k-Medoids algorithm

Procheta Sen

UNIVERSITY OF LIVERPOOL

Issues with k-Means algorithm

- Results can vary depending on initial random choices
- Can get trapped in a local minimum that isn't the global optimal solution
  - Repeat the clustering procedure multiple times with different initialisations and select the *best* final clustering
- Outliers have a larger effect on the mean value, hence cluster centre and the cluster
- Cluster centres (means) are not actual instances in the cluster
- Euclidean distance used in the algorithm is inappropriate for categorical features

k-Medoids algorithm

- Representative-based algorithms
  - The goal is to determine k representatives $\overline{Y}_1, \ldots, \overline{Y}_k$ that minimise the following objective function

$$\sum_{i=1}^{n} \left[\min_{j} d(\overline{X}_i, \overline{Y}_j) \right]$$

Unlike the k-Means, in the k-Medoids algorithm

- the **distance function** (dissimilarity measure) $d(\cdot, \cdot)$ can be any function convenient for the dataset (not necessarily the Euclidean distance)
- representatives are selected from the dataset

k-Medoids algorithm

Uses **hill-climbing strategy**:

1. Start with an arbitrary solution to a problem
2. Attempt to find a better solution by making an incremental change to the solution.
3. If the change produces a better solution, another incremental change is made to the new solution, and so on
4. Until no further improvements can be found.

k-Medoids algorithm

k-MedoidsClustering (Number of clusters: k, Dataset: $\mathscr{D} = \{\overline{X}_1, \ldots, \overline{X}_n\}$)

1. Initialisation phase

Choose k cluster representatives (**medoids**) $\overline{Y}_1, \ldots, \overline{Y}_k$ from the dataset randomly

2. Assignment phase

Assign all objects in the dataset to the closest representative. The resulting clusters: $C_1, \ldots, C_k$

3. Optimisation phase (hill-climbing step)

1. Find a pair $(\overline{X}, \overline{Y})$, where $\overline{X} \in \mathscr{D}$ and $\overline{Y} \in \{\overline{Y}_1, \ldots, \overline{Y}_k\}$ such that
2. Replacing $\overline{Y}$ with $\overline{X}$ in the set of representatives leads to the greatest possible improvement in the objective function
3. If improvement is positive then replace $\overline{Y}$ with $\overline{X}$ and go to phase 2. Otherwise return current clusters $C_1, \ldots, C_k$

k-Medoids algorithm

Pros

- Representatives are chosen from the dataset
  - allows for greater interpretability of the cluster representatives
  - more robust to noise and outliers than k-means
- Can be used with arbitrary dissimilarity measures
  - thus applicable to data of complex data type (categorical, mixed, time-series, etc.)

Cons

- Results can vary depending on initial random choices
- Can get trapped in a local minimum that isn't the global optimal solution
  - Repeat the clustering procedure multiple times with different initialisations and select the best final clustering
- Slower than the k-means algorithm

k-Medoids algorithm: time-complexity issue

If we have n objects in the dataset, then at each execution of the optimisation phase we need to compute $k \cdot n$ times the incremental objective function change (this is too expensive)

3. Optimisation phase (hill-climbing step)

1. Find a pair $(\overline{X}, \overline{Y})$, where $\overline{X} \in \mathscr{D}$ and $\overline{Y} \in \{\overline{Y}_1, \ldots, \overline{Y}_k\}$ such that
2. Replacing $\overline{Y}$ with $\overline{X}$ in the set of representatives leads to the greatest possible improvement in the objective function
3. If improvement is positive then replace $\overline{Y}$ with $\overline{X}$ and go to phase 2. Otherwise return current clusters $C_1, \ldots, C_k$

A solution to this is to use a randomly selected set of r pairs $(\overline{X}, \overline{Y})$, where $\overline{X} \in \mathscr{D}$ and $\overline{Y} \in \{\overline{Y}_1, \ldots, \overline{Y}_k\}$ and use the best of these pairs for the replacement. In this way we need to compute only r times the incremental objective function change.