# Reading Summary IR: Ch 17

Xin Qian
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
xinq@cs.cmu.edu

## 1. CH 17

*Hierarchical clustering* obtains an informative hierarchy instead of the unstructured set of flat clusters. The disadvantage of hierarchical clustering is its quadratic complexity, as compared to the linear complexity of K-means and EM. Bottom-up hierarchical clustering algorithms (HAC) merge pairs of clusters until a final single cluster. Top-down clustering, however, splits clusters until only individual documents. *Dendrogram* represents each merge with a horizontal line. The y-coordinate value of the line shows the similarity between two recently merged clusters. We assume HAC has monotonic merge operation. Otherwise, we are not choosing the optimal merge at each step. We might want to cut the clustering process at some point. We might choose to cut when a pre-specified K is reached, at a pre-defined similarity threshold, when the similarity drastically decrease between two successive merge (a prespecified number of clusters of a certain clustering threshold) or K is argmin of the residual-sum-distortion equation. A simple HAC implementation maintains a N by N similarity matrix. For every merge, the matrix is updated with the similarity of every un-clustered cluster with the newly merged composite cluster.

*Single-link clustering* measures cluster similarity by two most similar documents in each cluster, which is local and might cause unwanted elongated/straggling/chaining clusters. *Complete-link clustering* measures cluster similarity by two most dissimilar documents in each cluster, which is non-local but sensitive/tolerant to outliers. The names of these two concepts can be interpreted with graph theory. The time complexity of a HAC algorithm is $O(N^3)$. A more efficient priority-queue implementation has a time complexity of $O(N^2 logN)$. Next-best-merge array (NBM) is another optimization for single-link HAC, since it's best-merge persistent. However, NBM is not suitable to other similarity measurements, e.g. best-merge persistence does support complete-link clustering. In practice, we treat all 4 HAC algorithms as $O(N^2)$ complexity.

*Group-average agglomerative clustering*, also called as average-link cluster, or GAAC works around the problems in single-link and complete-link HAC algorithms. It measures cluster average similarity based on all pairs of documents. SIM-GA is the equation to compute such average similarity. One character of GAAC is store documents as normalized vectors and use dot products for similarity computation. Merge operation for GAAC is the same as complete-link clustering. From the intuitive idea that a merged cluster quality can be the average similarity between documents and the cluster centroid, we might also want to revisit SIM-GA equation to include self-similarities in group-average similarity. However, this revision penalize large clusters and prefer small clusters since the proportion of self-similarities decreases as the cluster gets larger.

*Centroid clustering* measures the similarity of two clusters as the similarity between their centroids, which equals the average similarity of all pairs of documents from the two clusters. Centroid clustering is not monotonic. Increasing similarity breaks the underlining assumption that small clusters are more coherent than large clusters.

Optimality of a hierarchical clustering is reached when this clustering have a largest "smallest combination similarity" among all possible clustering with cluster number either equal or smaller than this clustering. Centroid clustering is not optimal. For single-link, a cluster combination similarity is the smallest similarity between any possible bipartition of the cluster. For complete-link, it is the smallest similarity between any two points. For GAAC, it is the average of all pairwise similarities in the cluster. The section then proves the optimality of single-link clustering through induction. Complete-link clustering and GAAC are not optimal but favour sphericity.

*Divisive clustering* splits cluster recursively using a flat clustering algorithm, until each document has its own singleton cluster. It needs a flat clustering algorithm as a "subroutine" but becomes more efficient if we do not need to go deep to leaves. Evidence also show it's more accurate. *Differential cluster labeling* labels clusters by the term distribution from one cluster to others. *Clusrer-internal labeling* labels cluster merely through its cluster specialty, regardless of other clusters. There are several complications for hierarchical cluster labeling.

We might benefit from an inverted index on dot product computation. We might want to avoid dense centroids when using GAAC. Buckshot algorithm guarantees linearity computation cost with determinism and reliability of HAC as well as efficiency of K-means.



*11741'F16 Pittsburgh, Pennsylvania USA*



DOI: 10.1145/1235