Summarization of Text Clustering based Vector Space Model

Mingzhen Chen, Yu Song
Department of Compute Science
Zhejiang University
Hangzhou, Zhejiang Province 310027, China
cmzboy1983@163.com; ssqq@mail.hz.zj.cn

Abstract

Text clustering is an important task of natural language processing and is widely applicable in areas such as information retrieval and web mining. The representation of document and the clustering algorithm are the key issues of text clustering. This paper discusses Vector Space Model(VSM)-based clustering algorithms. This paper reviewed the text clustering algorithm.

.

Keywords: Vector Space Model (VSM), Text clustering, Document representation, Distance function.

1. Introduction

Text document clustering techniques were initially developed to improve precision and recall of information retrieval systems by effectively partitioning texts. More and more academic papers discussing text document clustering are published in international top-level conference. In this paper, we discuss partitional clustering algorithms and hierarchical clustering algorithms. And we will discuss the representation of document.

Clustering is a special type of classification. Clustering is not the records classification. Clustering is the classification of the reverse method. The purpose of clustering is based on maximizing the similarity within categories. To put similar things together is the significance of the clustering].

2. Document representation

First, we should learn a lexical ontology, HowNet. And we will use it to explore the concepts in the documents. HowNet is a lexical database for Chinese. HowNet consists of over 85,000 concepts and about 90,000 lexical entries. In addition, HowNet can provide the morphological capability for stemming the terms. In HowNet, the terms which do not appear for lexical entries will perform stemming. We use the HowNet-based clustering method to explore what beneficial effects can be achieved for document clustering by considering both noun phrases and semantic relationships. We use it for document representation.

The vector space model is the most established and well-known method of the document weighting approaches. The VSM is an approach that encodes a so-called "bag-of-words" representation, where explicit sequential order information is not explicitly captured.

Document refers to text or text fragment, and it generally refers to an article. Term is the basic semantic unit of the document, it typically include the words or phrases. Document can be represented by term list. $D(t_1,t_2,...,t_n)$, t_k is term and $1 \leqslant k \leqslant n$. Term Weight is the weight of term, it indicates their importance in the document. Document can be represented by term list with term weight. $D(t_1,w_1,t_2,w_2,...,t_n,w_n)$, w_k is term weight of term t_k and $1 \leqslant k \leqslant n$.

We use the TF-IDF (term frequency–inverse document frequency) measure. It is the commonest and classic statistical measure used in text mining. TF-IDF gives a weighting or relevance of how important a word is to a document.

In Vector Space Model, a document d_i is represented by a set of terms ($t_1,t_2 \ldots t_n$) wherein each t_j is a word that appears in the text document d_i, and n denotes the total number of various words in the index used to identify the meaning of the text document. Word t_j has a corresponding weight w_i calculated as a combination of the statistics term frequency $TF(t_j, d_i)$ and inverse document frequency $IDF(t_j)$.

$W_t(t_j,d_i)=TF(t_j,d_i)*N/DF(t_j)$,

where $TF(t_j,d_i)$ is the frequency of term t_j in document d_i, N is the total number of documents in the corpus, $DF(t_j)$ is the number of documents in which term t_j occurs.

Therefore, document d_i can be represented as a specific n-dimensional vector d_i as

$d_i=(w_1,w_2,w_3,\ldots,w_n)$.

We use another more efficient algorithm.

$$w(t,d)=\frac{tf(t,d)\times\log(N/n_i+0.01)}{\sqrt{\sum[tf(t,d)\times\log(N/n_i+0.01)]^2}} \quad (1)$$

$w(t,d)$ is the weight of term t in document d, $tf(t,d)$ is the term frequency of term t in document d, N is the total number of documents, n_t is the number of the documents which contains term t, the denominator of formula is normalization factor.

The TF value is proportional to the frequency of the word in the document. The IDF value is inversely proportional to its frequency in the documents. The function encodes the intuitions that: (i) the more often a word occurs in a document, the more it is representative of the content of the text; (ii) The more text the word occurs in, the less distinction it is. In classification, the Inverse Document frequency is a good

978-1-4244-5268-2/09/$25.00 ©2009 IEEE

index of the usefulness of a word. The test document is also subjected to TF and IDF weighting which is used for the training documents].

3. Similarity measure

VSM model which is applied to the index has a great advantage. To search terms and the calculation of the similarity between the texts is very convenient. The calculation of this similarity using between texts is also very natural.

There are forms of distance measures. Sometimes we call it distance function or similarity function. Vector inner product, cosine measure, Correlation distance, Spearman distance, Euclidean distance, City Block distance, Mahalanobis distance, Minkowski distance, Tanimoto distance, Hamming distance, Jaccard distance and so on. We use these distance function to find the similarity of documents.

$$sim(D_j, D_k) = \cos\theta = \frac{\sum_{i=1}^{n} w_{ij} w_{ik}}{\sqrt{\sum_{i=1}^{n} w_{ij}^2 \sum_{i=1}^{n} w_{ik}^2}} \qquad (2)$$

For the bisecting K-means clustering, the overall similarity criterion Op is used to bisect the clusters, i.e., a cluster is bisected so that the resulting 2-way clustering solution optimizes the criterion function:

$$O_p = \max \sum_{k=1}^{n} \sqrt{\sum_{d_i, d_j \in C_k} sim(\vec{d}_i, \vec{d}_j)} \qquad (3)$$

Where $\vec{d}_i$ *and* $\vec{d}_j$ *is the feature vector of document* d_i *and* d_j*, respectively,* C_k *is the* k*th cluster, and* n *is the total number of clusters.*

For the Hierarchical Agglomerative Clustering (HAC), the UPGMA scheme and is selected. Through empirical experiments, it has verified that this method outperforms other HAC methods. The UPGMA scheme defines the cluster similarity as follows:

$$sim(cluster_i, cluster_j) = \frac{\sum_{d_i \in cluster_i, d_j \in cluster_j} \cos(\vec{d}_i, \vec{d}_j)}{|cluster_i| * |cluster_j|} \qquad (4)$$

Where d_i and d_j are documents in $cluster_i$ and $cluster_j$, respectively, $|cluster_i|$ is the size of $cluster_i$, and $|cluster_j|$ is the size of $cluster_j$.

4. Document clustering

Clustering is one technology to find intrinsic structures in data sets. Text clustering method usually uses the document vector space model to split the document into vectors in high dimensional space, and then make clustering of these vectors. Text clustering Can generally be divided into partitional clustering algorithms and hierarchical clustering algorithms.

4.1. Partitional clustering algorithms

The k-means algorithm is the best known partitional clustering algorithm.

Step 1: Initialize the algorithm with guessed centers C.
Step 2: For each document di, compute its membership m(cj|di) in each center cj and its weight w(di).
Step 3: For each center cj, recomputed its location from all documents di according to their memberships and weights.
Step 4: Repeat steps 2 and 3 until convergence.

Another form of description of k-means algorithm is:
Algorithm k-means(k, D)
choose k data points as the initial centroids (cluster centers)
repeat
for each data point x ∈D do
compute the distance from x to each centroid;
assign x to the closest centroid // a centroid represents a cluster
endfor
re-compute the centroid using the current cluster memberships
until the stopping criterion is met

Then we will discuss one improved k-means algorithm which described in [1].

Step 1: Select k centers using k-means algorithm, vector is $\{C_1, C_2, ..., C_K\}$
Step 2: For each term d_i in the document D, compute the similarity of d_i and cluster C_i
Step 3: If term d_i have the maximum similarity of center c_i, and then we assign d_i into the cluster of center c_i. The clustering of document D is Clusters = $\{C_1, C_2, ..., C_K\}$. We store the similarity of each term and the cluster.
Step 4: Get cluster C_i which is produced with the clustering of turn t-1. Calculate the similarity of each term t_i and the cluster center C_i. Find the minimum of the similarity, and sign it with MinOfSim
Step 5: In the cluster C_i, we compute the similarity of each term t and the cluster center c_i. Select all terms which has the similarity over 1-b*(1-MinOfSim), and sign the collection as c_i'.
Step 6: compute the center of c_i'. In the clustering of turn t, we use the center of c_i' as the new cluster center.
Step 7: Repeat steps 2 to 6 until convergence.

To find good-quality clustering in spatial data is the goal of this work. K-means algorithm is essentially a greedy algorithm. The algorithm can guarantee local minimum, but is very difficult to guarantee global minimum. We made a number of the corresponding solution. A wrapper method is to use the algorithm multiple times. One commonly used wrapper method is simply running the clustering algorithm several times from different starting points (often called random restart), and taking the best solution. Algorithms such

as used in [4] push this technique to its extreme, at the cost of computation. In [5], it searches for the best initializations possible. In [6], it finds the appropriate number of clusters, and analyzes the difference between the cluster solution and the dataset. In [7], it used genetic algorithm to optimize the value of k. When we know the k value, how to obtain the initial accumulation point is also a matter of concern. These methods are useful, but they only want to fix the problems of clustering algorithms, rather than to improve the clustering algorithms themselves. We are interested in improving the clustering algorithms directly to make them less sensitive to initializations and give better solutions.

4.2. Hierarchical clustering algorithms

Hierarchical clustering is another major clustering approach. It has a number of desirable properties which make it popular. It clusters by producing a nested sequence of clusters like a tree. Singleton clusters (individual data points) are at the bottom of the tree and one root cluster is at the top, which covers all data points. Each internal cluster node contains child cluster nodes. Sibling clusters partition the data points covered by their common parent. Figure 1 shows an example.

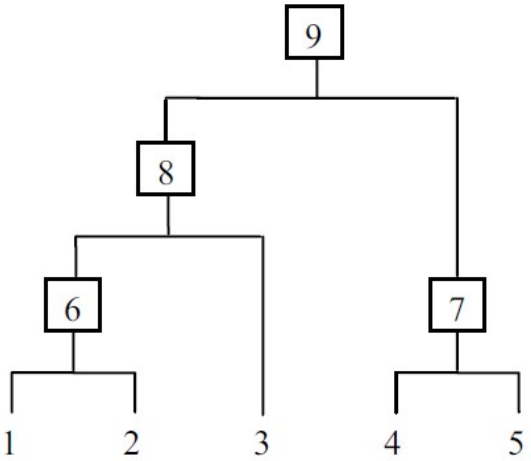

Figure 1. An illustration of hierarchical clustering

There are two main types of hierarchical clustering methods:

Agglomerative (bottom up) clustering:

It builds the dendrogram (tree) from the bottom level, and merges the most similar (or nearest) pair of clusters at each level to go one level up. The process continues until all the data points are merged into a single cluster (i.e., the root cluster).

Divisive (top down) clustering:

It starts with all data points in one cluster, the root. It then splits the root into a set of child clusters. Each child cluster is recursively divided further until only singleton clusters of individual data points remain, i.e., each cluster with only a single point.

One form of description of Agglomerative Algorithm:
Step 1: Make each data point in the data set D a cluster
Step 2: Compute all pair-wise distances of $x1,x2,\ldots,xn \in D$
Step 3: find two clusters that are nearest to each other
Step 4: merge the two clusters form a new cluster c
Step 5: compute the distance from c to all other clusters
Step 6: repeat step 3-5 until there is only one cluster left

Hierarchical clustering can use any form of distance function or similarity function. It produces better results than the k-means method.
All hierarchical clustering methods have main disadvantages. They are their computation complexities and space complexities. This is very inefficient and not practical for large data sets Compared to the k-means algorithm.

5. Conclusion

Text clustering has three issues. They are sparse high-dimensional, multi-word synonyms and polysemy. They lead the clustering to very high time complexity. And they greatly interfere with the accuracy of the clustering algorithm. They cause a sharp decline in the performance of clustering. This is the difficult of the technique.

In this paper, we describe some clustering algorithms which are widely used in the document clustering based VSM model.

Acknowledgement

This project was supported by Jiangdong district of Ningbo City, scientific and technological cooperation plan (20062058).

References

[1] Fang Y C,Parthasarathy S,Schwartz F.Using Clustering to Boost Text Classification[C]//Proc.of the IEEE ICDM Workshop on Text Mining.Maebashi City,Japan:[s.n.],2002:1-9

[2] Hamerly, G. and C. Elkan (2002). Alternatives to the k-means algorithm that find better clusterings, ACM New York, NY, USA.

[3] Rajan, K., V. Ramalingam, et al. (2009). "Automatic classification of Tamil documents using vector space model and artificial neural network." Expert Systems With Applications 36(8): 10914-10918

[4] A. Likas, N. Vlassis, and J. Verbeek. “The global k-means clustering algorithm.” Technical report, Computer Science Institute, University of Amsterdam, The Netherlands, February 2001. IAS-UVA-01-02

[5] M. Meila and D. Heckerman. "An experimental comparison of model-based clustering methods." Machine learning, 42:9–29, 2001

[6] D. Pelleg and A. Moore. X-means: "Extending K-means with efficient estimation of the number of clusters." In Proceedings of the

17th International Conf. on Machine Learning, pages 727–734. Morgan Kaufmann, San Francisco, CA, 2000

[7] Salton G, Wong A, Yang C S. A Vector Space Model for Automatic Indexing [J]. Communications of the ACM, 1975 (18): 613 - 620