K-Means Document Clustering using Vector Space Model

R. Malathi Ravindran and Dr. Antony Selvadoss Thanamani

Abstract— Document Clustering is the collection of similar documents into classes and the similarity is some function on the document. Document Clustering need not require any separate training process and manual tagging group in advance. The documents used in the same clusters are more similar, while the documents used in different clusters are more dissimilar. It is one of the familiar technique used in data analysis and is used in many areas including data mining, statistics and image analysis. The traditional clustering approaches lose its algorithmic approach when handling high dimensional data. For this, a new K-Means Clustering technique is proposed in this work. Here Cosine Similarity of Vector Space Model is used as the centroid for clustering. Using this approach, the documents can be clustered efficiently even when the dimension is high because it uses vector space representation for documents which is suitable for high dimensions.

Keywords— Data Mining, Document Clustering, High Dimensional Data, Vector Space Model, K-Means Clustering, Cosine Similarity

I. INTRODUCTION

Document Clustering partitions the documents into groups (called clusters) such that the similarity among the documents of the same group is maximized and the similarity among the documents of different groups is minimized[1]. Document Clustering concept is used in various areas like information retrieval, text mining etc. Initially, it was investigated to improve the precision in the information retrieval systems and concludes it as an efficient way to find the nearest neighbor of a document[3]. Although for a long period of time, research has been carried out in Document Clustering. It is still complicated to solve the problem. The existing information retrieval systems like Boolean model works well when the collection of data is exactly known and for the precise documents. In Boolean model, a document is represented as a set of keywords. Queries are represented as Boolean expressions of keywords. They are connected by the operators AND, OR, and NOT to indicate the scope of these operators including the brackets. List of documents are the outputs and there will be no partial matches or ranking. The Boolean model is very rigid search engine model. The ranking of documents is very difficult because the matched documents satisfy only the query to the same degree. The formation of query for most of the users are also difficult and if the size of the document collection is increased the complexity for retrieving the information will also be increased. The challenges faced in the Document Clustering are selecting appropriate features of the documents that should be used for clustering, selecting the appropriate similarity measure among the documents, selecting the appropriate clustering method utilizing the above mentioned similarity measure, implementing the clustering algorithm in an efficient way to make it simple in terms of required CPU and memory resources to find a way for assessing the quality of the performed clustering [2]. Since K-Means Clustering efficiently clusters only the numeric values, clustering the documents needs a very different approach that should be efficient and effective. In this work, vector space K-Means Clustering is used for clustering the documents with high dimensions.

II. LITERATURE REVIEW

In Erk, K., & Padó, S. [2] “A structured Vector Space Model for word meaning in context”, the computing vector space representations for the meaning of word occurrences are considered. It is recommended that the existing models for this task do not take syntactic structure sufficiently into account and they present a novel structured Vector Space Model that addresses the above issues by incorporating the selectional preferences for words' argument positions. So we can integrate syntax into the computation of word meaning in context and also the model performs at and above the state of the art for modeling the contextual adequacy of paraphrases. J.Mitchell, M.Lapata [3] “Vector-based models of semantic composition” proposes a framework for representing the meaning of phrases and sentences in vector space. The experimental results demonstrate that the multiplicative models are superior to the additive alternatives when compared against human judgments. K.Erk [4] “A simple, similarity-based model for selectional preferences” proposes a new, simple model for the automatic induction of selectional preferences, using corpus-based semantic similarity metrics. Focusing on the task of semantic role labeling, they compute selectional preferences for semantic roles. Final result shows that the similarity-based model gives lower error rates. Michael Steinbach et al., [5] “A Comparison of Document Clustering techniques” compares the two main approaches to Document Clustering, Agglomerative Hierarchical Clustering and K-Means. Their results indicate that the bisecting K-Means technique is better than the standard K-Means approach and as good as or better than the hierarchical approaches that they tested for a variety of cluster evaluation.
metrics. In Pankaj Jajoo [6] “Document Clustering”, the aim of the thesis is to improve the efficiency and accuracy of Document Clustering. The initial approach shows an improvement in the graph partitioning techniques which is used for Document Clustering. In this paper, heuristic is used for processing the graph and also the standard graph partitioning algorithm is applied which leads to improve the quality of clusters to a higher level. The next approach is completely a different approach. Here initially the words are clustered and then the documents are clustered. It leads to decrease the noise in data and it also improves the quality of the clusters. In the above mentioned approaches for improving the quality and efficiency, the parameters can be changed with respect to the dataset. In I.N.Bharambe and R.K.Makhijani [7] “Design of Search Engine using Vector Space Model for Personalized Search”, the previous search engines results are discussed. The results produced by using the search engines are up to some extent but also having some drawback. To overcome this, some alternate reranking methods are also to be considered by using some of the statistical methods such as Latent Semantic Analysis.

III. RANKING THE DOCUMENTS

For clustering algorithms documents are represented using the vector-space model. This paper proposed the Vector Space Model representation of documents, also known as bag of words model to reduce the dimension of a data and to rank the documents according to its relevance. Vector Space Model is good when it is compared to Boolean model where expressing complex user requests are difficult, handling of high dimensional data is difficult and ranking the output is difficult. Vector Space Model is a statistical model of representation where the document is represented by a bag of words.

The core idea is to present every document as a vector of some weighted word frequencies. For doing this, below given parsing extraction steps are necessary.

- To ignore, collect all unique words from the entire document set.
- Eliminate non-content bearing stop words.
- For every document, the number of occurrences of each word is counted.
- After the elimination suppose w unique words remain, assign a separate identifier between 1 and w to each remaining word, and a separate identifier between 1 and d to each document.

Assume that after preprocessing, there are s distinct terms and are denoted by the vocabulary or index terms. The vector space is formed by these “orthogonal”.

Dimension = s = | vocabulary |

Each term, $x_i$ in a document or query, $y_j$ is given a real-valued weight, $w_{xy}$. Queries and documents are represented as s-dimensional vectors:

$$d_y = (w_{1y}, w_{2y}, \ldots, w_{sy})$$

In the Vector Space Model, a collection of $n$ documents can be represented by a term-document matrix. The “weight” of a term in the document is corresponding to an entry in the matrix. If it is zero, the term has no significant in the document else it does not exist in the document. With respect to the given weighting model, terms are weighted which may include global weighting, local weighting or both.

$$T_1 \quad T_2 \quad \ldots \quad T_t$$

$$D_1 \quad w_{11} \quad w_{12} \quad \ldots \quad w_{1l}$$

$$D_2 \quad w_{21} \quad w_{22} \quad \ldots \quad w_{2l}$$

$$\vdots \quad \vdots \quad \vdots \quad \ddots \quad \vdots$$

$$D_n \quad w_{nl} \quad w_{2n} \quad \ldots \quad w_{ml}$$

On using local weights, the terms weights are represented as term frequencies, $sf$. On using global weights, the term’s weight is given by $IDF$ values. In this work Salton's Vector Space Model is used which incorporates both global and local weights. So it is called as $sf \times IDF$ weighting

$$a \text{ term's weight} = sf \times IDF$$

The terms occurred more frequently are very important. It means more indicative of the topic.

$$f_{xy} = \text{frequency of term } x \text{ in document}$$

To normalize term frequency ($sf$) across the entire corpus:

$$sf_{xy} = f_{xy} / \max_x \{f_{xy}\}$$

Terms which are appearing in many other documents are represented as

$$df_x = \text{document frequency of term } x$$

$$= \text{number of documents containing term } x$$

$$xdf_x = \text{inverse document frequency of term } x,$$

$$= \log_2 (N / df_x)$$

$(N = \text{total number of documents})$

IDF provides low values for common words and high values for rare words.

For example

$$\log(10000/10000) = 0 ,$$

$$\log(10000/5000) = .301 ,$$

$$\log(10000/20) = 2.698 ,$$

$$\log(10000/1) = 4$$

A typical combined term importance indicator is $sf \times xdf$ weighting:

$$w_{xy} = sf_{xy} \times xdf_x = sf_{xy} \log_2 (N / df_x)$$

High weight is given to a term occurring frequently in the document but occasionally in the rest of the collection. Query vector is treated as a document and also sf-xdf is weighted. A document should be normalized by document length because of two reasons: long documents have higher term frequencies that have the same term that appears more often and more terms increase the number of matches between a query and a document.
IV. SIMILARITY MEASURE

To compute the degree of similarity between two vectors, a function called similarity measure is used. The similarity measure is used among the query and each document to rank the retrieved documents with the order of presumed relevance and to enforce some threshold. So the size of the retrieved set is controlled. Similarity between query \( z \) and vectors for the document \( dx \) is calculated as the vector inner product

\[
sim(dy, z) = wxy \cdot wxz \quad \text{where } wxy \text{ is the weight of term } x \text{ in document } y \text{ and } wxz \text{ is the weight of term } x \text{ in the query.}
\]

The inner product for binary vector is the number of matched query terms in the document. The sum of the products of the matched terms' weight is called the inner product for weighted term vectors.

Euclidean distance is a choice to find the vector space proximity but since it results in large value, we use cosine measure for ranking the documents. Cosine is a monotonically decreasing function for the interval \([0^\circ, 180^\circ]\). It lessens the impact of long documents.

V. COSINE SIMILARITY MEASURE

The cosine of the angle between two vectors is measured by Cosine Similarity measure.

A document vector's similarity to a query vector = cosine of the angle between them

\[\cos(\theta) = \frac{\sum_{i=1}^{n} w_{xy} \cdot w_{xz}}{\sqrt{\sum_{i=1}^{n} w_{xy}^2} \cdot \sqrt{\sum_{i=1}^{n} w_{xz}^2}}\]

Cosine is a normalized dot product. Documents ranked by a decreasing cosine value \( \cos(\theta) \) = 1 when \( d = q \), \( \cos(\theta) = 0 \) when \( d \) and \( q \) share no terms. \( S \) contains the set of documents and their corresponding representations, the centroid vector \( c \) is defined as

\[c = \frac{1}{|S|} \sum_{d \in S} d\]

It is calculated by finding out the average weight of different terms in the documents of \( S \). Analogously to documents, the similarity among the two centroid vectors and among a document and a centroid vector are calculated by using the cosine measure, i.e.,

\[\cosine(d, c) = (d \cdot c) / \| d \| \| c \| \]
\[\cosine(c_1, c_2) = (c_1 \cdot c_2) / \| c_1 \| \| c_2 \|

To find the closest document centroid to a given document, the cosine measure is used in K-Means Clustering. Whenever the median is used as the centroid for K-Means Clustering, use the mean that is very easy to compute than the median. It has more mathematical properties.

For example, calculating the dot product between a document and a cluster centroid is equivalent to calculating the average similarity between that document and all the documents that comprise the cluster the centroid represents.

VI. K-MEANS CLUSTERING ALGORITHM USING COSINE SIMILARITIES

The proposed K-Means Clustering algorithm uses Cosine Similarity measure of vector representation as the distance measure to cluster the documents. K-Means is one of a partitional clustering algorithm. It performs iterative relocation to partition a dataset into clusters, locally minimizing the average squared distance between the cluster centers and the data points. The objective of K-Means is to reduce the sum of distance from every data point to its closest center. Let

\[\{c_x\}_{x=1}^{k}\]

be the centroids of \( k \) clusters. Let \( \eta(.) \) be the assignment function. Then \( \eta(x) = y \) means the \( x^{th} \) item is assigned to the \( y^{th} \) cluster. K-Means reduces the following objective

\[n(x) = \arg \min_{e,n} \sum_{i=1}^{n} |e_x - c_{\eta(x)}|^2\]

Lowering the objective function leads to more compact clusters, where each item gets closer to its cluster centroid. Finding the global optima for the K-Means objective function is an NP-complete problem [GJW82].

**Algorithm K – Means**

**Input**: A set of data items

**Output**: A disjoint set of clusters

1. Initialise clusters: cluster centroids \( \{c_x\}_{x=1}^{k} \) are selected at random
2. Repeat until coverage
   2(a). Assign data items following the nearest neighbor rule, that is
   \[n(x) = \arg \min_{e,n} \sum_{i=1}^{n} |e_x - c_{\eta(x)}|^2\]
   2(b). Update Centroids
   \[c_y^{(x)} = \frac{\sum_{x=\eta(x)=y} e_x}{n_y}\]
   2(c). \( t = t + 1 \)

In the proposed K-Means Clustering algorithm, Cosine Similarity normally find the distance from every document to its most close center. Since the document is represented in
vector space, it is highly efficient to cluster with the high dimensional data. Before text data can be analyzed, the popular Vector Space Model ("bag-of-words") represent raw text data as high-dimensional vectors, where each dimension of the vector is a unique term. The adoption of Cosine Similarity is based on the observation that text vectors have only non-negative entries, and the high-dimensional text vectors with directional properties which means their direction is highly discriminative than the length of the vectors. In this approach, each data item and cluster centroids are normalized and are represented as unit-length vectors.

\[ K - \text{Means} = \sum_{i=1}^{k} \sum_{x \in c_i} e_i^x \cdot \mu_n \]

The algorithm is computationally efficient for sparse high dimensional data vectors, which are very common in domains like text clustering.

VII. EXPERIMENTAL RESULTS

Consider the following collection of 5 documents

\[ D_1 = \text{Health tips for healthy life} \]
\[ D_2 = \text{Tips for long life} \]
\[ D_3 = \text{Nutritious food} \]
\[ \text{Query Q} = \text{Healthy food for life} \]

The table shows the term frequency, document frequency, weightage of each term and the query. The terms in the documents are sorted in alphabetical order and term frequency is tabulated. The document frequency \( df \) is calculated according to the number of occurrences of the term in all documents. The frequency of inverse document is also calculated and it gives high values to rare words and low values to common words. Likewise, document’s weightage and the query are also calculated.

<table>
<thead>
<tr>
<th>Terms</th>
<th>Count</th>
<th>( d_f )</th>
<th>( \text{idf} )</th>
<th>Q</th>
<th>( D_1 )</th>
<th>( D_2 )</th>
<th>( D_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Health</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.4771</td>
<td>0</td>
<td>0.4771</td>
<td>0</td>
</tr>
<tr>
<td>Tips</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0.1760</td>
<td>0</td>
<td>0.1760</td>
</tr>
<tr>
<td>For</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.4771</td>
<td>0.4771</td>
<td>0.4771</td>
</tr>
<tr>
<td>Healthy</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.1760</td>
<td>0.1760</td>
</tr>
<tr>
<td>Life</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.4771</td>
<td>0.4771</td>
<td>0.4771</td>
</tr>
<tr>
<td>Long</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.4771</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Nutritious</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.4771</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Food</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.4771</td>
<td>0.4771</td>
</tr>
</tbody>
</table>

Length Normalization

\[ D_1 = \sqrt{(0.4771)^2 + (0.1760)^2 + (0.1760)^2 + (0.4771)^2 + (0.1760)^2} = 0.7403 \]
\[ D_2 = \sqrt{(0.1760)^2 + (0.1760)^2 + (0.1760)^2 + (0.4771)^2} = 0.5661 \]
\[ D_3 = \sqrt{(0.4771)^2 + (0.4771)^2} = 0.6747 \]

Query Q’s Length

\[ Q = \sqrt{(0.1760)^2 + (0.1760)^2 + (0.1760)^2 + (0.4771)^2} = 0.7191 \]

Cosine Similarity

\[ \cos e D_1 = \frac{Q \cdot D_1}{\|Q\| \cdot \|D_1\|} = \frac{0.2895}{(0.7191 \times 0.7403)} = 0.5438 \]
\[ \cos e D_2 = \frac{Q \cdot D_2}{\|Q\| \cdot \|D_2\|} = \frac{0.0619}{(0.7191 \times 0.5661)} = 0.1520 \]
\[ \cos e D_3 = \frac{Q \cdot D_3}{\|Q\| \cdot \|D_3\|} = \frac{0.2276}{(0.7191 \times 0.6747)} = 0.4691 \]

Finally, in the point of similarity values, the resultant documents are listed as D1, D3, and D2.

VIII. OBSERVATIONS

This experiment illustrates frequent terms such as "a", "in", and "of" tend to receive a low weight - a value of zero in this case. Thus, this model correctly finds that the occurrence of very common terms in many documents in a collection are not good discriminators of relevancy. This reasoning is based on global information; ie., the IDF term. Precisely, that is the reason why this model looks better than term count model. In proposed K-Means Clustering algorithm, this value is used as the similarity measure to cluster documents having high dimensional data.

IX. CONCLUSION

In this paper, a new approach to K-Means Clustering is proposed to cluster the documents with high dimensional data. The principal challenge in extending cluster analysis to high dimensional data is to overcome the "curse of dimensionality". 
In this work, Vector Space Model is used to represent the documents with high dimensional data. The vector similarity function is a cosine function, whose returned value is [0,1]. The greater the returned value is, the greater the similarity is. This value is used as distance measure in the proposed K-Means Clustering. Since the demand for cluster computing will grow in future, data analytics such as Map Reduce and Hadoop is considered as the future enhancement of this work.

REFERENCES


BIOGRAPHY

Ms. R. Malathi Ravindran received her MCA degree from Madurai Kamarajar University, Madurai, India. She has completed her Master of Philosophy in Periyar University, Salem. Presently she is working as an Assistant Professor of MCA in NGM College, Pollachi. She has 10 years of teaching experience. Her area of interest includes data Mining. Now she is pursuing her Ph.D in Computer Science in the Research Department of Computer Science, NGM College, Pollachi under Bharathiar University, Coimbatore, India.

Dr. Antony Selvadoss Thanamani is presently working as Associate Professor and Head, Research Department of Computer Science, NGM College, Pollachi, Coimbatore, India. He has published more than 100 papers in international/national journals and conferences. He has authored many books on recent trends in Information Technology. His areas of interest include E-Learning, Knowledge Management, Data Mining, Networking, Parallel and Distributed Computing. He has to his credit 25 years of teaching and research experience. He is a senior member of International Association of Computer Science and Information Technology, Singapore and Active member of Computer Science Society of India, Computer Science Teachers Association, NewYork.