Technology Review

Anant Ashutosh Sharma
anantas2

A Survey Of Text Clustering Algorithms

Introduction

In the recent years, the amount of text data being generated has been increasing exponentially. Data mining techniques like clustering are being used by organizations to mine actionable information from the newly generated data, as well as the already existing data. Traditional clustering methods have focussed on quantitative data, with numerical attributes, as well as categorical data, with attributes as nominal values. In addition to such traditional use cases of clustering, it can also find applicability in number of other task such as document organization and browsing, corpus summarization, document classification and many more. There a lot of existing algorithms like the K-means clustering which are general purpose algorithms can be universally used on different kinds of data, including textual data.

Any textual document can be represented on form of a binary vector where the vector represents presence and absence of specific works. Using this representation, any categorical data clustering algorithms can be used {1]. This representation is generally further modified to include the different frequencies of words using the TF-IDF weighing. With his representation, quantitative data clustering algorithms can be implemented to carry out clustering. While these techniques can be used to cluster textual data, they do not work particular well due to various reasons. First, the dimensionality of textual data is very large whereas the data is usually sparse. Secondly, the works are correlated with one another, meaning that the number of principal components are smaller than the feature space. Thirdly, different document would contain very different words, thus normalization of document representation is essential to perform accurate clustering.

Text clustering algorithms can be divided in to a variety of different types of algorithms namely – partitioning, agglomerative and standard parametric modelling based clustering algorithms. Different representations of the textual data are pivotal to different cases of clustering algorithms since each have their own advantages and disadvantages. The trade-offs are there in terms of effectiveness and efficiency. In this review, we first have a look at different methods for feature selection and transformation for text clustering. We will then discuss about some of the common clustering algorithms for distance based similarity. We will then briefly discuss upon

the methods for clustering patterns and phrases. Finally, we end with conclusion and summary.

Methods for Feature Selection and Transformation

In this section, we will discuss a few methods which are available for the feature selection process for textual documents. **Document Frequency based selection** is one which simplest approached which uses the document frequency to remove irrelevant features. There are certain *stop words* available in textual data which are typically very common words and are do not provide any discriminative advantage from the perspective of clustering. While the TF-IDF weighing is capable of reducing the contribution of such frequent words, it is a better approach to completely remove them and filter the textual data before going ahead with clustering. The **Term Strength** is a more aggressive approach for the similar process of stop word removal as mentioned above. It measures how informative a word is in identifying two related documents. The term strength can either be defines as a probability or in some cases as a random sampling of pairs of the related documents. Such kind of an approach required no training data for the selection of features.

Another approach used for feature selection is the **Entropy based Ranking** [2]. In this approach, the term quality is measured through the entropy reduction once the term is reduced. However, the calculation of entropy reduction for each word is more computationally extensive is impractical for large corpus which contains many terms. The **Term Contribution** approach is based on the idea that the clusters formed by text clustering algorithms are highly dependent on document similarity. This the term contribution and document similar can be viewed as similar. However, similar to the previous case, since the normalized frequencies in all the pair of documents need to be determined for the term contribution, this approach is also computationally expensive. At the same time, it is seen to favour high frequency words which do not contribute to the discriminative ability in clustering.

While the above approached are used to extract features from the textual data, also known as dimensions. Methods such as LSI i.e. **Latent Semantic Indexing** are used to remove and filter the noisy dimensions which act as obstacles in the clustering process. LSI is closely related to Principal Component Analysis (PCA). Except that it uses an approximation of the covariance matrix, which is appropriate for the sparse nature of textual data. Methods like **Concept Decomposition** [3] are also used to reduce the noisy dimensions in the textual data using standard clustering techniques. In the second phase of clustering, the reduced representation is used which is obtained from the first phase of clustering.

Clustering Algorithms

There are **Distance-based Clustering** Algorithms which use similarity functions to measure the closeness between textual objects. Most commonly used similarity function is the cosine similarity. Then there are **Agglomerative Hierarchal clustering** which support searching methods as they create a tree like hierarchy. The idea behind it is to successively merge groups based on their similarity. Typically, different methods of merging the documents would lead to different agglomerative clustering.

Distance based partitioning algorithms are also uses, like the k-medoids and k-means clustering algorithm. Both these algorithms make use of k representatives around which the clusters are built. They work on iterative approach to successively improve the clustering in each of the iterations. However, the disadvantage is that they hugely depend in the starting data and could to lead to very different results depending on the starting seed. While Hierarchal clustering are more robust but they turn out to be not very efficient, at the same time, k-means are more efficient but may not be very effective because of their tendency to rely on seeds. Thus, there are also **Hybrid** approaches which use both the advantages of hierarchal and partitioning algorithms [4].

Probabilistic document clustering can be achieved through topic modelling. The idea behind this is to create a probabilistic generative model. The two methods which are used for probabilistic modelling are Probabilistic Latent Semantic Indexing (PLSI) [5] and Latent Dirichlet Allocation (LDA) [6]. LDA is generally more extensively used since it is less susceptible to overfitting than PLSI. In some application, prior knowledge is available which can be used for clustering of the data, these case **Semi-Supervised Clustering.** They also form a natural bridge between clustering and classification.

Conclusion

In this brief technology review, we first look at different techniques feature selection as well as to represent textual data and then discussed few popular text clustering techniques. In the recent years, main focus has been on dynamic applications, such as in social media networks and heterogeneous applications, where text is available with a mix of other data. The field of text clustering is a bit too vast to cover in a brief technology review. The main purpose of this review was to provide an overview of the techniques used in this area.

References

Aggarwal C.C., Zhai C. (2012) A Survey of Text Clustering Algorithms. In: Aggarwal C., Zhai C. (eds) Mining Text Data. Springer, Boston, MA. https://doi.org/10.1007/978-1-4614-3223-4_4

[1] P. Andritsos, P. Tsaparas, R. Miller, K. Sevcik. LIMBO: Scalable Clustering of Categorical Data. EDBT Conference, 2004.
[2] M. Dash, H. Liu. Feature Selection for Clustering, PAKDD Confer- ence, pp. 110–121, 1997

[3] I. Dhillon, D. Modha. Concept Decompositions for Large Sparse Data using Clustering, 42(1), pp. 143–175, 2001.

[4] D. Cutting, D. Karger, J. Pedersen, J. Tukey. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. ACM SIGIR Conference, 1992.

[5] T. Hofmann. Probabilistic Latent Semantic Indexing. ACM SIGIR Conference, 1999.

[6] D. Blei, A. Ng, M. Jordan. Latent Dirichlet allocation, Journal of Machine Learning Research, 3: pp. 993–1022, 2003.