Implementation K-medoids (PAM) algorithm for clustering objects

Jakub Wieczorek

Faculty of Electronics and Information Technology
Warsaw University of Technology

December 17, 2020

Outline

Outline

Clustering

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters).

- It is a main task of exploratory data mining
- Common technique for statistical data analysis, used in many fields, including:
  - Pattern recognition
  - Image analysis
  - Information retrieval
  - Machine learning

Medoid

- Medoids are representative objects of a data set or a cluster with a data set whose average dissimilarity to all the objects in the cluster is minimal
- Medoids are similar in concept to means or centroids
- Medoids are always restricted to be members of the data set.
- Most commonly used on data when a mean or centroid cannot be defined, such as graphs

k-medoids vs k-means similarities

- The k-medoids problem is a clustering problem similar to k-means
- Both the k-means and k-medoids algorithms break the dataset into clusters and attempt to minimise the distance between points labelled to be in a cluster and a point designated as the centre of that cluster

k-medoids vs k-means differences

- In contrast to the k-means algorithm, k-medoids chooses data points as centres, and thereby allows for greater interpretability of the cluster centres than in k-means where the centre of a cluster is not necessarily one of the input data points (it is the average between the points in the cluster)
- K-medoids can be used with arbitrary dissimilarity measures, whereas k-means generally requires Euclidean distance for efficient solutions
- More robust to noise and outliers than k-means

Outline

Algorithms

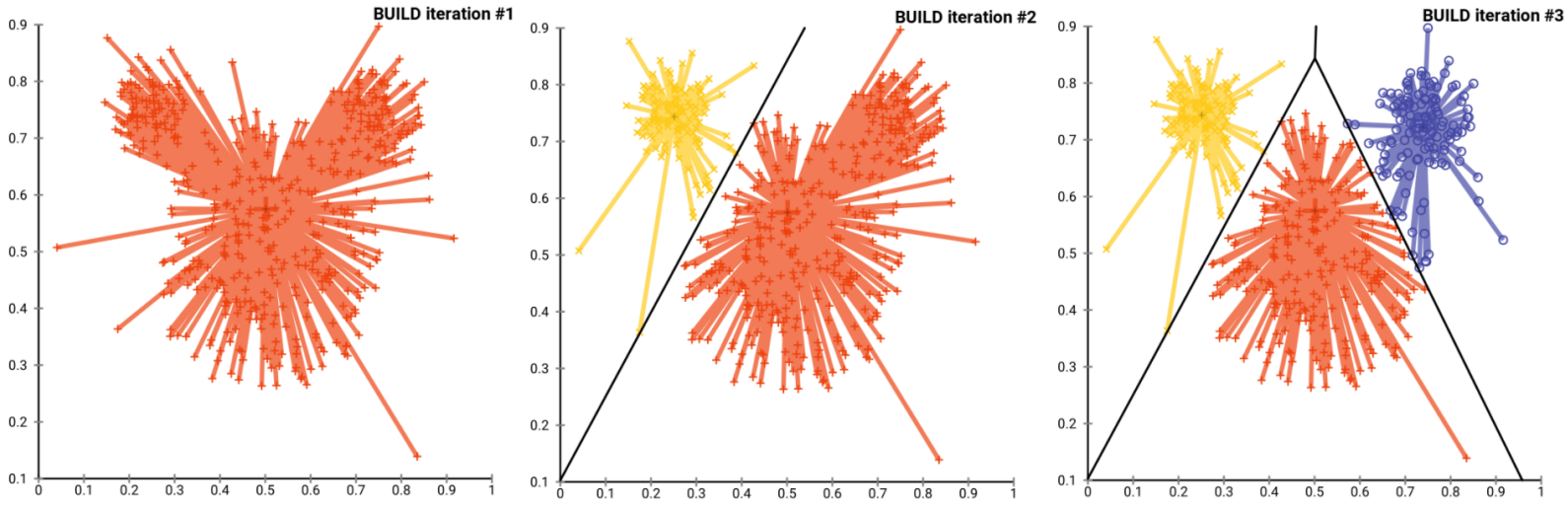
- In general, the k-medoids problem is NP-hard to solve exactly. Many heuristic solutions exist
- Voronoi iteration
- approximate algorithms
  - CLARA
  - CLARANS
- Partitioning Around Medoids (PAM). Uses a greedy search which may not find the optimum solution, but it is faster than exhaustive search

Outline

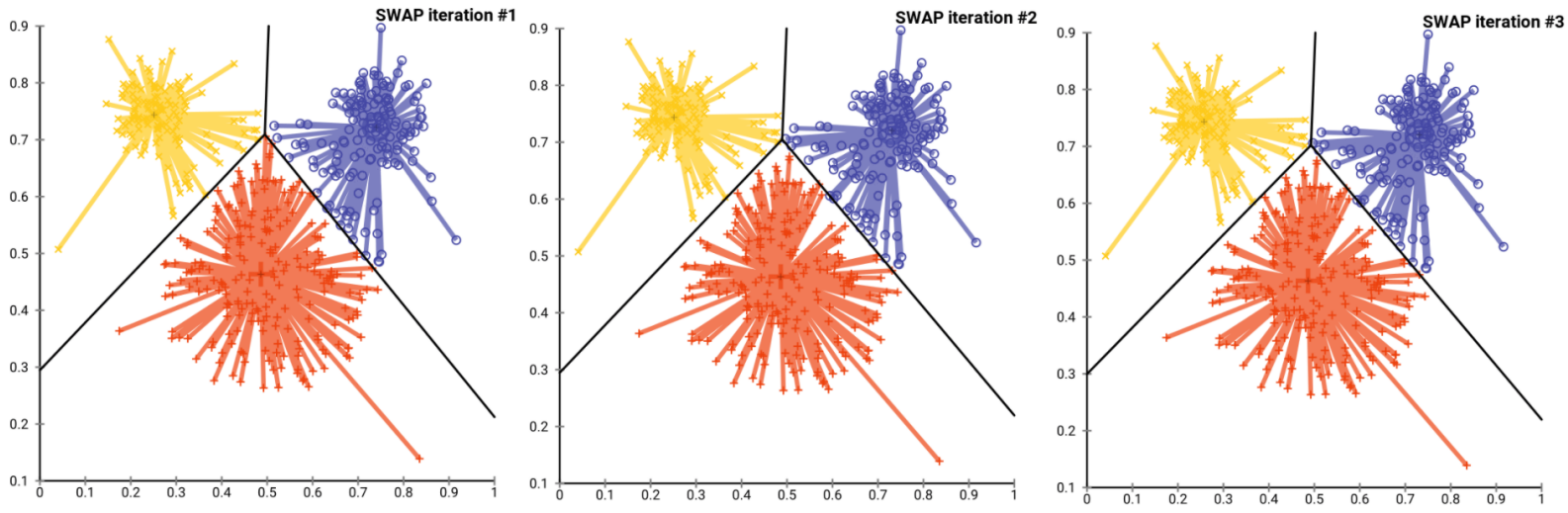
Partitioning Around Medoids (PAM)

1. (BUILD) Initialize: greedily select k of the n data points as the medoids to minimize the cost
2. Associate each data point to the closest medoid
3. (SWAP) While the cost of the configuration decreases:
    1. For each medoid m, and for each non-medoid data point o:
        1. Consider the swap of m and o, and compute the cost change
        2. If the cost change is the current best, remember this m and o combination
    2. Perform the best swap if it decreases the cost function. Otherwise, terminate

Build



SWAP



Thank you for your attention

Jakub Wieczorek