Single Linkage Clustering

Parismita Das

15 March 2017

Single Linkage Criterion

In Single Linlage Criterion of Agglomerative Hierarchical clustering, The distance between the elements of each cluster is minimised. As the Naive single linkage clustering has a worst case time complexity of O(N^3). It becomes useless for large data which are frequency used in the field of bioinformatics

Hence Nearest-neighbor chain algorithm is being used here

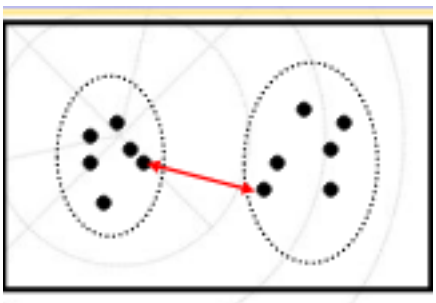
Nearest-neighbor chain algorithm

In the theory of cluster analysis, the nearest-neighbor chain algorithm is an algorithm that can speed up several methods for agglomerative hierarchical clustering. And Single Linkage Clustering is one of them.

The Algorithm

- Initialize the set of active clusters to consist of n one-point clusters, one for each input point.
- Let S be a stack data structure, initially empty, the elements of which will be active clusters.
- While there is more than one cluster in the set of clusters:
- If S is empty, choose an active cluster arbitrarily and push it onto S.
- Let C be the active cluster on the top of S. Compute the distances from C to all other clusters, and let D be the nearest other cluster.
- If D is already in S, it must be the immediate predecessor of C. Pop both clusters from S and merge them.
- Otherwise, if D is not already in S, push it onto S.

Here The cost function used is the minimum distance between the nearest cluster element.



As single linkage criterion can be implemented by using the Lance and William Algorithm

$$d(h,k) = \alpha_i d(i,k) + \alpha_j d(j,k) + \beta d(i,j) + \gamma \mid d(i,k) - d(j,k) \mid \qquad (1)$$

TABLE 2

Parameter Values for Combinatorial SAHN Clustering Methods

Clustering Method	α_i	α_j	β	γ
Single linkage (Nearest neighbor)	1/2	1/2	0	-1/2
Complete linkage (Furthest neighbor)	1/2	1/2	0	1/2
Group Average (UPGMA)	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	0	0
Weighted Average (WPGMA)	1/2	1/2	0	0
Unweighted Centroid (UPGMC)	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	$\frac{-n_i n_j}{(n_i+n_j)^2}$	0
Weighted Centroid (WPGMC)	1/2	1/2	-1/4	0
Minimum Variance (Ward)	$\frac{n_i+n_k}{n_i+n_j+n_k}$	$\frac{n_j+n_k}{n_i+n_j+n_k}$	$\frac{-n_k}{n_i+n_j+n_k}$	0
Flexible (Lance and Williams)	$\frac{1-\beta}{2}$	$\frac{1-\beta}{2}$	$\beta<1$	0

NOTE: n_i is the number of objects in cluster i.

Hence by tuning the cost function of FusionedClasses.cpp

```
FusionCost = 0.5 * (MyValue()  - abs(MyValue() - MyMatrix->Value(NextAvailableIndex,
    NextAvailableIndex)) + MyMatrix->Value(NextAvailableIndex, NextAvailableIndex));
```

The Output

```
h<-5
resP <- chaclust(s,h)
View(head(t(resP),10))
```

	V1	V2	V3
1	-41	-42	-3.4258040
2	-40	1	-11.0790827
3	2	-43	-17.3496360
4	-39	3	-37.4071973
5	-45	-46	-1.6078040
6	-82	-83	-0.8554707
7	-48	-49	-0.7090707
8	-65	-66	-0.2597373
9	-64	8	-0.8917493
10	7	-50	-0.1611373

References: https://users.cs.duke.edu/~edels/Papers/1984-J-05-HierarchicalClustering.pdf http://www.lx.it.pt/~afred/tutorials/B_Clustering_Algorithms.pdf http://orion.lcg.ufrj.br/Dr.Dobbs/books/book5/chap16.htm https://en.wikipedia.org/wiki/Nearest-neighbor_chain_algorithm