Chapter 6
Text Clustering

6.1 Text Similarity Measures

Different clustering algorithms can produce different results by adopting different perspectives, but almost all of them are performed based on similarity measures. Therefore, the key problem of text clustering is how to effectively measure the similarity of texts.

In text clustering, a cluster is represented by a collection of similar documents, and there are three main types of text similarities:

- Similarity between two documents;[1]
- Similarity between two document collections;
- Similarity between a document and a document collection.

We will introduce the three kinds of similarity measures below.

6.1.1 The Similarity Between Documents

(1) Distance-Based Similarity

In a vector space model, a document is represented as a vector in the vector space. The simplest way to measure document similarity is to use the distance between two vectors in vector space. The smaller the distance between two vectors, the higher the similarity of the two documents. The commonly used distance metrics include Euclidean distance, Manhattan distance, Chebyshev distance, Minkowski distance, Mahalanobis distance, and Jaccard distance.

[1]For the simplicity of description, we use "document" to refer to a piece of text at different levels (e.g., sentence, document, etc.).

C. Zong et al., *Text Data Mining*, https://doi.org/10.1007/978-981-16-0100-2_6

Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be the vector representations of two documents, and the following distances are defined as follows.

a. Euclidean distance

$$d\left(\boldsymbol{a}, \boldsymbol{b}\right) = \left(\sum_{k=1}^{M}(a_k - b_k)^2\right)^{1/2} \tag{6.1}$$

b. Manhattan distance

$$d(\boldsymbol{a}, \boldsymbol{b}) = \sum_{k=1}^{M}|a_k - b_k| \tag{6.2}$$

c. Chebyshev distance

$$d(\boldsymbol{a}, \boldsymbol{b}) = \max_{k}|a_k - b_k| \tag{6.3}$$

d. Minkowski distance

$$d(\boldsymbol{a}, \boldsymbol{b}) = \left(\sum_{k=1}^{M}(a_k - b_k)^p\right)^{1/p} \tag{6.4}$$

(2) Cosine Similarity

Cosine similarity computes the similarity between two vectors by calculating the cosine of the angle between the two vectors:

$$\cos(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{b}}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|} \tag{6.5}$$

The range of cosine similarity is $[-1, 1]$. The smaller the angle between the two vectors is, the higher the cosine similarity. When the angle between two vectors is 0° (i.e., the same direction), the cosine similarity is 1; when the angle between two vectors is 90° (i.e., orthogonal direction), the cosine similarity is 0; when the angle between two vectors is 180° (i.e., opposite direction), the cosine similarity is -1.

The inner product of two vectors is proportional to the cosine similarity. The inner product of two vectors after L-2 normalization (see Chap. 3) is equivalent to the cosine similarity: $\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}^{\mathrm{T}}\boldsymbol{b}$ (Fig. 6.1).

Distance-based similarity measures the absolute distance between two vectors in the vector space. Cosine similarity measures the angle of vectors in the vector space and is the most widely used method for measuring the similarity of texts.

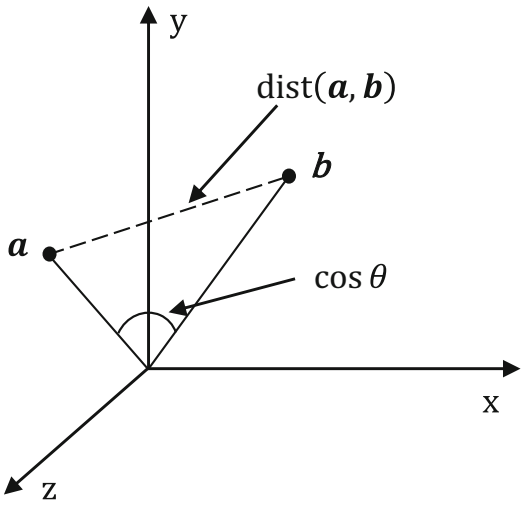


Fig. 6.1 Distance measurement samples in the vector space model

(3) Distribution-Based Similarity

The previous two kinds of similarity measures are performed based on the vector space. However, a document is sometimes represented by a distribution rather than a vector space model, especially in generative models. In this case, the statistical distance can be used to measure the similarity between two documents.

Statistical distance measures the difference between two distributions. A commonly used metric is the Kullback–Leibler (K-L) distance (also called K-L divergence). Based on the BOW assumption, a document can be represented by a categorical distribution over terms. Suppose P and Q are two categorical distributions, and the K-L distance of P and Q is defined as

$$D_{\mathrm{KL}}(P\|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{6.6}$$

The K-L distance is not symmetrical, that is, $D_{\mathrm{KL}}(P\|Q) \neq D_{\mathrm{KL}}(Q\|P)$. A symmetrical K-L distance can therefore be used instead:

$$D_{\mathrm{SKL}}(P, Q) = D_{\mathrm{KL}}(P\|Q) + D_{\mathrm{KL}}(Q\|P) \tag{6.7}$$

It is worth noting that when a document is of short length, it is meaningless to use a categorical distribution to represent it and use the K-L distance to measure the similarity of two documents. In fact, such distribution-based metrics are more suitable for measuring the similarity between two collections of texts than that between two short pieces of texts.

(4) Other Measures

There are other methods for similarity measures. For example, the Jaccard similarity coefficient is another widely used metric that measures the similarity between two sets; it is defined as the size of the intersection divided by the size of the union of the two sets:

$$J\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \frac{|\boldsymbol{x}_i \cap \boldsymbol{x}_j|}{|\boldsymbol{x}_i \cup \boldsymbol{x}_j|} \tag{6.8}$$

where a document is represented by a set of words.

Note that the aforementioned similarity measures can be used not only in text clustering but also in other text data mining tasks.

6.1.2 The Similarity Between Clusters

A cluster is a collection of similar documents. The similarity between two clusters can be computed based on the similarities of the documents contained in them. Suppose $d(C_m, C_n)$ denotes the distance between clusters C_m and C_n, $d(\boldsymbol{x}_i and \boldsymbol{x}_j)$ denotes the distance between documents $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. There are several ways to measure the similarity between the two clusters as follows.

(1) A single linkage denotes the shortest distance between two documents extracted from two clusters respectively:

$$d(C_m, C_n) = \min_{\boldsymbol{x}_i \in C_m, \boldsymbol{x}_j \in C_n} d(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{6.9}$$

(2) A complete linkage denotes the longest distance between two documents extracted from two clusters respectively:

$$d(C_m, C_n) = \max_{\boldsymbol{x}_i \in C_m, \boldsymbol{x}_j \in C_n} d(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{6.10}$$

(3) The average linkage denotes the average distance between two documents extracted from two clusters respectively:

$$d(C_m, C_n) = \frac{1}{|C_m| \cdot |C_n|} \sum_{\boldsymbol{x}_i \in C_m} \sum_{\boldsymbol{x}_j \in C_n} d(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{6.11}$$

(4) The centroid method is the distance between the centroid of two clusters:

$$d(C_m, C_n) = d(\bar{\boldsymbol{x}}(C_m), \bar{\boldsymbol{x}}(C_n)) \tag{6.12}$$

where $\bar{\boldsymbol{x}}(C_m)$ and $\bar{\boldsymbol{x}}(C_n)$ denote the centroids of the clusters C_m and C_n, respectively.

(5) Ward's method. For each cluster, we first define the within-cluster variance as the sum of squares of the distance between each document and the cluster centroid. The increase in total within-cluster variance after merging the two clusters can therefore be used as a cluster distance metric:

$$\begin{aligned} d(C_m, C_n) = & \sum_{\boldsymbol{x}_k \in C_m \cup C_n} d(\boldsymbol{x}_k, \bar{\boldsymbol{x}}(C_m \cup C_n)) \\ & - \sum_{\boldsymbol{x}_i \in C_m} d(\boldsymbol{x}_i, \bar{\boldsymbol{x}}(C_m)) - \sum_{\boldsymbol{x}_j \in C_n} d(\boldsymbol{x}_j, \bar{\boldsymbol{x}}(C_n)) \end{aligned} \tag{6.13}$$

where $d(\boldsymbol{a}, \boldsymbol{b}) = \|\boldsymbol{a} - \boldsymbol{b}\|^2$.

Ward's method is a criterion applied in hierarchical clustering. It minimizes the total within-cluster variance by finding the pair of clusters at each step that leads to a minimum increase in total within-cluster variance after they are merged.

In addition to the five abovementioned methods, the K-L divergence can also be used for calculating the distance between two clusters. The equation for the K-L divergence is shown as Eq. (6.6). The difference is that the categorical distributions P and Q are estimated by a cluster rather than a document.

6.2 Text Clustering Algorithms

There are extensive types of text clustering methods, including partition-based methods, hierarchy-based methods, density-based methods, grid-based methods, and graph-based methods, each of which contains some typical algorithms. In the following, we introduce several representative text clustering algorithms.

6.2.1 K-Means Clustering

The K-means algorithm, proposed by MacQueen in 1967, is a widely used partition-based clustering algorithm.

For a given dataset $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, the goal of K-means clustering is to divide the N samples into K ($K \leq N$) clusters to minimize the sum of the squared distances within each cluster, which is called the within-cluster sum of

squares (WCSS):

$$\arg\min_{C} \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in C_k} \|\boldsymbol{x} - \boldsymbol{m}_k\|^2 \tag{6.14}$$

To achieve this objective, the standard K-means clustering algorithm (also called the Lloyd–Forgy method) uses the iterative optimization method. In each iteration step, the distances between each sample and the K centroids (i.e., the means) of the cluster are first calculated. The samples are then assigned to the clusters with the nearest centroid, and the centroids of existing clusters are updated. This process is repeated until the minimum WCSS is reached.

Formally, given the initial centroids of the K clusters $\boldsymbol{m}_1^{(0)}, \boldsymbol{m}_2^{(0)}, \ldots, \boldsymbol{m}_K^{(0)}$, the algorithm iterates in the following two steps:

(1) Assignment: Assign each sample into the cluster that minimizes the sum of squares within clusters:

$$C^{(t)}(\boldsymbol{x}_i) = \arg\min_{k=1,\ldots,K} ||\boldsymbol{x}_i - \boldsymbol{m}_k^{(t-1)}||^2 \tag{6.15}$$

where t denotes the steps of the iterations and $C(\boldsymbol{x})$ denotes the index of the cluster to which $\boldsymbol{x}$ is assigned.

(2) Updating: Update the centroids for each of the K clusters:

$$\boldsymbol{m}_k^{(t+1)} = \frac{1}{|C_k^{(t)}|} \sum_{\boldsymbol{x}_i \in C_k^{(t)}} \boldsymbol{x}_i \tag{6.16}$$

The two steps are iteratively performed until the algorithm converges to a local minimum. But such an alternated iterative optimization cannot guarantee the global minimum of the WCSS.

In practice, we can also choose different distance metrics. For example, in text clustering, the cosine similarity is more often used:

$$d\left(\boldsymbol{x}, \boldsymbol{m}_k^{(t)}\right) = \frac{\boldsymbol{x} \cdot \boldsymbol{m}_k^{(t)}}{\|\boldsymbol{x}\| \left\|\boldsymbol{m}_k^{(t)}\right\|} \tag{6.17}$$

However, it should be noted that the above iterative optimization can ensure the decrease in WCSS only under the Euclidean distance metric. If different distance metrics are used, there is a risk that the algorithm may not converge.

In summary, the K-means clustering algorithm is described as follows.

Table 6.1 displays a small text clustering dataset that contains ten short documents extracted from the domains of education, sports, technology, and literature.

Algorithm 1: K-means clustering algorithm

Input : dataset $\mathscr{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, number of clusters K;
Output: clusters $\{C_1, C_2, \ldots, C_K\}$.

```
Randomly select K samples in 𝒟 as the initial mean vectors {m_1, m_2, ..., m_K};
while not converged do
    for i = 1, ..., N do
        for k = 1, ..., K do
            calculate the distance d(x_i, m_k) = ||x_i − m_k||^2 between x_i and m_k;
        end
        divide sample x_i into the cluster of nearest mean vector arg min_k {d(x_i, m_k)}
    end
    for i = 1, ..., K do
        update the mean vector of each cluster: m_k^new = 1/|C_k| Σ_{x_i ∈ C_k} x_i.
    end
end
```

Table 6.1 Text clustering dataset

ID	Sentence
$\boldsymbol{x}_1$	Beijing Institute of Technology was established in 1958 as one of the earliest universities that established a computer science major in China.
$\boldsymbol{x}_2$	Students from Beijing Institute of Technology won the 4th China Computer Go Championship.
$\boldsymbol{x}_3$	The Gymnasium of Beijing Institute of Technology is the venue for the preliminary volleyball competition of the 2008 Beijing Olympic Games in China.
$\boldsymbol{x}_4$	In the 5th East Asian Games, the total number of medals of China reached a new high. Both the men's and women's volleyball teams won championships.
$\boldsymbol{x}_5$	Artificial intelligence, also known as machine intelligence, refers to the intelligence represented by an artificially produced system.
$\boldsymbol{x}_6$	Artificial intelligence is a branch of computer science that attempts to produce an intelligent machine that can react in a manner similar to human intelligence.
$\boldsymbol{x}_7$	The three Go competitions between artificial intelligence AlphaGo and human champion Jie Ke end with the human's thorough defeat.
$\boldsymbol{x}_8$	The first sparrow of spring! The year beginning with youngest hope than ever!
$\boldsymbol{x}_9$	The brooks sing carols and glees to the spring. The symbol of youth, the grass blade, like a long green ribbon, streams from the sod into the summer.
$\boldsymbol{x}_{10}$	The grass flames up on the hillsides like a spring fire, not yellow but green is the color of its flame.

Let $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{10}\}$ denote this clustering dataset, in which $\boldsymbol{x}_i$ corresponds to the i-th document. Before text clustering, we first perform feature selection. The dataset includes 118 words. Due to the small scale of the corpus, we have not chosen supervised feature selection methods (such as MI and IG) for feature selection. Instead, we use an unsupervised feature selection method, term frequency, to select those features with a frequency of no less than two in this corpus. This method results in a simplified vocabulary that contains 22 words:

Table 6.2 Dimension-reduced text clustering dataset

ID	Sentence
$\boldsymbol{x}_1$	Beijing institute technology university computer science China
$\boldsymbol{x}_2$	Beijing institute technology China computer champion
$\boldsymbol{x}_3$	Beijing institute technology volleyball competition game China
$\boldsymbol{x}_4$	game China volleyball win champion
$\boldsymbol{x}_5$	artificial intelligence machine intelligence intelligence
$\boldsymbol{x}_6$	artificial intelligence computer science intelligent machine human intelligence
$\boldsymbol{x}_7$	artificial intelligence go competition human champion
$\boldsymbol{x}_8$	spring young
$\boldsymbol{x}_9$	spring young grass green
$\boldsymbol{x}_{10}$	grass spring green

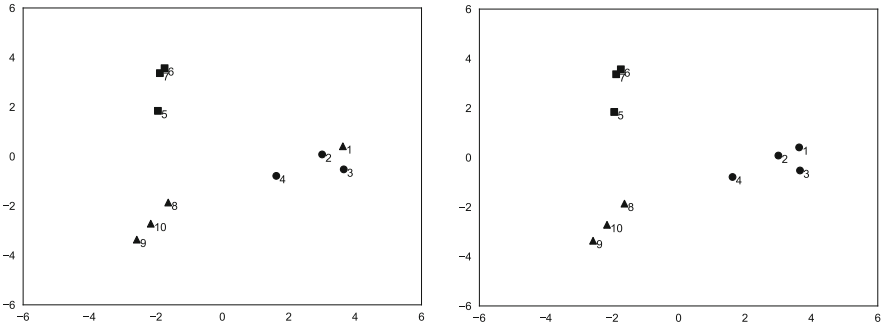
"volleyball," "Beijing," "China," "institute," "win," "go," "champion," "computer," "science," "technology," "human," "race," "university," "artificial," "intelligence," "machine," "game," "competition," "spring," "young," "green," "grass."

The dimension-reduced dataset is shown in Table 6.2.

We perform K-means clustering on the corpus dimension-reduced dataset by setting $K = 3$ and use the Euclidean distance as the similarity measure. We use principal component analysis (PCA) to reduce the dimension of the feature space and take the top two components as the x-axis and y-axis to visualize the clustering process:

(i) Initialization: The initial clusters are $\{C_1 : \{\boldsymbol{x}_4\}, C_2 : \{\boldsymbol{x}_5\}, C_3 : \{\boldsymbol{x}_8\}\}$;

(ii) The first iteration: Calculate the distance of each document to the centroid of each cluster. Taking $\boldsymbol{x}_3$ as an example, its distances to the three centroids $\boldsymbol{x}_4, \boldsymbol{x}_5, and \boldsymbol{x}_8$ are 2.45, 3.16, and 3, respectively. Thus, $\boldsymbol{x}_3$ is assigned to its nearest cluster C_1. After assignment for each document, the updated clusters become $\{C_1 : \{\boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4\}, C_2 : \{\boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7\}, C_3 : \{\boldsymbol{x}_1, \boldsymbol{x}_8, \boldsymbol{x}_9, \boldsymbol{x}_{10}\}\}$, as shown in Fig. 6.2a.



(a) after 1st iteration (b) after 2nd iteration

Fig. 6.2 Clustering text with K-means algorithm ($K = 3$)

(iii) The second iteration: Calculate the distance of each document to the centroid of each cluster after the first iteration. Taking $\boldsymbol{x}_1$ as an example, its distances to the three centroids are 2.08, 3.02, and 2.29. Thus, $\boldsymbol{x}_6$ is assigned to its nearest cluster C_1. After assignment for each document, the updated clusters become $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4\}, C_2 : \{\boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7\}, C_3 : \{\boldsymbol{x}_8, \boldsymbol{x}_9, \boldsymbol{x}_{10}\}\}$, as shown in Fig. 6.2b.

(iv) The third iteration: According to the distance between each document and the centroid of each cluster after the third iteration, the cluster assignments no longer need to be changed, and the algorithm converges. The final clusters are $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4\}, C_2 : \{\boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7\}, C_3 : \{\boldsymbol{x}_8, \boldsymbol{x}_9, \boldsymbol{x}_{10}\}\}$, as shown in Fig. 6.2b.

Although the K-means algorithm is widely used because of its simplicity and efficiency, it still has several shortcomings: ① it remains difficult to determine the value of clustering number K, and ② the result depends on the selected initial centroids or metric selection. For example, if documents $\boldsymbol{x}_2$, $\boldsymbol{x}_5$, and $\boldsymbol{x}_8$ are selected as initial centroids of three clusters, the algorithm will terminate within one iteration, and the final clustering results will be $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4\}, C_2 : \{\boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7\}, C_3 : \{\boldsymbol{x}_8, \boldsymbol{x}_9, \boldsymbol{x}_{10}\}\}$.

6.2.2 Single-Pass Clustering

In comparison with K-means, single-pass clustering is an even simpler and more efficient clustering algorithm, as it only needs to traverse a collection of documents once to perform the clustering. In the initial stage, the algorithm takes a document from the corpus and constructs a cluster with this document. It then iteratively processes a new document and computes the similarity between this document and each existing cluster. If the similarity is lower than a predefined threshold, a new cluster will be generated; otherwise, it will be assigned to the cluster with the highest similarity. This process repeats until all the documents in the dataset have been processed.

Single-pass clustering involves a similarity computation between a document and a cluster, the methods for which are summarized in Sect. 6.2. In standard single-pass clustering, the similarity between the document and the mean vector of the cluster is employed.

The detailed algorithm is described as follows.

We perform single clustering on the dimension-reduced dataset shown in Table 6.2. The opposite value of the Euclidean distance is used as the similarity metric, and the threshold T is set to be -2.3. All documents are processed in sequence. The clustering process is as follows:

(i) Read the first document $\boldsymbol{x}_1$, establish an initial cluster C_1, and assign $\boldsymbol{x}_1$ to C_1. The initial cluster is $\{C_1 : \{\boldsymbol{x}_1\}\}$;

Algorithm 2: Single-pass clustering algorithm

Input : dataset $\mathscr{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, similarity threshold T;
Output: clusters $\{C_1, C_2, \ldots, C_M\}$.

$M = 1$; $C_1 = \{x_1\}$; $\boldsymbol{m}_1 = \boldsymbol{x}_1$
for $i = 2, \ldots, N$ **do**
  for $k = 1, \ldots, M$ **do**
    calculate the similarity $d(\boldsymbol{x}_i, \boldsymbol{m}_k)$ between $\boldsymbol{x}_i$ and $\boldsymbol{m}_k$
  end
  select the cluster of highest similarity $k^* = \arg\max_k \{d(\boldsymbol{x}_i, \boldsymbol{m}_k)\}$
end
if $d(\boldsymbol{x}_i, \boldsymbol{m}_{k^*}) > T$ **then**
  add x_i into cluster C_{k^*}: $C_{k^*} \leftarrow (C_{k^*} \cup \boldsymbol{x}_i)$
  update the mean vector of C_{k^*}: $\boldsymbol{m}_{k^*} = \frac{1}{|C_{k^*}|} \sum_{\boldsymbol{x}_j \in C_{k^*}} \boldsymbol{x}_j$
end
else
  $M+ = 1$; $C_M = \{\boldsymbol{x}_i\}$
end

(ii) Process document $\boldsymbol{x}_2$. Because the similarity between $\boldsymbol{x}_2$ and the centroid of C_1 is -2.18, which is higher than T, we assign $\boldsymbol{x}_1$ to C_1. The updated clusters are $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2\}\}$;

(iii) Process document $\boldsymbol{x}_3$. The similarity between $\boldsymbol{x}_3$ and the centroid of the existing clusters C_1 is -2.18, which is higher than T; therefore, we assign $\boldsymbol{x}_3$ to C_1. The updated clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}\}$;

(iv) Process document $\boldsymbol{x}_4$. The similarity between $\boldsymbol{x}_4$ and the centroid of the existing cluster C_1 is -2.47. The highest similarity is lower than T; therefore, we assign $\boldsymbol{x}_4$ to C_2. The updated clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}, C_2 : \{\boldsymbol{x}_4\}\}$;

(v) Process document $\boldsymbol{x}_5$. The similarities between $\boldsymbol{x}_5$ and the centroids of existing clusters C_1 and C_2 are -2.85 and -2.83, respectively. The highest similarity is lower than T; therefore, we establish a new cluster C_3 and assign $\boldsymbol{x}_5$ to it. The updated clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}, C_2 : \{\boldsymbol{x}_4\}, C_3 : \{\boldsymbol{x}_5\}\}$;

(vi) Process document $\boldsymbol{x}_6$. The similarities between $\boldsymbol{x}_6$ and the centroids of existing clusters C_1, C_2 and C_3 are -3.02, -3.32, and -1.73 respectively. The highest similarity is higher than T (with C_3); therefore, we assign $\boldsymbol{x}_6$ to C_3. The updated clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}, C_2 : \{\boldsymbol{x}_4\}, C_3 : \{\boldsymbol{x}_5, \boldsymbol{x}_6\}\}$;

(vii) Process document $\boldsymbol{x}_7$. The similarities between $\boldsymbol{x}_7$ and the centroids of existing clusters C_1, C_2 and C_3 are -3.13, -3.0, -2.18 respectively. The highest similarity is higher than T (with C_3); therefore, we assign $\boldsymbol{x}_7$ to C_3. The updated clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}, C_2 : \{\boldsymbol{x}_4\}, C_3 : \{\boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7\}\}$;

(viii) Process document $\boldsymbol{x}_8$. The similarities between $\boldsymbol{x}_8$ and the centroids of existing clusters C_1, C_2 and C_3 are -2.67, -2.65, -2.33 respectively. The highest similarity is lower than T; therefore, we establish a

new cluster C_4 and assign $\boldsymbol{x}_8$ to it. The updated clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}, C_2 : \{\boldsymbol{x}_4\}, C_3 : \{\boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7\}, , C_4 : \{\boldsymbol{x}_8\}\}$;

(ix) Process document $\boldsymbol{x}_9$. The similarities between $\boldsymbol{x}_9$ and the centroids of existing clusters C_1, C_2, C_3, C_4, and C_5 are -3.02, -3.0, -2.73 and -1.41 respectively. The highest similarity is higher than T (with C_4); therefore, we assign $\boldsymbol{x}_9$ to C_4. The updated clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}, C_2 : \{\boldsymbol{x}_4\}, C_3 : \{\boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7\}, , C_4 : \{\boldsymbol{x}_8, \boldsymbol{x}_9\}\}$;

(x) Process document $\boldsymbol{x}_{10}$. The similarities between $\boldsymbol{x}_{10}$ and the centroids of existing clusters C_1, C_2, C_3 and C_4 are -2.85, -2.83, -2.53 and -1.22 respectively. The highest similarity is higher than T (with C_4); therefore, we assign $\boldsymbol{x}_{10}$ to C_4. The updated clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}, C_2 : \{\boldsymbol{x}_4\}, C_3 : \{\boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7\}, C_4 : \{\boldsymbol{x}_8, \boldsymbol{x}_9, \boldsymbol{x}_{10}\}\}$;

Thus, all documents in the corpus are processed. The final clustering result is shown in Fig. 6.3.

Because of its simplicity and efficiency, the single-pass clustering algorithm is suitable for scenarios including large-scale and real-time streaming data, such as topic detection and tracking, which we will introduce in Chap. 9. However, it also contains some inherent flaws. For example, its performance greatly depends on the order of processed documents, and the threshold is sometimes hard to determine in advance.

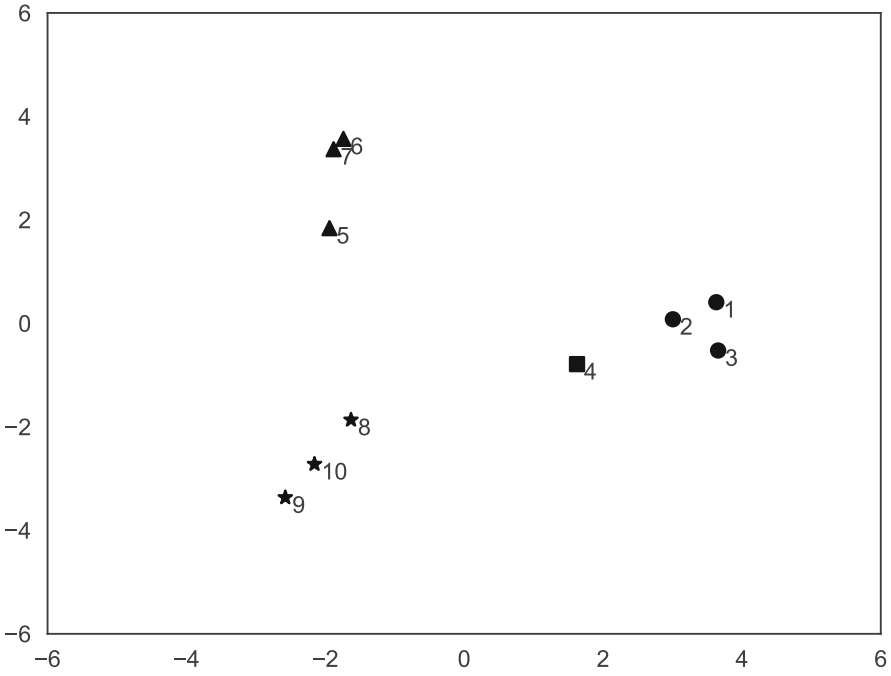


Fig. 6.3 Clustering result with single-pass clustering algorithm

6.2.3 Hierarchical Clustering

Hierarchical clustering is a class of cluster analysis methods that seek to build a hierarchy of clusters. It can be divided into two main types:

(1) Agglomerative hierarchical clustering: This is a bottom-up approach where each element starts in its own cluster and similar pairs of clusters are merged as we move up the hierarchy.
(2) Divisive hierarchical clustering: This is a top-down approach where all elements start in one cluster and splits are performed recursively as we move down the hierarchy.

In agglomerative hierarchical clustering, each document is initially considered as an individual cluster, and the most similar two clusters are merged together in each iteration until one cluster or K clusters are formed.

In the clustering process, the similarity between two clusters needs to be calculated. The commonly used measures, including single linkage, complete linkage, average linkage, and Ward's method, are described in detail in Sect. 6.2.

Algorithm 3: Agglomerative hierarchical clustering algorithm

Input : dataset $\mathscr{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, number of clusters K;
Output: clusters $\{C_1, C_2, \ldots, C_K\}$.

for $i = 1, \ldots, N$ **do**
  $C_i = \{\boldsymbol{x}_i\}$
end
for $i = 1, \ldots, N$ **do**
  for $j = 1, \ldots, N$ **do**
    calculate the similarity between two clusters $d(C_i, C_j)$
  end
end
while *size*($\mathscr{C}$) $> K$ **do**
  find the nearest two clusters C_{i^*} and C_{j^*}.
  for $h = 1, \ldots,$ *size*($\{C_k\}$) **do**
    if $h \neq i^*$ *and* $h \neq j^*$ **then**
      update the similarity $d(C_h, C_{i^*} \cup C_{j^*})$
    end
    delete C_{i^*} and C_{j^*} from $\mathscr{C}$
    add $C_{i^*} \cup C_{j^*}$ to $\mathscr{C}$
    update the index of each cluster and record samples in each cluster.
  end
end

The results of hierarchical clustering can be represented by a dendrogram, which is a tree-like diagram that records the sequences of merges or splits, as shown in Fig. 6.4. Each leaf node represents a document, and each intermediate node has two subnodes, indicating that the two component clusters merged into one cluster. The

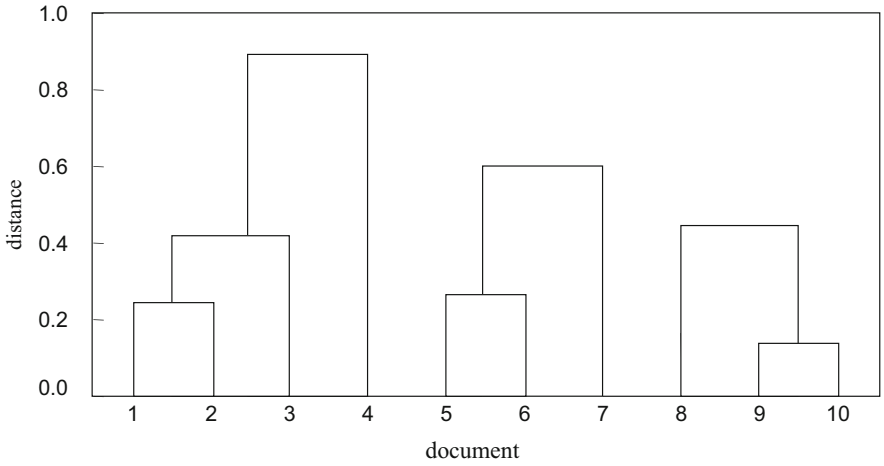


Fig. 6.4 The agglomerative hierarchical clustering results (dendrogram)

height of the leaf nodes is 0, and the height of each intermediate node represents the distance of its two subnodes and is inversely proportional to their similarity. Cutting the tree horizontally at a given height yields partitioning clustering results at a selected level.

We perform agglomerative hierarchical clustering on the dimension-reduced clustering dataset shown in Table 6.2 by using cosine to measure the similarity between documents and average linkage to measure the similarity between clusters and setting the expected number of clusters K as 3. The clustering process is as follows:

(i) Initialize a cluster for each document. This results in ten clusters in our task. The initial clusters are $\{C_1 : \{\boldsymbol{x}_1\}, C_2 : \{\boldsymbol{x}_2\}, C_3 : \{\boldsymbol{x}_3\}, C_4 : \{\boldsymbol{x}_4\}, C_5 : \{\boldsymbol{x}_5\}, C_6 : \{\boldsymbol{x}_6\}, C_7 : \{\boldsymbol{x}_7\}, C_8 : \{\boldsymbol{x}_8\}, C_9 : \{\boldsymbol{x}_9\}, C_{10} : \{\boldsymbol{x}_{10}\}\}$.

(ii) Compute the similarities between each cluster pair. Because the similarity between clusters C_9 and C_{10} is the highest (0.87), the two clusters are merged. The updated clusters are $\{C_1 : \{\boldsymbol{x}_1\}, C_2 : \{\boldsymbol{x}_2\}, C_3 : \{\boldsymbol{x}_3\}, C_4 : \{\boldsymbol{x}_4\}, C_5 : \{\boldsymbol{x}_5\}, C_6 : \{\boldsymbol{x}_6\}, C_7 : \{\boldsymbol{x}_7\}, C_8 : \{\boldsymbol{x}_8\}, C_9 : \{\boldsymbol{x}_9, \boldsymbol{x}_{10}\}\}$.

(iii) Compute the similarities between each cluster pair and merge the two clusters C_1 and C_2, which have the highest similarity. The updated clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2\}, C_3 : \{\boldsymbol{x}_3\}, C_4 : \{\boldsymbol{x}_4\}, C_5 : \{\boldsymbol{x}_5\}, C_6 : \{\boldsymbol{x}_6\}, C_7 : \{\boldsymbol{x}_7\}, C_8 : \{\boldsymbol{x}_8\}, C_9 : \{\boldsymbol{x}_9, \boldsymbol{x}_{10}\}\}$.

(iv) Compute the similarities between each cluster pair and merge the two clusters C_5 and C_6, which have the highest similarity. The updated clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2\}, C_3 : \{\boldsymbol{x}_3\}, C_4 : \{\boldsymbol{x}_4\}, C_5 : \{\boldsymbol{x}_5, \boldsymbol{x}_6\}, C_7 : \{\boldsymbol{x}_7\}, C_8 : \{\boldsymbol{x}_8\}, C_9 : \{\boldsymbol{x}_9, \boldsymbol{x}_{10}\}\}$.

(v) Compute the similarities between each cluster pair and merge the two clusters C_1 and C_3, which have the highest similarity. The updated clustering

result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}, C_4 : \{\boldsymbol{x}_4\}, C_5 : \{\boldsymbol{x}_5, \boldsymbol{x}_6\}, C_7 : \{\boldsymbol{x}_7\}, C_8 : \{\boldsymbol{x}_8\}, C_9 : \{\boldsymbol{x}_9, \boldsymbol{x}_{10}\}\}$.

(vi) Compute the similarities between each cluster pair and merge the two clusters C_8 and C_9, which have the highest similarity. The updated clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}, C_4 : \{\boldsymbol{x}_4\}, C_5 : \{\boldsymbol{x}_5, \boldsymbol{x}_6\}, C_7 : \{\boldsymbol{x}_7\}, C_8 : \{\boldsymbol{x}_8, \boldsymbol{x}_9, \boldsymbol{x}_{10}\}\}$.

(vii) Compute the similarities between each cluster pair and merge the two clusters C_5 and C_7, which have the highest similarity. The updated clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}, C_4 : \{\boldsymbol{x}_4\}, C_5 : \{\boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7\}, C_8 : \{\boldsymbol{x}_8, \boldsymbol{x}_9, \boldsymbol{x}_{10}\}\}$.

(viii) Compute the similarities between each cluster pair and merge the two clusters C_1 and C_4, which have the highest similarity. The updated clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4\}, C_5 : \{\boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7\}, C_8 : \{\boldsymbol{x}_8, \boldsymbol{x}_9, \boldsymbol{x}_{10}\}\}$.

At this point, the number of clusters reaches the preset value ($K = 3$), and the hierarchical clustering ends. The clustering results in terms of the dendrogram are shown in Fig. 6.4.

The top-down divisive hierarchical clustering process follows the opposite process as the bottom-up clustering process. Initially, all the documents are contained in one cluster, and the documents that are not similar are separated iteratively from the cluster until all documents are divided into different clusters.

6.2.4 Density-Based Clustering

In density-based clustering, clusters are defined as areas of higher density than the remainder of the data. The basic concept is that the densely distributed data points in the data space are separated by the sparsely distributed data points; the connected high-density regions are the target clusters we are looking for.

Density-based spatial clustering of applications with noise (DBSCAN) is a representative algorithm of density-based clustering. Given a set of data points in the data space, the points that are closely connected (points with many nearby neighbors) will be grouped together and marked as high-density regions, and the points that lie alone in low-density regions (whose nearest neighbors are too far away) will be marked as outliers.

Let r denote the radius of the neighborhood and n denote the minimum number of data points required to construct a high-density region. On this basis, the following basic concepts are defined.

- r-neighborhood: The r-neighborhood of a sample P refers to the circular domain with P at the center and r as the radius.
- Core point: Point P is a core point if P's r-neighborhood contains at least n points.
- Directly reachable: Point Q is directly reachable from P if Q is in the r-neighborhood of P.
- Reachable: If there exists a sequence of data points $P_1, P_2, \ldots, P_T$ and P_{t+1} is directly reachable from P_t for any $t = 1, \ldots, T - 1$, we say that point

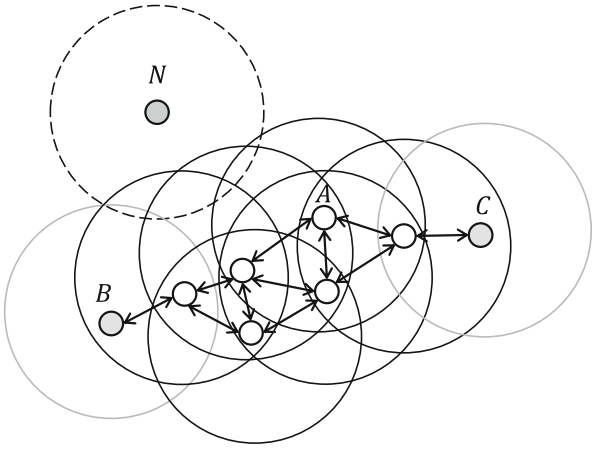


Fig. 6.5 An illustration of the DBSCAN algorithm

P_T is reachable from P_1. According to the definition of direct reachability, $P_1, P2, \ldots, P_{T-1}$ in the sequence are all core points.

- Density-connected: Two points Q_1 and Q_2 are density-connected if both Q_1 and Q_2 can be reachable from a core point P.

The DBSCAN algorithm supposes that for any core point P, the points in the dataset that are reachable from P belong to the same cluster. Figure 6.5 gives an example of the DBSCAN algorithm where $n = 4$. Point A and other hollow points are core samples, and boundary points B and C are non-core points. Points B and C are reachable from point A, that is, B and C are density-connected; therefore, together with the core points, they construct a cluster. Point N is a noise point that is not density-connected to A, B, or C.

Starting from a core point, the DBSCAN algorithm expands continuously to reachable regions to obtain a maximum region containing core points and boundary points. In this region, any two points are connected with each other and aggregated into a cluster. The process is repeated for each unlabeled core point until all core points in the dataset are processed. The points that are not included in any clusters are called noise points and grouped in a noise cluster.

We perform DBSCAN clustering on the dimension-reduced clustering dataset shown in Table 6.2, using cosine distance with $r = 0.6$ and $n = 3$. The clustering process is as follows.

(i) Initially, mark all data points as unvisited. Select $\boldsymbol{x}_1$ first and mark it as visited. The r-neighborhood of $\boldsymbol{x}_1$ includes points $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$. Because its size is not smaller than n, make the connected high-density region $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$. The clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}\}$;

(ii) Select an unvisited point $\boldsymbol{x}_4$ and mark it as visited. The r-neighborhood of $\boldsymbol{x}_4$ includes $\boldsymbol{x}_1, \boldsymbol{x}_2$, and $\boldsymbol{x}_3$. The updated clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4\}\}$;

(iii) Select an unvisited point $\boldsymbol{x}_5$ and mark it as visited. The r-neighborhood of $\boldsymbol{x}_5$ includes $\boldsymbol{x}_5$, $\boldsymbol{x}_6$, and $\boldsymbol{x}_7$, the size of which is not smaller than n.

Algorithm 4: DBSCAN algorithm

Input : dataset $\mathscr{D}$, radius r, the number of samples n required to construct a high-density region;
Output: set of clusters $\mathscr{C}$.

```
C = ∅
for P in D do
    if P has been visited then
        continue
    end
    find a set R_P of all samples in the r-neighborhood of P
    if |R_P| < n then
        mark P as a noise sample
    end
    else
        add sample P to a new cluster C
        find a set S_P of directly reachable samples from P
        for Q in S_P do
            if Q is a noise sample then
                add Q to cluster C
            end
            if Q has not been visited then
                add Q to cluster C
            end
            find a set R_Q of samples within the r-neighborhood of Q
            if |R_P| ≥ n then
                S_P = S_P ∪ R_Q
            end
            add C to C
        end
    end
end
```

Therefore, make the connected high-density region $\{\boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7\}$ a new cluster. The updated clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4\}, C_2 : \{\boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7\}\}$;

(iv) Select an unvisited point $\boldsymbol{x}_8$ and mark it as visited. The r-neighborhood of $\boldsymbol{x}_8$ includes $\boldsymbol{x}_8$, $\boldsymbol{x}_9$, and $\boldsymbol{x}_{10}$, the size of which is smaller than n. Therefore, make the connected high-density region $\{\boldsymbol{x}_8, \boldsymbol{x}_9, \boldsymbol{x}_{10}\}$ a new cluster. The updated clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4\}, C_2 : \{\boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7\}, C_3 : \{\boldsymbol{x}_8, \boldsymbol{x}_9, \boldsymbol{x}_{10}\}\}$;

(v) At this point, all points in the dataset are marked as visited, and clustering is finished. The final clustering result is $\{C_1 : \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4\}, C_2 : \{\boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7\}, C_3 : \{\boldsymbol{x}_8, \boldsymbol{x}_9, \boldsymbol{x}_{10}\}\}$, as shown in Fig. 6.6.

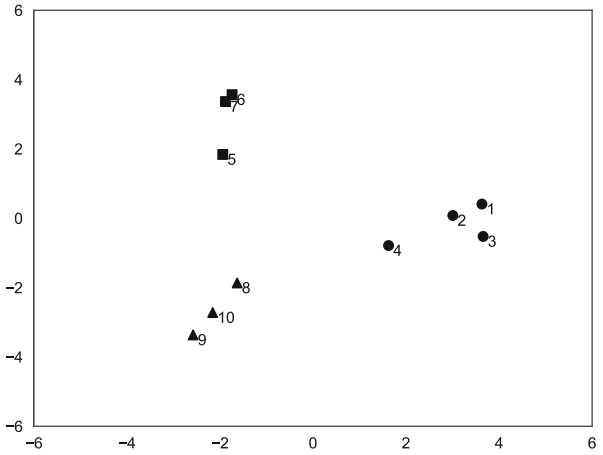


Fig. 6.6 The clustering result with the DBSCAN clustering algorithm

6.3 Evaluation of Clustering

The evaluation of clustering is also called cluster validity analysis. There are two main categories of methods for evaluating clustering: external criteria and internal criteria. The main difference between them is whether external information is used for clustering validation.

6.3.1 External Criteria

In external criteria, the quality of clustering is measured by the consistency between the clustering result and a clustering reference, which is considered the ground truth. The clustering reference is usually manually labeled.

For a dataset $\mathscr{D} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n\}$, assume that the clustering reference is denoted by $\mathscr{P} = \{P_1, P_2, \ldots, P_m\}$, where P_i represents the i-th cluster in the clustering reference, and the clustering result is $\mathscr{C} = \{C_1, C_2, \ldots, C_k\}$, where C_i is a model-obtained cluster. For any two different samples $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$ in $\mathscr{D}$, define the following four relationships based on their co-occurrences in $\mathscr{C}$ and $\mathscr{P}$, respectively:

(1) SS: $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$ belong to the same cluster in $\mathscr{C}$ and the same cluster in $\mathscr{P}$;
(2) SD: $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$ belong to the same cluster in $\mathscr{C}$ but different clusters in $\mathscr{P}$;
(3) DS: $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$ belong to different clusters in $\mathscr{C}$ but the same cluster in $\mathscr{P}$;
(4) DD: $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$ belong to different clusters in $\mathscr{C}$ and different clusters in $\mathscr{P}$;

Let a, b, c, d denote the number of SS, SD, DS, and DD, respectively. The following evaluation measures can be defined:

(a) Rand index

$$\mathrm{RS} = \frac{a+d}{a+b+c+d} \tag{6.18}$$

(b) Jaccard index

$$\mathrm{JC} = \frac{a}{a+b+c} \tag{6.19}$$

(c) Fowlkes and Mallows index

$$\mathrm{FMI} = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}} \tag{6.20}$$

The range of the above three indices is [0, 1]. The larger the value of the index is, the higher the similarity of $\mathscr{C}$ and $\mathscr{P}$ and the better the performance of the clustering result $\mathscr{C}$.

6.3.2 *Internal Criteria*

The internal criteria are based on internal information (such as distribution and structure) and evaluate a cluster without reference to external information. Cohesion and separation are two key factors for evaluating the clustering performance in internal criteria. Generally, internal criteria prefer clusters with high similarity within a cluster (high cohesion) and low similarity between clusters (high separation).

The typical internal criteria include the silhouette coefficient, I index, Davies–Bouldin index, Dunn index, Calinski–Harabasz index, Hubert's Γ statistic, and the cophenetic correlation coefficient. Most of these metrics include factors of both cohesion and separation. In the following, we will introduce the representative measure: the silhouette coefficient. Readers can refer to (Liu et al. 2010) for the details of other methods.

The silhouette coefficient was first proposed by Peter J. Rousseeuw in 1986 and has become a commonly used internal criterion for clustering evaluation. Assuming $\boldsymbol{d}$ is a sample belonging to cluster C_m, we first calculate the average distance between $\boldsymbol{d}$ and the other samples in C_m as:

$$a(\boldsymbol{d}) = \frac{\sum\limits_{\boldsymbol{d}' \in C_m, \boldsymbol{d} \neq \boldsymbol{d}'} \mathrm{dist}(\boldsymbol{d}, \boldsymbol{d}')}{|C_m| - 1} \tag{6.21}$$

We then calculate the minimum average distance between $\boldsymbol{d}$ and the samples in the other clusters:

$$b(\boldsymbol{d}) = \min_{C_j:1\le j\le k, j\neq m} \left\{ \frac{\sum\limits_{\boldsymbol{d}'\in C_j} \mathrm{dist}(\boldsymbol{d}, \boldsymbol{d}')}{|C_j|} \right\} \tag{6.22}$$

Among them, $a(\boldsymbol{d})$ reflects the degree of cohesion in the cluster to which $\boldsymbol{d}$ belongs; $b(\boldsymbol{d})$ reflects the degree of separation between $\boldsymbol{d}$ and the other clusters.

On this basis, the silhouette coefficient with respect to $\boldsymbol{d}$ is defined as follows:

$$\mathrm{SC}(\boldsymbol{d}) = \frac{b(\boldsymbol{d}) - a(\boldsymbol{d})}{\max\{a(\boldsymbol{d}), b(\boldsymbol{d})\}} \tag{6.23}$$

The overall silhouette coefficient is then defined as the average silhouette coefficient across all samples in the dataset:

$$\mathrm{SC} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{SC}(\boldsymbol{d}_i) \tag{6.24}$$

The range of the silhouette coefficient is $[-1, 1]$. The higher the silhouette coefficient is, the better the clustering performance.

6.4 Further Reading

The performance of text clustering depends on the quality of the text representation. Traditional text clustering methods mainly use the vector space model for text representation. This type of representation has some inherent shortcomings, including high-dimensional and sparsity problems, which are inefficient for similarity calculation and text clustering.

In text classification, supervised feature selection methods (e.g., MI and IG) are widely used to improve the quality of text representation. However, because the labels of documents are unknown in text clustering, we can only use unsupervised feature selection methods (e.g., document frequency and term frequency). The unsupervised feature extraction algorithms (e.g., PCA, ICA) are also options for dimension reduction in text clustering. In addition, topic models such as latent semantic analysis (LSA), probabilistic latent semantic analysis (PLSA), and latent Dirichlet distribution (LDA) also provide a way to represent a document by transforming the high-dimensional sparse vectors of words into low-dimensional dense vectors of topics. In addition, some studies also attempt to use the concepts in a knowledge base (such as WordNet, HowNet, Wikipedia, etc.) to guide text representation, similarity calculation, and clustering.

In recent years, with the rise of deep learning, distributed representations such as word embedding have been widely used in text data mining. For example, as introduced in Chap. 3, a piece of text at different levels (e.g., word, phrase, sentence, and document) can be represented by a densely distributed low-dimensional vector. Another advantage of representation learning is that it can learn a task-related representation. Both advantages bring new perspectives to text clustering.

In addition to the clustering methods we described above, there are some special clustering algorithms, such as suffix tree clustering (STC), that are specific to text processing. As a type of data structure, a suffix tree was first proposed to support effective matches and queries for strings. By using the suffix tree structure to represent and process text, the suffix tree clustering algorithm regards text as a sequence of words rather than a set of words and captures more word order information.

Clustering text streams is a special problem of text clustering, which has been widely used in the fields of topic detection and tracking and social media mining. Unlike traditional text clustering, the text data in these fields often appear in the form of online text streams, which creates challenges for text clustering. The single-pass clustering algorithm is a widely used method for real-time large-scale text stream clustering. We will also see in Chap. 9 that some online variants of the traditional clustering algorithms, such as group-average agglomerative clustering (Allan et al. 1998a; Yang et al. 1998), have also been proposed to address these challenges.

Exercises

6.1 Please point out the similarities and differences between the classification and clustering problems.

6.2 What is the relationship between Euclidean distance and cosine similarity when measuring the similarity of two documents?

6.3 Is KL divergence suitable for the similarity calculation of short documents? In addition to KL divergence, can you think of other distribution-based similarity calculation methods?

6.4 Please give the detailed K-means clustering process for the clustering dataset in Table 6.2 when document $\boldsymbol{x}_1$, $\boldsymbol{x}_5$, and $\boldsymbol{x}_8$ are selected as the initial centroids.

6.5 What is the single-pass clustering results if the document order for processing is reversed in Table 6.2?

6.6 Please try to perform divisive hierarchical clustering on the clustering dataset in Table 6.2.