Chapter 7 - Text Clustering

Jianzhang Zhang

Alibaba Business School
Hangzhou Normal University

May 25, 2022

Table of Contents

Table of Contents

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 analysis.

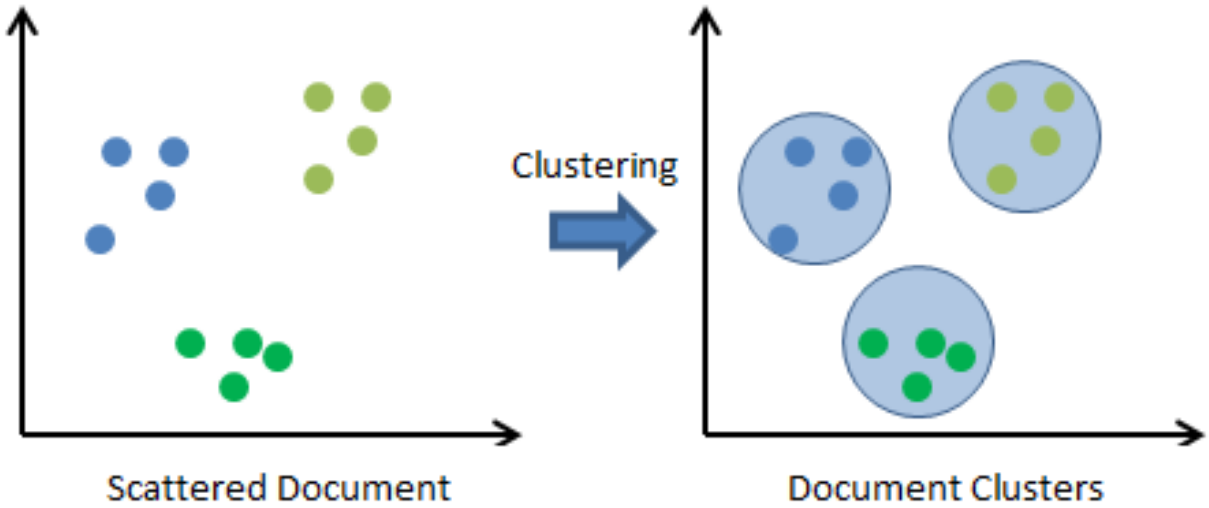


Figure 1: Text Clustering

Hierarchical clustering outputs a hierarchy, a structure that is more informative than the unstructured set of clusters returned by flat clustering (e.g., K-means, EM).

Hierarchical Agglomerative Clustering

Hierarchical clustering algorithms are either top-down or bottom-up. Bottom-up algorithms treat each document as a singleton cluster at the outset and then successively merge (or agglomerate) pairs of clusters until all clusters have been merged into a single cluster that contains all documents. Top-down clustering requires a method for splitting a cluster.

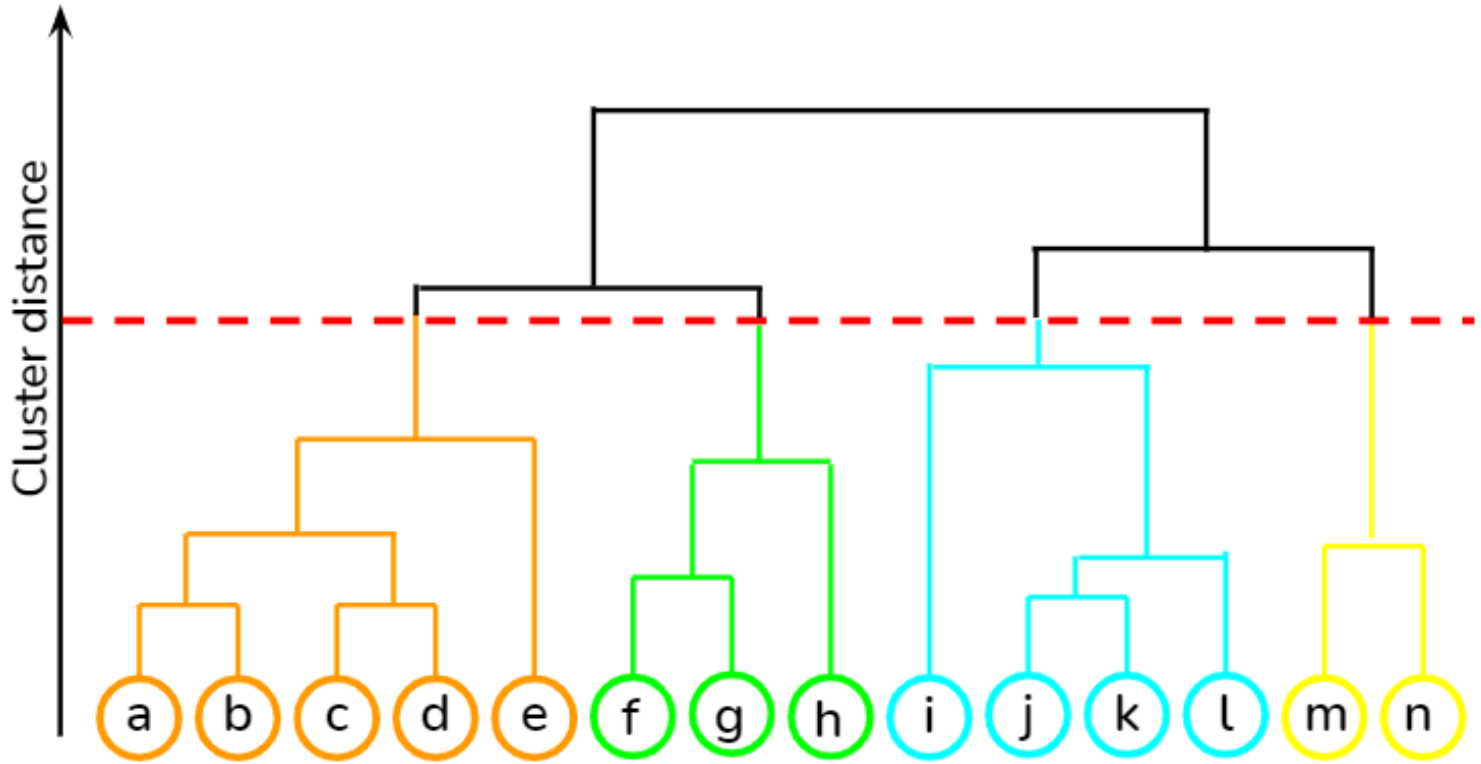
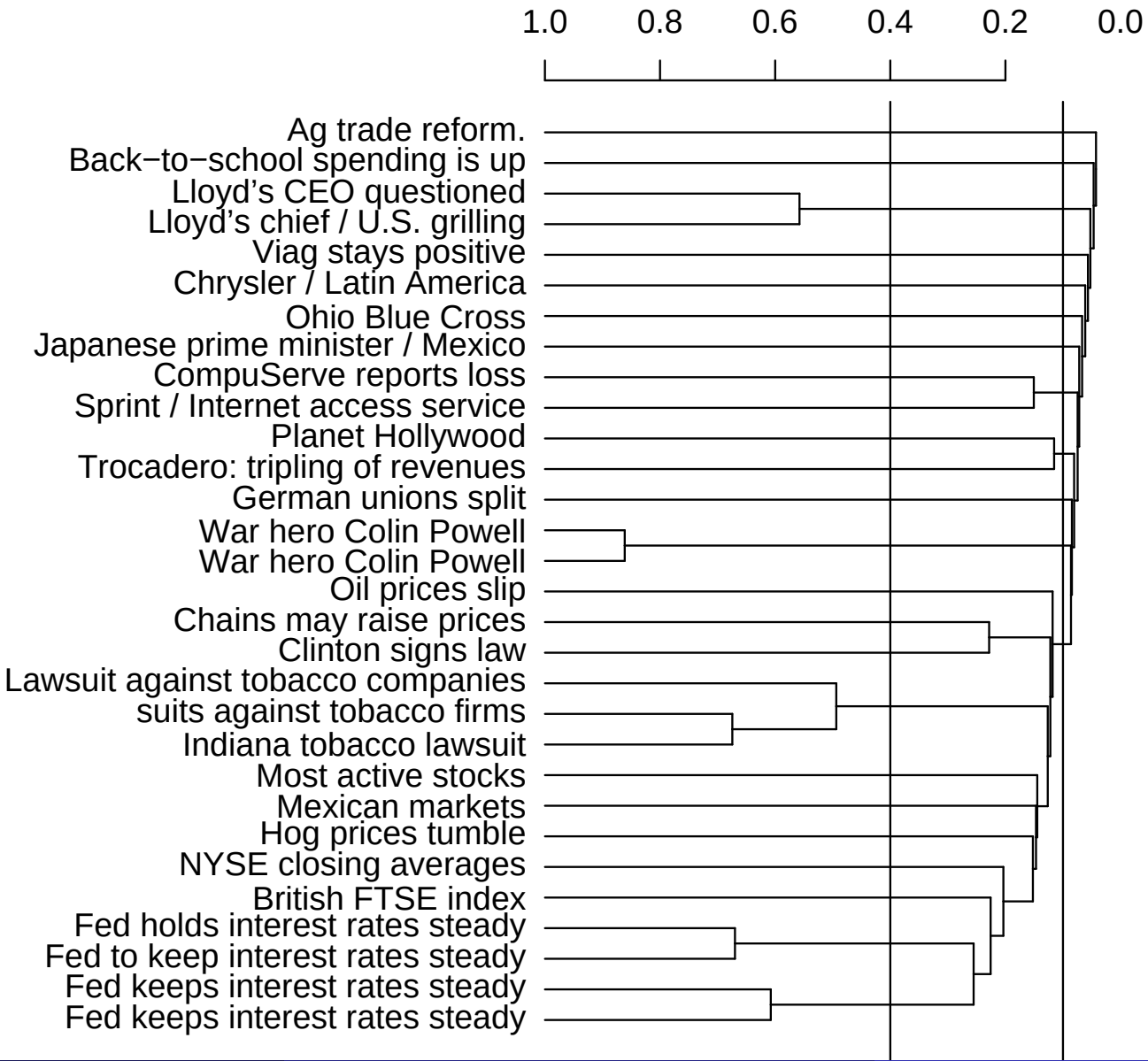


Table of Contents

Dendrogram


1.0
0.8
0.6
0.4
0.2
0.0
Ag trade reform.
Back−to−school spending is up
Lloyd's CEO questioned
Lloyd's chief / U.S. grilling
Viag stays positive
Chrysler / Latin America
Ohio Blue Cross
Japanese prime minister / Mexico
CompuServe reports loss
Sprint / Internet access service
Planet Hollywood
Trocadero: tripling of revenues
German unions split
War hero Colin Powell
War hero Colin Powell
Oil prices slip
Chains may raise prices
Clinton signs law
Lawsuit against tobacco companies
suits against tobacco firms
Indiana tobacco lawsuit
Most active stocks
Mexican markets
Hog prices tumble
NYSE closing averages
British FTSE index
Fed holds interest rates steady
Fed to keep interest rates steady
Fed keeps interest rates steady
Fed keeps interest rates steady

Dendrogram: An HAC clustering is typically visualized as a dendrogram. Each merge is represented by a horizontal line. The y-coordinate of the horizontal line is the similarity of the two clusters that were merged, where documents are viewed as singleton clusters.

Combination Similarity: We call this similarity the combination similarity of the merged cluster. For example, the combination similarity of the cluster consisting of `Lloyd's CEO questioned` and `Lloyd' s chief/U.S. grilling` in the above Figure is ≈ 0.56.

Process: By moving up from the bottom layer to the top node, a dendrogram allows us to reconstruct the history of merges that resulted in the depicted clustering. For example, we see that the two documents entitled `War hero Colin Powell` were merged first in the above Figure and that the last merge added `Ag trade reform` to a cluster consisting of the other 29 documents.

Monotonic: A fundamental assumption in HAC is that the merge operation is monotonic. Monotonic means that if $s_1, s_2, ..., s_{K-1}$ are the combination similarities of the successive merges of an HAC, then $s_1 \geq s_2 \geq ... \geq s_{K-1}$ holds.

Cutting point: Hierarchical clustering does not require a prespecified number of clusters. However, in some applications we want a partition of disjoint clusters just as in flat clustering. In those cases, the hierarchy needs to be cut at some point (cutting point).

The Number of Clusters

1. **Cut at a prespecified level of similarity**: For example, we cut the dendrogram at 0.4 if we want clusters with a minimum combination similarity of 0.4. In the above Figure, cutting the diagram at $y = 0.4$ yields 24 clusters (grouping only documents with high similarity together) and cutting it at $y = 0.1$ yields 12 clusters (one large financial news cluster and 11 smaller clusters).
2. **Cut the dendrogram where the gap between two successive combina similarities is largest**: Such large gaps arguably indicate "natural" clusterings. Adding one more cluster decreases the quality of the clustering significantly, so cutting before this steep decrease occurs is desirable.

3. **Minimum RSS**:

$$K = \operatorname*{argmin}_{K'} [RSS(K') + \lambda K']$$

RSS is the residual sum of squares and λ is a penalty for each additional cluster. Instead of RSS, another measure of distortion (失真) can be used.

4. As in flat clustering, we can also prespecify the number of clusters K and select the cutting point that produces K clusters.

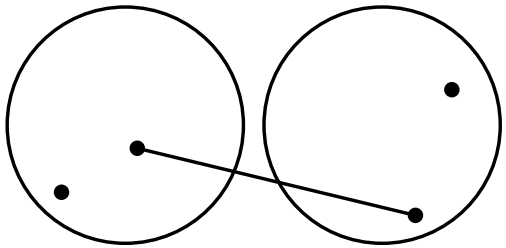
Table of Contents

SIMPLEHAC$(d_1, \ldots, d_N)$

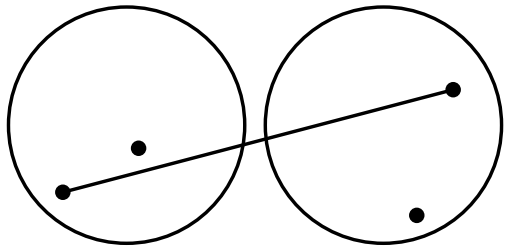
1 **for** $n \leftarrow 1$ **to** N
2 **do for** $i \leftarrow 1$ **to** N
3 **do** $C[n][i] \leftarrow \text{SIM}(d_n, d_i)$
4 $I[n] \leftarrow 1$ *(keeps track of active clusters)*
5 $A \leftarrow []$ *(assembles clustering as a sequence of merges)*
6 **for** $k \leftarrow 1$ **to** $N - 1$
7 **do** $\langle i, m \rangle \leftarrow \arg\max_{\{\langle i,m \rangle : i \neq m \wedge I[i]=1 \wedge I[m]=1\}} C[i][m]$
8 $A.\text{APPEND}(\langle i, m \rangle)$ *(store merge)*
9 **for** $j \leftarrow 1$ **to** N
10 **do** $C[i][j] \leftarrow \text{SIM}(i, m, j)$
11 $C[j][i] \leftarrow \text{SIM}(i, m, j)$
12 $I[m] \leftarrow 0$ *(deactivate cluster)*
13 **return** A

- We first compute the $N \times N$ similarity matrix C (line 1 - 3). The algorithm then executes $N - 1$ steps of merging the currently most similar clusters (line 6 - 7).
- In each iteration, the two most similar clusters are merged and the rows and columns of the merged cluster i in C are updated (line 9 - 11).
- The clustering is stored as a list of merges in A (line 5, 8). I indicates which clusters are still available to be merged (line 4, 12).
- The function $SIM(i, m, j)$ computes the similarity of cluster j with the merge of clusters i and m (line 10 - 11).

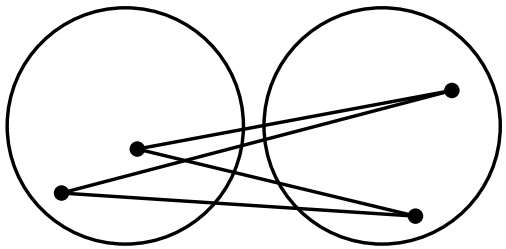
The **different notions of cluster similarity** used by the four HAC algorithms. An inter-similarity is a similarity between two documents from different clusters.



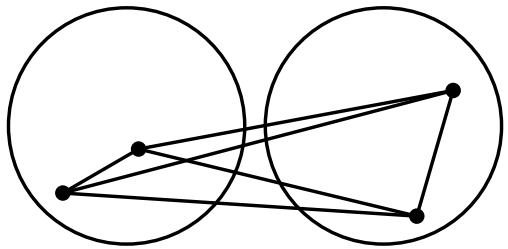
(a) single-link: maximum similarity



(b) complete-link: minimum similarity



(c) centroid: average inter-similarity

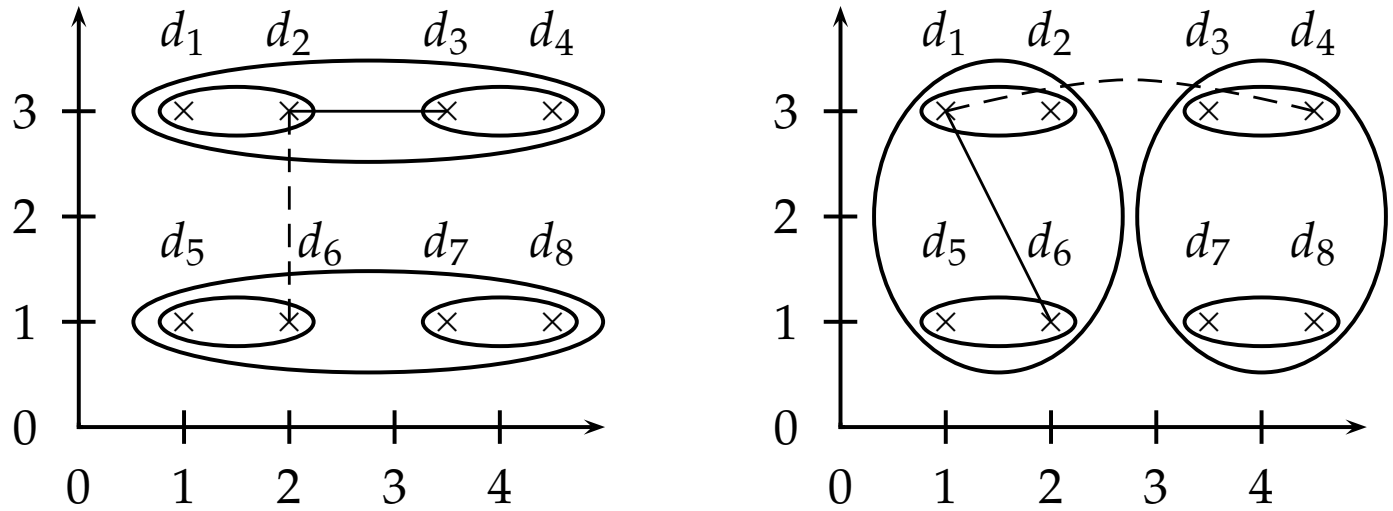


(d) group-average: average of all similarities

Table of Contents

In **single-link clustering**, the similarity of two clusters is the similarity of their most similar members (min min-distance). This merge criterion is local. We pay attention solely to the area where the two clusters come closest to each other. Other, more distant parts of the cluster and the clusters' overall structure are not taken into account.

In **complete-link clustering**, the similarity of two clusters is the similarity of their most dissimilar members (min max-distance). This merge criterion is non-local; the entire structure of the clustering can influence merge decisions. This results in a preference for compact clusters with small diameters over long, straggly clusters, but also causes sensitivity to outliers.



The first four steps, each producing a cluster consisting of a pair of two documents, are identical, then:

Single-link clustering (left) joins the upper two pairs (and after that the lower two pairs).

Complete-link clustering (right) joins the left two pairs (and then the right two pairs).

Drawbacks of Single/Complete link Clustering

Single-link and complete-link clustering reduce the assessment of cluster quality to a single similarity between a pair of documents. A measurement based on one pair cannot fully reflect the distribution of documents in a cluster.

It is therefore not surprising that **both algorithms often produce undesirable clusters**.

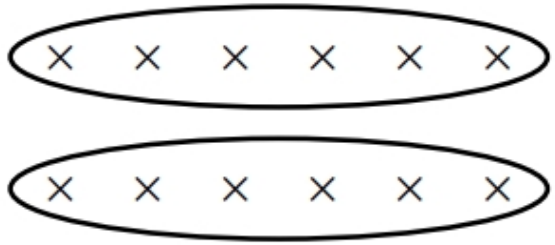


Figure 3: *Chaining* effect of single-link clustering

Since the merge criterion is strictly local, a chain of points can be extended for long distances without regard to the overall shape of the emerging cluster.

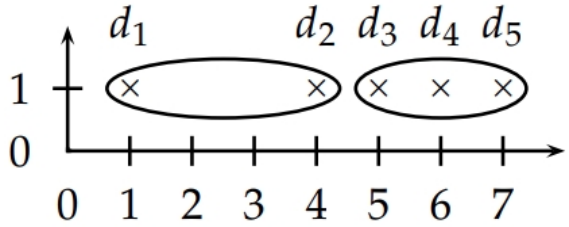
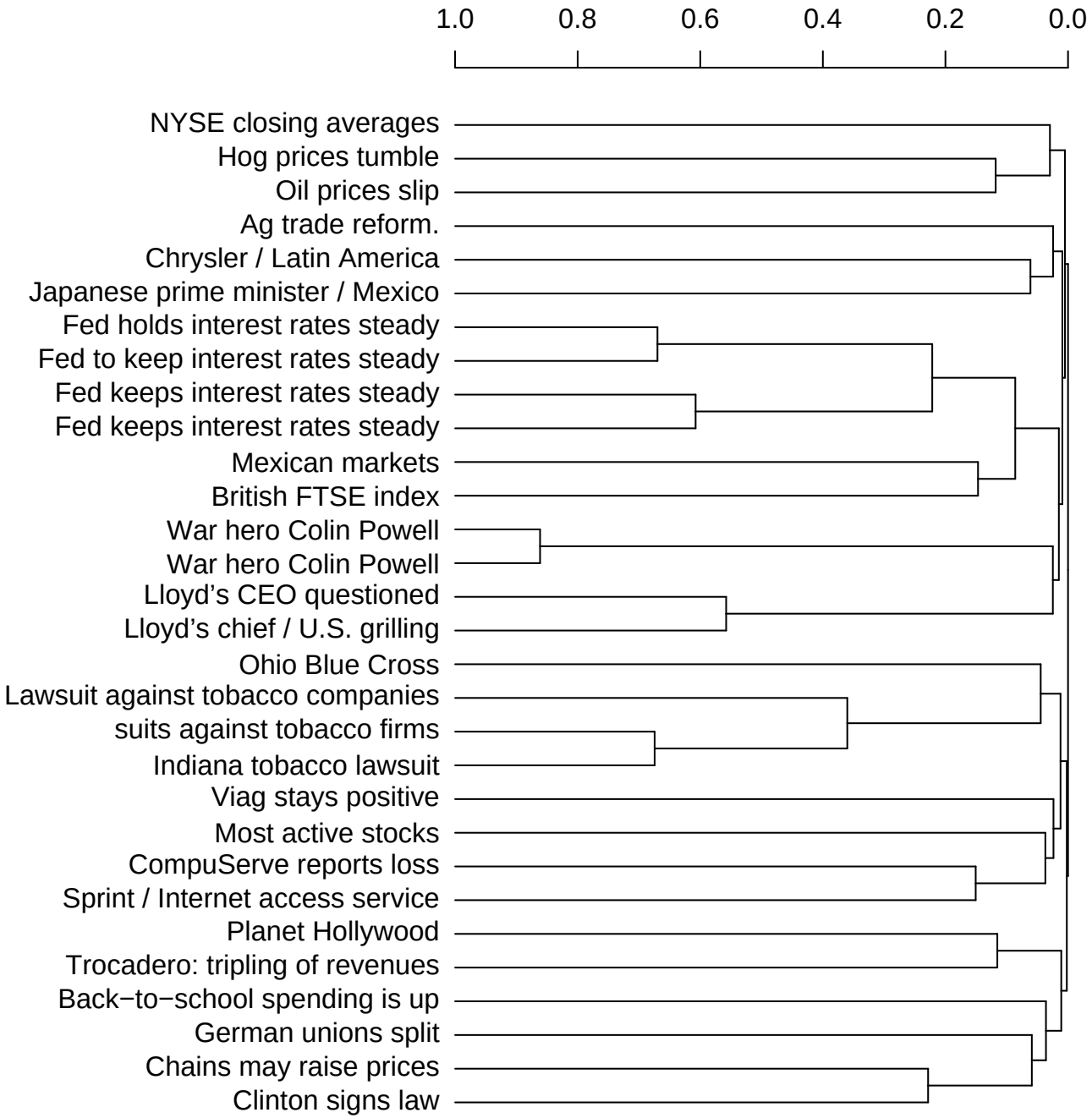


Figure 4: Outliers in complete-link clustering

The five documents have the x-coordinates $1 + 2\epsilon$, 4, $5 + 2\epsilon$, 6 and $7 - \epsilon$. Complete-link clustering creates the two clusters shown as ellipses.

4. Single-link and Complete-link Clustering



In the first dendrogram, the last eleven merges of the single-link clustering (those above the 0.1 line) add on single documents or pairs of documents, corresponding to a chain.

In the second dendrogram, the complete-link clustering avoids this problem. Documents are split into two groups of roughly equal size when we cut the dendrogram at the last merge. In general, this is a more useful organization of the data than a clustering with chains.

Table of Contents

Group-average agglomerative clustering (average-link clustering) (GAAC) evaluates cluster quality based on all similarities between documents, thus avoiding the pitfalls of the single-link and complete-link criteria, which equate cluster similarity with the similarity of a single pair of documents.

GAAC computes the average similarity $SIM\text{-}GA$ of all pairs of documents, including pairs from the same cluster. But self-similarities are not included in the average.

$$SIM\text{-}GA(w_i, w_j) = \frac{1}{(N_i + N_j)(N_i + N_j - 1)} \sum_{d_m \in w_i \cup w_j} \sum_{d_n \in w_i \cup w_j, d_n \neq d_m} \vec{d_m} \cdot \vec{d_n}$$

where $\vec{d}$ is the length-normalized vector of document d, $\cdot$ denotes the dot product, and N_i and N_j are the number of documents in w_i and w_j, respectively. The dot product of two length-normalized vectors is equal to their cosine similarity.

The motivation for GAAC is that our goal in selecting two clusters w_i and w_j as the next merge in HAC is that the resulting merge cluster $w_k = w_i \cup w_j$ should be coherent. To judge the coherence of w_k, we need to look at all document-document similarities within w_k, including those that occur within w_i and those that occur within w_j.

We can compute the measure $SIM\text{-}GA$ efficiently because the sum of individual vector similarities is equal to the similarities of their sums.

$$\sum_{d_m \in w_i} \sum_{d_n \in w_j} (\vec{d_m} \cdot \vec{d_n}) = (\sum_{d_m \in w_i} \vec{d_m}) \cdot (\sum_{d_n \in w_j} \vec{d_n})$$

With the above equation, we have:

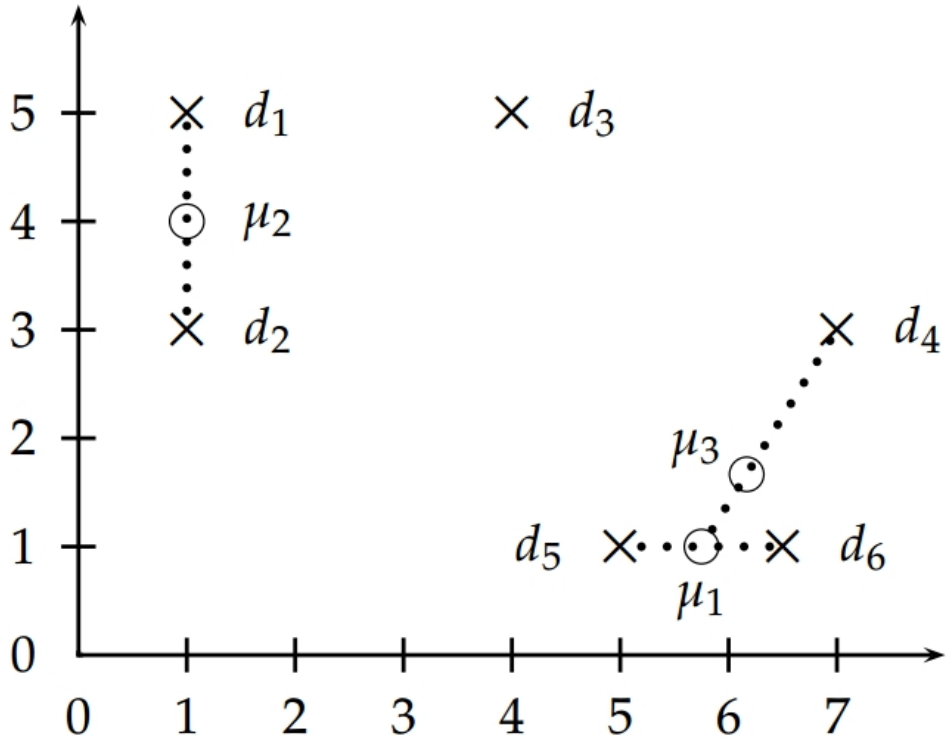
$$SIM\text{-}GA(w_i, w_j) = \frac{1}{(N_i + N_j)(N_i + N_j - 1)} [(\sum_{d_m \in w_i \cup w_j} \vec{d_m})^2 - (N_i + N_j)]$$

The term $(N_i + N_j)$ on the right is the sum of $N_i + N_j$ self-similarities of value 1.0.

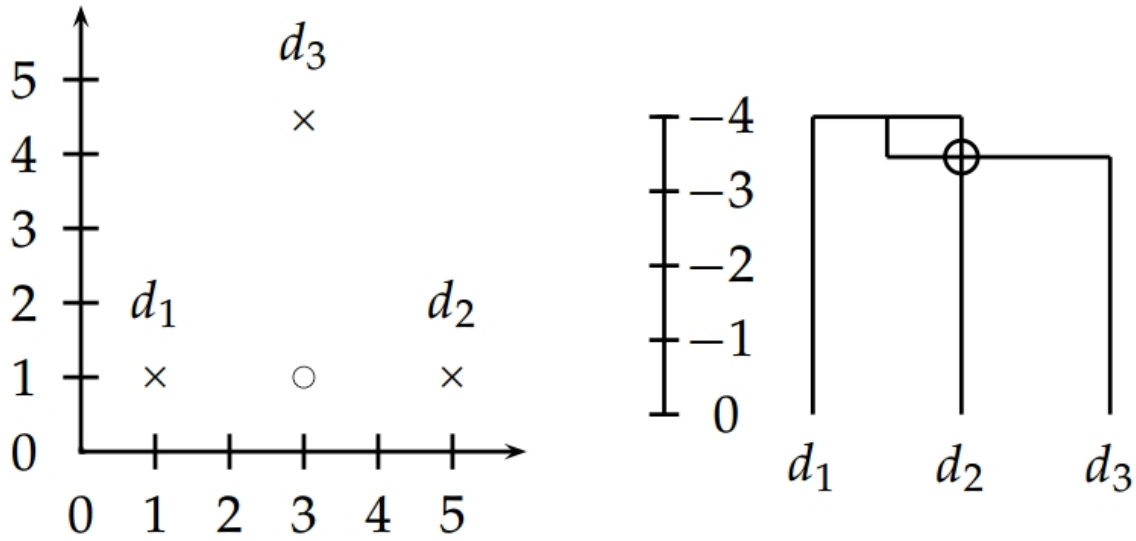
In centroid clustering, the similarity of two clusters is defined as the similarity of their centroids:

$$\begin{aligned} SIM\text{-}CENT(w_i, w_j) &= \vec{\mu}(w_i) \cdot \vec{\mu}(w_j) \\ &= (\frac{1}{N_i} \sum_{d_m \in w_i} \vec{d_m}) \cdot (\frac{1}{N_j} \sum_{d_n \in w_j} \vec{d_n}) \\ &= \frac{1}{N_i N_j} \sum_{d_m \in w_i} \sum_{d_n \in w_j} \vec{d_m} \cdot \vec{d_n} \end{aligned}$$

The centroid similarity is equivalent to average similarity of all pairs of documents from different clusters. Thus, the difference between GAAC and centroid clustering is that GAAC considers all pairs of documents in computing average pairwise similarity whereas centroid clustering excludes pairs from the same cluster.

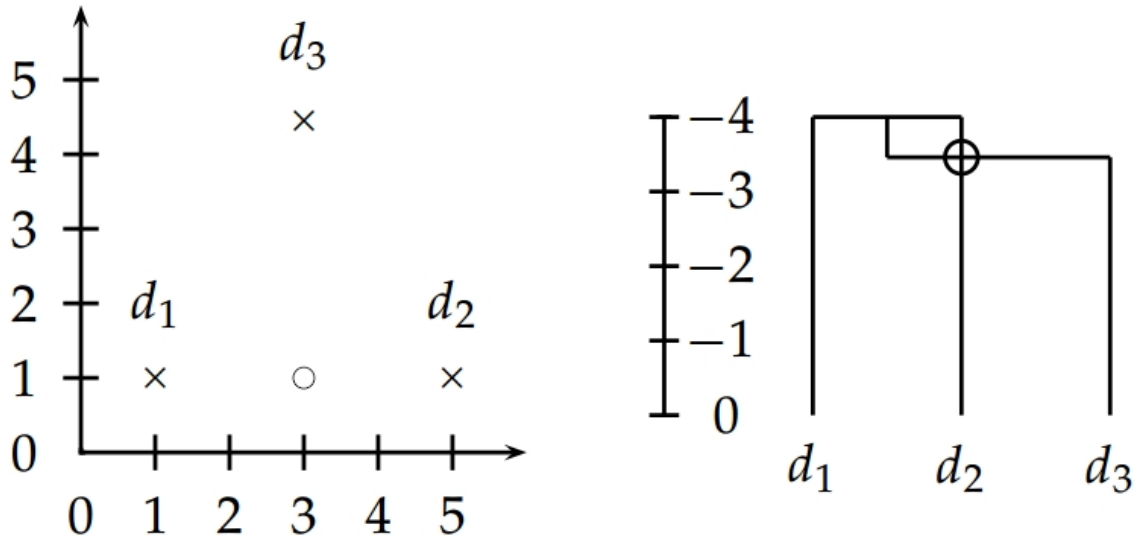


The first two iterations form the clusters $\{d_5, d_6\}$ with centroid μ_1 and $\{d_1, d_2\}$ with centroid μ_2 because the pairs $< d_5, d_6 >$ and $< d_1, d_2 >$ have the highest centroid similarities. In the third iteration, the highest centroid similarity is between μ_1 and μ_4 producing the cluster $\{d_4, d_5, d_6\}$ with centroid μ_3.



Centroid clustering is not monotonic. The documents d_1 at $(1+\epsilon, 1)$, d_2 at $(5, 1)$, and d_3 at $(3, 1+2\sqrt{3})$ are almost equidistant, with d_1 and d_2 closer to each other than to d_3.

We define similarity as negative distance. In the first merge, the similarity of d_1 and d_2 is $-(4-\epsilon)$. In the second merge, the similarity of the centroid of d_1 and d_2 (the circle) and d_3 is $\approx -cos(\pi/6) \times 4 = -\sqrt{3}/2 \times 4 \approx -3.46 > -(4-\epsilon)$. This is an example of an inversion: similarity increases in this sequence of two clustering steps.



The non-monotonic inversion in the hierarchical clustering of the three points appears as an intersecting merge line in the dendrogram. The intersection is circled.

Despite its non-monotonicity, centroid clustering is often used because its similarity measure (the similarity of two centroids) is conceptually simpler than the average of all pairwise similarities in GAAC.

Table of Contents

Ward's method says that the distance between two clusters, A and B, is how much the sum of squares (离差平方和) will increase when we merge them:

$$\begin{aligned}\Delta &= \sum_{i \in A \cup B} ||\vec{x}_i - \vec{m}_{A \cup B}||^2 - \sum_{i \in A} ||\vec{x}_i - \vec{m}_A||^2 - \sum_{i \in B} ||\vec{x}_i - \vec{m}_B||^2 \\ &= \frac{n_A n_B}{n_A + n_B} ||\vec{m}_A - \vec{m}_B||^2\end{aligned}$$

where $\vec{m}_j$ is the center of cluster j, and n_j is the number of points in it. Δ is called the **merging cost** of combining the clusters A and B.

With hierarchical clustering, the sum of squares starts out at zero (because every point is in its own cluster) and then grows as we merge clusters. Ward's method keeps this growth as small as possible. Given two pairs of clusters whose centers are equally far apart, Ward's method will prefer to merge the smaller ones.

THE END