A.Z
DATA MINING
DEC, 2023

lec 4 Clustering II

Abstract

K-clustering and Hierarchical clustering

1 K-means

K-means is only for numerical data (because of its reliance on the concept of a "centroid" and the calculation of distances between data points and centroids)

Notations: Data points $\mathbf{x}_i \in \mathcal{D}$, clusters $C_1, \ldots, C_K$, centroids $\mathbf{c}_1, \ldots, \mathbf{c}_k$, mean of data $\mathbf{m}$.

Objective: minimize $SSE = \sum_{j=1}^{K} \sum_{\mathbf{x} \in C_j} L_2^2(\mathbf{x}, \mathbf{c}_j)$

which means minimizes wc and maximizes bc since

$\sum_{\mathbf{x} \in \mathcal{D}} L_2^2(\mathbf{x}, \mathbf{m}) = \sum_{j=1}^{K} \sum_{\mathbf{x} \in C_j} L_2^2(\mathbf{x}, \mathbf{c}_j) + \sum_{j=1}^{K} |C_j| L_2^2(\mathbf{c}_j, \mathbf{m})$

Designed only for L_2 norm , but many K-representative variants for other distance measures

1.1 How to choose K?

1.1.1 SSE elbow

SSE decreases with K, there is a elbow in SSE curve, but not always clear

1.1.2 Silhouette

Silhouette tell how well an individual data point is clustered.

Silhouette of a point x is

$$S(\mathbf{x}) = \begin{cases} 0 & \text{if singleton} \\ \frac{b-a}{\max\{a,b\}} & \text{otherwise} \end{cases}$$

a = mean distance of x to points in the same cluster

b = mean distance of x to points in the closest neighbouring cluster

Average Silhouette: $S_{avg} = \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{D}} S(\mathbf{x})$

1.1.3 Calinski-Harabasz

Well suitable K-means, based on inter-cluster and intra-cluster variances

$$S_{CH} = \frac{(n-K)B}{(K-1)W}$$

between-cluster variance $B = \sum_{i=1}^{K} |C_i| \, L_2^2 \left(\mathbf{c}_i, \mathbf{m}\right)$, $\mathbf{m}$ is the mean of the whole data, the higher the better

within-cluster variance $W = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} L_2^2 \left(\mathbf{x}, \mathbf{c}_i\right)$, the lower the better

1.1.4 Gap statistic

Cluster data and evaluate $W_K = \sum_{r=1}^{K} \frac{1}{2|C_r|} \sum_{\mathbf{x}, \mathbf{y} \in C_r} d(\mathbf{x}, \mathbf{y})$

Evaluate W_K in B random data sets , $W_{K1}, \ldots, W_{KB}$

$\operatorname{Gap}(K) = \frac{1}{B} \sum_{b=1}^{B} \log \left(W_{Kb}\right) - \log \left(W_K\right)$

Choose $\min K : \operatorname{Gap}(K) \geq \operatorname{Gap}(K+1) - \sigma_{K+1}$

where $\sigma_K =$ standard deviation of $W_{K1}, \ldots, W_{KB}$

If $d = L_2^2, W_K$ estimates SSE

good: suits to any clustering method and distance d

bad: computationally heavy (B random simulations for all tested K)

paper: Estimating the number ofclusters in a data set via the gap statistic. Journal of the RoyalStatistical Society, 2001.

2 K-means extensions

2.1 k-medians

Idea: uses L1 measure and medians, determine median values along each dimension separately

$$S = \sum_{k=1}^{K} \sum_{x_{i \in c_k}} \mid x_{ij} - \text{ med }_{kj} \mid$$

good: more robust to outliers

bad: computationally more costly

2.2 K-medoids

Medoid is the center-most data point in a cluster, so medoids are actual data samples

Suits to any data type as long as given distance function

2.3 K-modes

For categorical data

Objective: minimize $\sum_{\mathbf{x} \in C} \sum_{i=1}^{k} d_s\left(x_i, c_i\right)$

where

$$d_s\left(x_i, y_i\right)=\begin{cases}1 & \text{if } x_i \neq y_i \\ 0 & \text{otherwise}\end{cases}$$

$video : https : //www.youtube.com/watch?v = b39_v ipRkUo$

2.3.1 K-prototypes

For mixed data

Objective: minimize $\sum_{\mathbf{x} \in C}\left(\sum_{i=1}^{q}\left(x_i-c_i\right)^2+\gamma \sum_{i=q+1}^{k} d_s\left(x_i, c_i\right)\right)$

where

$x_1, \ldots, x_q$ numerical values

$x_{q+1}, \ldots, x_k$ categorical values

$\gamma =$ balancing weight

cluster centroids c are 'prototypes'

2.4 Kernel-K-means

Idea: map data implicitely to a higher dimensional space and perform K-means there

Robust, can detect arbitrary shapes but expensive

3 Hierarchical clustering

video: https://www.youtube.com/watch?v=EUQY3hL38cw

Two ways

- agglomerative clustering (bottom up approach)

- divisive clustering (top down approach)

Approach:

Given D = intercluster distance (linkage metric)

Initialize distance matrix M

Repeat until termination

1. pick cloeset pair of cluster C_i and C_j where $D_{min}(C_i, C_j)$
2. merge clusters $C_{ij} = C_i \cup C_j$
3. update M

Some linkage metrics

Single	$\min_{\mathbf{x}_1 \in C_1, \mathbf{x}_2 \in C_2} \{d(\mathbf{x}_1, \mathbf{x}_2)\}$
Complete	$\max_{\mathbf{x}_1 \in C_1, \mathbf{x}_2 \in C_2} \{d(\mathbf{x}_1, \mathbf{x}_2)\}$
Average	$\frac{\sum_{\mathbf{x}_1 \in C_1, \mathbf{x}_2 \in C_2} d(\mathbf{x}_1, \mathbf{x}_2)}{\lvert C_1 \rvert \lvert C_2 \rvert}$
Minimum variance (Ward)	$-\mathrm{SSE}(C_1 \cup C_2) - \mathrm{SSE}(C_1)$
Distance of centroids	$d(\mathbf{c}_1, \mathbf{c}_2)$

Warning :

- Linkage metric has a strong effect on results.

- Most linkage metrics are sensitive to data order, which means results may change if you shuffle data

- Single linkage is prone to "chaining effect"

3.1 Connection to graph theory

Single linkage is related to connected components

Complete linkage is related to cliques

Single linkage:

1. Initialize: Create graph G without edges, all data points in their own clusters

2. Repeat until one connected componet
   1. add new edge e_i with smallest d_i to G
   2. form clusters from connected components of G

Complete linkage:

1. Initialize: Create graph G without edges, all data points in their own clusters
2. Repeat until one connected componet
   1. add new edge e_i with smallest d_i to G
   2. if two of the current clusters form a clique in G, merge them

3.1.1 Single linkage clustering from MST

Begin from complete distance graph G and search its minimum spanning tree (MST)

Repeat until all objects belong to one cluster:

1. Merge two clusters that are connected in the MST andhave the smallest edge weight
2. Set the edge weight as inf

3.2 Bisecting K-means

Idea: combine divisive hierarchical and K-means. Given K and q = number of iterations

1. Initialization: put all data points into one cluster
2. Repeat until K clusters:
   1. choose cluster C to split (with largest SSE)
   2. split C q times with 2-means
   3. keep the best split (two new clusters)

Both efficient (like K-means) and good results (comparable to hierarchical)