x_j) = \sum_{a=1}^{n} w_a s(x_i^a, y_i^a) / \sum_{a=1}^{n} w_a$$

$s()$ is a similarity measure, n is the number of attributes, x_i^a and x_j^a are the a^{th} attribute value for the data points x_i and x_j, respectively, and w_a is a value between 0 and 1 corresponding to the weight contribution of the attribute a.

Clustering

Goals

- Obtain the "natural" grouping of a set of data
  - **Key issue**: notion of similarity
  - Observations on the same group are supposed to share some properties
  - Most methods use the information on the distances among observations in a data set to decide on the natural grouping of the cases
- Provide some abstraction of the found groups, gain novel insights of data

Applications

- **Biology** - group genes that have similar functionality
- **Business and Marketing** - group stocks with similar price fluctuations
- **Web Mining** - find communities in social networks

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**: generate a hierarchy from bottom to top (from n to 1 group)
    - **Divisive**: create a hierarchy in a top down way (from 1 to n groups)

Clustering Partitional Methods

Partition the given set of data into k groups by either minimizing/maximizing a pre-specified criterion.

- **Key Issues**:
    - The user needs to select the number of groups
    - The number of possible divisions of n cases into k groups can grow fast
- **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
    - 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

Centroid

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

can also be the median of its data objects

k-Means

Partition-based method that obtains k groups of a data set

Algorithm

- Initialize the center 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
- Uses the squared Euclidean distance as criterion
- Maximizes inter-cluster dissimilarity

Advantages

- Fast algorithm that scales well
- Stochastic approach that frequently works well (tens to identify local minima)

Disadvantages

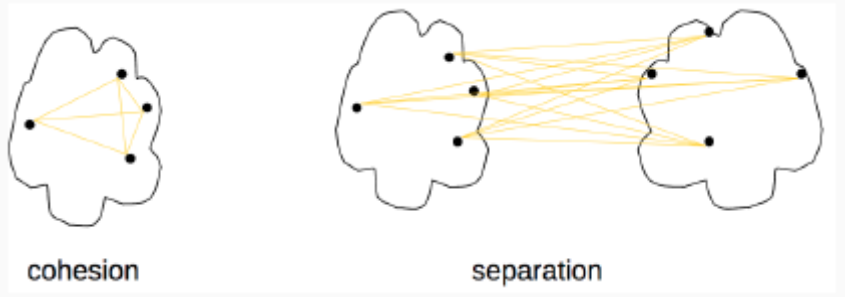
- Does not ensure an optimal clustering
- We may obtain different solutions with different starting points
- The initial guess of k for the number of clusters, maybe away from the real optimal value of k

Clustering Validation

- 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 exists?
- How to compare alternative solutions?

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



Silhouette Coefficient

- Popular coefficient that incorporates both the notions of cohesion and separation
- The coefficient takes values between -1 and 1

Best Number of Clusters

- An inappropriate choice of k can result in a clustering with poor performance
- Ideally you should have some 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 thumbs, where n is the number of attributes

For several possible numbers of clusters, k:

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

Elbow Methods

For several possible numbers 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 Clustering Partitional Methods

PAM (Partitioning Around Medoids)

- 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: L1 - norm, which is bases on absolute error instead of the squared error used in k-means

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 additional computational complexity that may be too much for very large data sets
- CLARA tries to solve these efficiency problems
    - it does that by using sampling

Algorithm

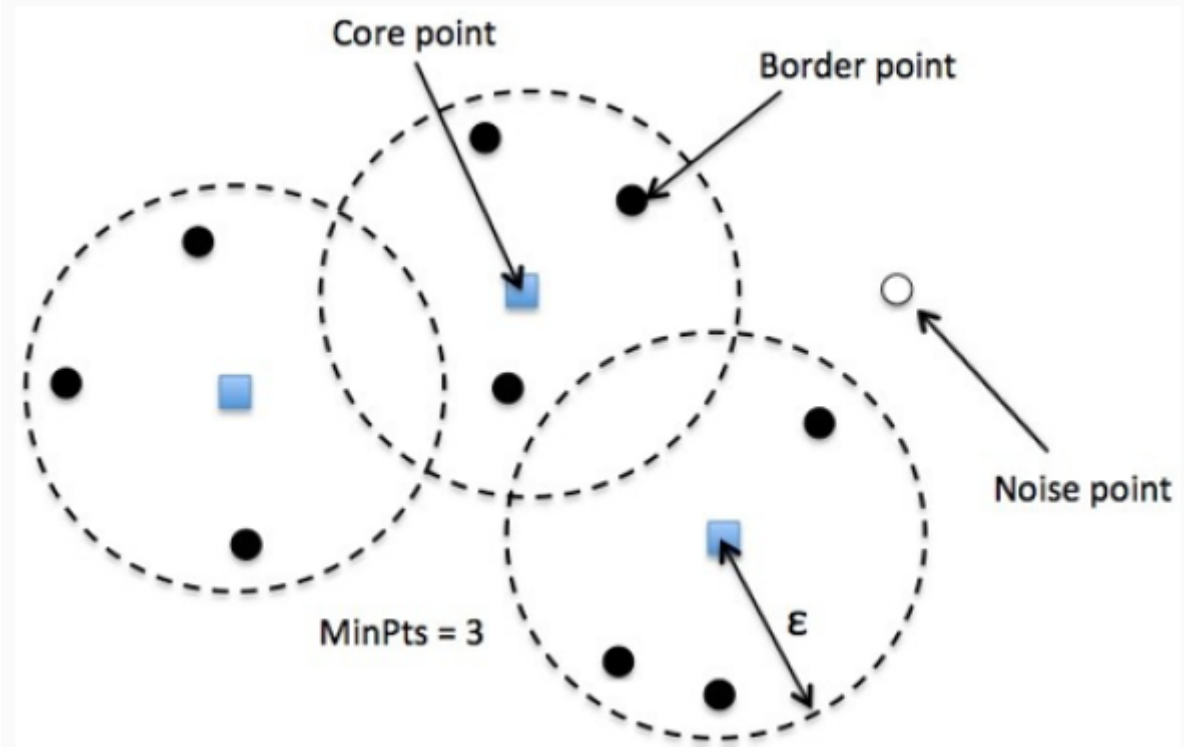
- Repeat n times the following
  - Draw a random sample of size m
  - Apply PAM to this random sample to obtain k centroids
  - Allocate the full set of observations to one of these centroids
  - Calculate sum of dissimilarities of the resulting clustering (as in PAM)
- Return as result of the clustering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

Common problems

- clusters are of different sizes, densities and with non-globular shape
- data contains outliers/noise

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

- The density of a single observation is estimated by the number of observations that are within a certain radius
- Based on this idea observations are classified as:
  - **core points**: if the number of observations within its radius are above a certain threshold
  - **border points**: if the number 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



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: 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)

Advantages

- Can handle clusters with different shapes and sizes
- Resistant to noise

Disadvantages

- Varying densities
- High-dimensional data

Hierarchical Clustering

Goal

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

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

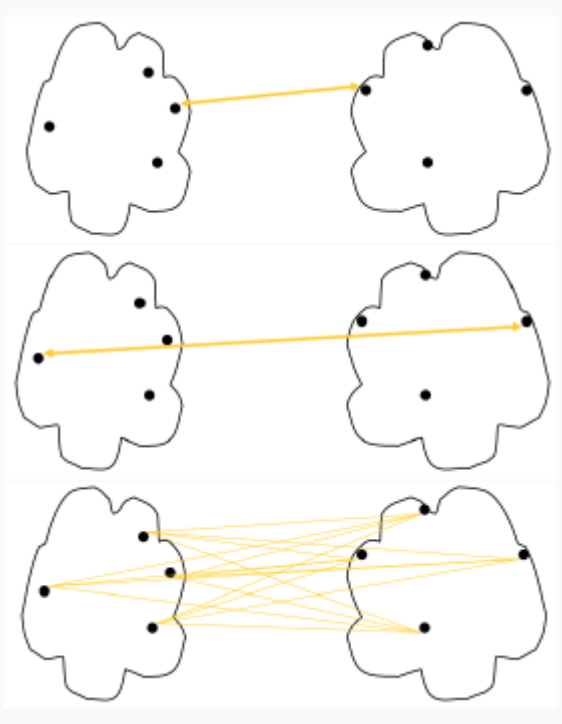
Top-down (much less used)

- Start with a single group
- On each level select a group to be split in 2
- The selected group is the one with smallest uniformity

Proximity measures

- Single link
- Complete link

- Average link



Agglomerative Methods

Algorithm

- Compute the proximity matrix
- Let each data point be a cluster
- Repeat
    - Merge the 2 closest clusters
    - Update the proximity matrix
- Until only single cluster remains

single-link

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

complete-link

- biases towards globular clusters
- uses a non-local merge citerion
- chooses the pair of clusters whose merge has the smallest diameter
- the similarity of 2 clusters is the similarity of their most dissimilar members
- sensitive to noise/outliers

average-link

- it is a compromise between single and complete link

Divisive Methods

Algorithm

- Compute the proximity matrix
- Start with a single cluster that contains all data points
- Repeat
    - choose the cluster with the largest diameter
    - 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

Wrap-up

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

Data

- ability do handle different types of data
- dependency on the order of data points

Domain

- find the number of clusters or needs it as input
- number of parameters necessary
- required domain knowledge

Results

- shape of clusters that is able to find
- interpretability