COMPARATIVE STUDY ON TEXT DOCUMENT CLUSTERING ALGORITHMS BASED ON LATENT SEMANTIC INDEXING

R.Jensi

Department of Computer Science and Engineering, Dr.Sivanthi Aditanar College of Engineering,Tiruchendur

ABSTRACT

In this paper, a comparative analysis of text document clustering algorithms based on latent semantic indexing dimension reduction technique is done. In recent days, the huge amount of textual information is available in electronic form. In order to bring out the interesting patterns from very large text databases, several heuristic algorithms have been developed and still it seems to be quite challenging. Text document clustering is the fastest growing research area for grouping enormous text documents in such a way that documents within a cluster have high intra-similarity and low inter-similarity to other clusters. One of the major issues in document clustering is high dimensionality. Latent Semantic Indexing (LSI) is used with clustering algorithms to analyze the performance of text document clustering algorithms. The experimental results on the dataset constructed from Reuters21578 collections show that dimensionality reduction can improve clustering performance with respect to the computation time and average fitness of the clustered documents.

KEYWORDS

Latent Semantic Indexing, Principal Component Analysis, text clustering algorithm, Text mining

1. INTRODUCTION

With the large amount Information on the Web that is present in the form of text documents (formatted in HTML), many research projects were proposed on how to organize such information so that end users browse or find the information they want efficiently and accurately. Text data mining techniques play an important role in organizing such text documents in an effective manner.

Text mining shares many concepts with traditional data mining methods. Many data mining techniques can uncover inherent structure in the underlying data. Among those data mining techniques, one such technique is clustering. Clustering is an unsupervised pattern classification technique which is defined as group *n* objects into *m* clusters without any prior knowledge. Clustering produces clusters which exhibit high intra-cluster similarity and low inter-cluster similarity [1]. Text document clustering methods attempt to partition the set of documents into groups in such a way that each group represents some topic that is different from the other groups [2].

In general, the major clustering methods can be classified into partitioning, hierarchical, density-based, grid-based and model-based methods [3].Among these, partitioning and hierarchical clustering algorithms are mainly used for text document clustering. The partition clustering algorithm organizes the set of n objects into k partitions ($k \leq n$), where each partition represents a cluster. The clusters are formed to optimize an objective partitioning criterion, such as

dissimilarity function based on distance is used. The resulting clusters have the following property (i.e) the objects within a cluster are similar, whereas objects of different clusters are dissimilar.

A hierarchical clustering method work by grouping data objects into a tree of clusters. Based on the hierarchical decomposition formation, hierarchical clustering methods can be further classified as either agglomerative or divisive. Agglomerative hierarchical clustering algorithm is a bottom-up approach in which each object is considered as a singleton cluster and then successively merge pairs of clusters until all objects belong to single cluster. On the other hand, divisive hierarchical clustering algorithm follows top-down approach in which initially all objects belong to single cluster and proceeds by splitting until individual objects are reached. Recently, many studies showed that hierarchical clustering algorithms do not contain any provision for the reallocation of entities.

Fuzzy c-means (FCM) is a clustering technique in which a set of data items is partitioned into n clusters with every data point in the dataset belonging to every cluster to a certain degree. A certain data point that lies close to the center of a cluster will have a high degree of belonging or membership to that cluster and another data point that lies far away from the center of a cluster will have a low degree of belonging or membership to that cluster.

The partitional clustering algorithms such as K-Means, Spherical K-Means (SK-Means) [9], Principal Direction Divisive Partitioning (PDDP) and Fuzzy c-means (FCM) are compared for their performance in clustering text documents.

To improve the performance of text documents clustering in terms of time, Dimensionality Reduction (DR) is used. Dimensionality reduction is a technique by which high-dimensional data is transformed into a meaningful representation of reduced dimensionality. The reduced dimensionality of data is the minimum number of parameters needed for the observed properties of the data. Dimensionality reduction is important in many domains, since it facilitates classification, visualization, and compression of high-dimensional data, by mitigating the curse of dimensionality and other undesired properties of high-dimensional spaces. Over the last decade, a large number of new (linear) techniques for dimensionality reduction have been proposed. The main advantage of using dimensionality reduction is more economical representation of data, and better semantic representation. In this paper, one technique is used for dimensionality reduction. It is Singular Value Decomposition (SVD).

The basic outline of this paper is as follows:
Section II provides an outline of Document representation and term weighting scheme for clustering. Section III discusses the clustering algorithms used. Section IV presents dimension reduction techniques used. Section V measures cluster quality that will be used as the basis for our comparison of different document clustering techniques. Section VI gives the details of the test data used, the results and discussions and finally conclusion is given in section VII.

2. DOCUMENT REPRESENTATION AND TERM WEIGHTING SCHEME

Fig.1 shows the overall processing of text documents.

2.1. Text document preprocessing

The preprocessing basically consists of a process to optimize the list of terms that identify the collection. The first process is to strip all formatting from the article, (i.e) remove capitalization, punctuation, and extraneous markup.

Then the stop words are removed. Stop words are the words that don't carry any semantic meaning. Using a list of stop words, Stop words may be eliminated. Removing stop words greatly reduces the amount of noise in our collection, as well as reduces computational time. Removing these stop words leaves us with an abbreviated version of the article containing content words only.

The next process is to stem a word. Stemming is the process of converting derived words to their root form. It is language-specific. For English documents, Porter stemmer algorithm is used to remove common endings from words, leaving behind an invariant root form. Thus the performance of document retrieval can be improved.

2.2. Text Document Encoding: Term-Document Matrix (TDM)

Let **D** = (D_1, D2, . . . ,D_N) be a collection of documents and **T**=(T1, T2, . . . , T_M) be the complete vocabulary set of the document collection **D**, where N is the number of documents and M is the number of unique terms[4,9]. Text documents can be represented in several ways. In this paper vector space model is applied, widely used in IR and text mining, to represent the text documents. In this model each document D_i is represented by a point in an m dimensional vector space, D_i = (w_{i1}, wi2, . . . ,w_{im}), i = 1, . . . , N, where the dimension is equal to the number of terms in the document collection. Each component of such a vector reflects a term connected with the given document. The value of the component depends on the degree of relationship between its associated term and the respective document. Many schemes have been proposed for measuring this relationship. Term weighting is the process of calculating the degree of relationship (or association) between a term and a document. One of the more advanced term weighting schemes is the tf-idf (term frequency-inverse document frequency) [4]. The tf-idf scheme aims at balancing the local and the global term occurrences in the documents. In this scheme, w_{ij} can be calculated as

$$w_{ij} = n_{ij} \times \log\left(\frac{N}{n_j}\right) \quad (1)$$

where n_{ij} is the term frequency, and n_j denotes the number of documents in which term T_j appears. The term log (N/n_j), which is often called the idf factor, defines the global weight of the term T_j. Indeed, when a term appears in all documents in the collection, then n_j = N, and thus the balanced term weight is 0, indicating that the term is useless as a document discriminator. The idf factor has been introduced to improve the discriminating power of terms in the traditional clustering.

3. DIMENSION REDUCTION TECHNIQUES

Dimension reduction is important in cluster analysis, because it reduces the high dimensional data and the computational cost, as well as provides users with a clear picture and visual examination of the data of interest. The goals of dimension reduction methods are to reduce the number of predictor components and to help ensure that these components are independent.

In this paper, one dimension reduction technique, Latent Semantic Indexing, is used.

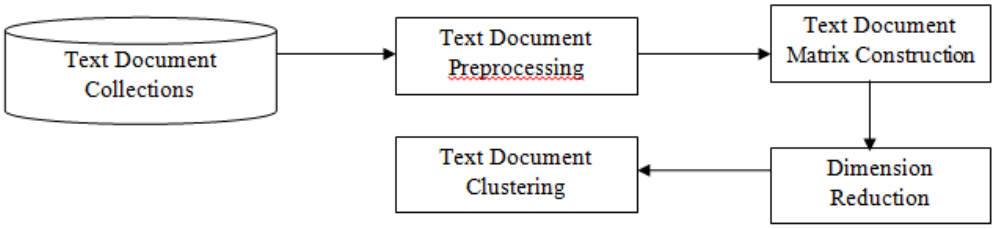

Figure 1. Main process of text mining

3.1. Latent Semantic Indexing (LSI)

Latent Semantic Indexing uses a mathematical technique called Singular value decomposition (SVD) to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text. It is an indexing and retrieval method. LSI follows the basic principle that words that are used in the same contexts be likely to have similar meanings. A key feature of LSI is its ability to extract the conceptual content of a body of text by establishing associations between those terms that occur in similar contexts.

After a term-document matrix X (m×n) is constructed, such that there are m distinct terms and n documents. The Singular Value Decomposition of X is given by

$$X=USV^T \tag{2}$$

where U and V are the matrices of the left and right singular vectors. D is the diagonal matrix of singular values. LSI approximates X with a rank *k* matrix.

$$X_k=U_kS_kV^T_k \tag{3}$$

where U_k is composed of the first *k* columns of the matrix U and V^T_k is comprised of the first *k* rows of matrix V^T. S_k=diag(s_1,…..s_k) is the first *k* factors.

When LSI is used for text document clustering, a document D_i is represented by [5]

$$D_i=D^T_iU_k \tag{4}$$

Next the text corpus can be ordered by another representation of document-term matrix D(n×m) and the corpus matrix is organized by

$$C=DU_k \tag{5}$$

4. CLUSTERING ALGORITHMS

Clustering is the process that is used to group and divide underlying data based on similarities and dissimilarities. It discovers both the dense and the sparse regions in a data set. In this paper, four clustering algorithms are used to compare the performance of clustering. The four clustering algorithms are *K-Means, Spherical K-Means, Principal Direction Divisive Partitioning* and *Fuzzy C-Means.*

4.1. K-Means

K-means is the most widely used clustering technique; it belongs to the class of iterative centroid-based divisive algorithm. The algorithm tries to determine k partitions that minimize the squared-error function. The k-means method can be applied only when the mean of cluster is defined. The k-means algorithm for partitioning is based on each cluster's center which is represented by the mean value of the objects in the cluster [6].

Finally, this algorithm optimizes (minimize/maximize) an *objective function*, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \tag{6}$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the *n* data points from their respective cluster centres.

It takes the input parameter, k, number of clusters and a dataset D containing n objects, it partitions the dataset into k clusters by using the following steps:

1) Randomly select 'c' cluster centers.

2) Calculate the distance between each data point and cluster centers.

3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers.

4) Recalculate the new center using

$$V_i = \frac{1}{c_i} \sum_{j=1}^{ci} d_j \tag{7}$$

where c_i is the number of data points in the i^{th} cluster, d_j is the document vector that belongs to cluster c_i and V_i is the new centroid vector.

5) Recalculate the distance between each data point and new obtained cluster centers.

6) If no data point was reassigned then stop, else repeat from step (3).

4.2. Spherical K-Means

The standard spherical k-means problem is to minimize

$$\sum_i \left(1 - \cos(x_i, p_{c(i)})\right) \tag{8}$$

Over all assignments *c* of documents *i* to cluster ids *c(i)* ϵ *{1,.....,k}* and over all prototypes *p1,.....,pk* in the same feature space as the feature vectors x_i representing the documents. With the memberships μ_{ij} of documents *i* to classes *j* defined by

$$\mu_{ij}=\begin{cases}1, if c(i)=j\\ 0, otherwise\end{cases} \quad (9)$$

and the membership matrix M=[μ_{ij}], the standard spherical k-means can be formulated as minimizing

$$\phi(M,P)=\sum_{i,j}\mu_{ij}(1-\cos(x_i, p_j)) \quad (10)$$

over all membership matrices M with unit row sums and prototype matrices.

4.3. Principal Direction Divisive Partitioning

PDDP is a representative of the non-iterative technique based upon the Singular Value Decomposition (SVD) of a matrix built from the data set [10]. The algorithm is as follows:

Step 1: Compute the centroid W as

$$W=\frac{1}{N}\sum_{j=1}^{N}Xj \quad (11)$$

Step 2: Compute the auxiliary matrix $\tilde{M}=M-we$, where e is a N-dimensional row vector of ones, namely e=[1,1,1,1,……,1].

Step 3: Compute the singular value decomposition (SVD) of $\tilde{M}, \tilde{M}=U\Sigma V^T$, where $\sum$ is a diagonal P×N matrix and U and V are orthogonal unitary square matrices having dimension P×P and N×N respectively.

Step 4: Take the first column vector of U, say u=U and divide M=[x_1, x_2… x_n] into sub clusters M_L and M_R according to the following rule,

$X_i \in M_L$ if U^T (x-w) ≤ 0
$X_i \in M_R$ if U^T (x-w) ≤ 0

4.4. Fuzzy C-Means

Fuzzy C-Means algorithm assigns membership to each data point corresponding to each cluster center on the basis of distance between the cluster center and the data point. If more number of data is near to the cluster center means its membership towards the particular cluster center is more. Obviously, summing up membership of each data point should be equal to one. Membership and cluster centers are updated after each iteration [11-14].

Main objective of fuzzy c-means algorithm is to minimize:

$$J(U,V)=\sum_{i=1}^{n}\sum_{j=1}^{c}(\mu_{ij})^m\left\|x_i-v_j\right\|^2 \quad (12)$$

where, $'||x_i-v_j||'$ is the Euclidean distance between i^{th} data and j^{th} cluster center.

The steps for Fuzzy c-means clustering algorithm are as follows:

Let X = {x_1, x_2, x_3 ..., x_n} be the set of data points and V = {v_1, v_2, v_3 ..., v_c} be the set of centers.

1) Randomly select *'c'* cluster centers.

2) Calculate the fuzzy membership *'μ_{ij}'* using:

$$\mu_{ij} = 1/\sum\left(\frac{d_{ij}}{d_{ik}}\right)^{\left(\frac{2}{m}-1\right)} \quad (13)$$

3) Compute the fuzzy centers *'v_j'* using:

$$v_j = \frac{\left(\sum_{i=1}^{n}(\mu_{ij})^m x_i\right)}{\left(\sum_{i=1}^{n}(\mu_{ij})^m\right)}, \forall j = 1,2,\ldots.c \quad (14)$$

4) Reiterate step 2) and 3) until the minimum *'J'* value is achieved or

$||U^{(k+1)} - U^{(k)}|| < \beta$.

where,

'k' is the iteration step.

'β' is the termination criterion between [0, 1].

'$U = (\mu_{ij})_{n*c}$' is the fuzzy membership matrix.

'J' is the objective function.

5. EVALUATION METHOD OF THE TEXT CLUSTERING

To measure the quality of the clustering, F-measure, Purity and Entropy are used. These measures are called external quality measures because external information about data is available; it is typically in the form of externally derived class labels for the data objects [1]. When relating to Information retrieval, each cluster is considered as the result of a query, whereas each pre-defined set of documents can be considered as the desired set of documents for that query.

If n_i is the number of members of class i, n_j is the number of members of cluster j, and n_{ij} is the number of members of class i in cluster j then the F-measure, Purity and Entropy can be defined [15] as follows:

5.1. F-measure

The F-measure is harmonic combination of the precision and recall values used in information retrieval. *precision* $P(i,j)$ and *recall* $R(i,j)$ can be calculated as

$$P(i,j) = \frac{n_{ij}}{n_j} \quad . \quad (15)$$

$$R(i,j) = \frac{n_{ij}}{n_i} \quad (16)$$

The corresponding *F-measure* $F(i,j)$ is defined as

$$F(i,j) = \frac{2 * P(i,j) * R(i,j)}{P(i,j) + R(i,j)} \quad (17)$$

Then the *F-measure* for the whole clustering result is defined as

$$F = \sum_i \frac{n_i}{n} \max_j (F(i,j)) \quad (18)$$

where n is the number of documents in the collection. In general, the larger value of the F-measure provides the better clustering result.

5.2. Purity

The purity of a cluster is defined as the fraction of the cluster corresponding to the largest class of documents assigned to that cluster. Thus the purity of a cluster j is defined as

$$Purity(j) = \frac{1}{n_j} \max_i (n_{ij}) \quad (19)$$

The overall purity of a clustering is a weighted sum of the cluster purities and is defined as

$$Purity = \sum_n \frac{n_j}{n} Purity(j) \quad (20)$$

In general, the larger the purity value gives the better clustering result.

5.3. Entropy

The Entropy of a cluster is defined as the degree to which each cluster consists of objects of a single class. The entropy of a cluster j is calculated using the standard formula,

$$e_j = -\sum_{i=1}^{L} p_{ij} \log p_{ij} \quad (21)$$

where L is the number of classes and p_{ij} is the probability that a member of cluster j belongs to class i.

The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster. Thus the total entropy e is defined as

$$e = \sum_{j=1}^{k} \frac{n_j}{n} e_j \quad (22)$$

where k is the number of clusters and n is the total number of documents in the corpus.

6. EXPERIMENTAL RESULTS

The experimental setup consisted of three data sets. The top most categories acq, crude, earn, interest, grain from the Reuters-21578 [7] are used.

Data Set 1: This dataset has 40 documents distributed over 4 classes.

Data Set 2: This dataset has 105 documents distributed over 5 classes.

Data Set 3: This dataset has 160 documents distributed over 4 classes.

These datasets are standard text datasets that are often used as benchmarks for document clustering. General characteristics of the datasets are summarized in TABLE 1.

The analysis is done using TMG toolbox [8] for matlab, available for text mining.

The spherical k-means and PDDP produced better clustering results than K-means algorithm which is simple and straightforward. Clustering quality is measured in terms of F-measure and purity as shown in TABLE 2 and TABLE 3 respectively.

The PDDP algorithm belongs to the class of singular value decomposition (SVD)-based data processing algorithms. PDDP provides a unique solution, given a data-set. PDDP is a SVD-based partitioning technique [10].

Fuzzy c-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. Multiple categories labels are removed. FCM clustering algorithm results of clustering more stable and accurate than the traditional k-means algorithm.

Table 1. EXPERIMENTAL DATASETS

S.No	Data Set	No. of documents	Number of terms		No. of Classes	Class Size
			Before preprocessing	After preprocessing		
1	Data Set1	40	1828	1303	4	Equal
2	Data Set2	105	3114	2122	5	Unequal
3	Data Set3	160	4010	2777	4	Equal

Table 2. F-MEASURE

S.No	Data Set	LSI: Number of factors	K-Means	Spherical K-Means	PDDP	FCM
1	Data Set1	40	0.3936 0.3936	0.4806 **0.6137**	0.6221 0.6221	0.4451 0.4451
2	Data Set2	100	0.3433 0.3447	**0.6707** 0.5505	**0.8267** 0.8267	0.4180 0.4180
3	Data Set3	120	0.3985 0.3451	**0.6575** **0.7077**	0.6890 **0.6890**	0.4479 0.4479

Table 3. PURITY

S.No	Data Set	LSI: Number of factors	K-Means	Spherical K-Means	PDDP	FCM
1	Data Set1	40	0.3250 0.3250	0.4750 **0.6000**	0.6250 0.6250	0.3750 0.3750
2	Data Set2	100	0.3143 0.3238	**0.6857** 0.5429	**0.8286** 0.8286	0.3238 0.3238
3	Data Set3	120	0.2687 0.3333	**0.6750** **0.7250**	0.7188 **0.7188**	0.4125 0.4125

Table 4. ENTROPY

S.No	Data Set	LSI: Number of factors	K-Means	Spherical K-Means	PDDP	FCM
1	Data Set1	40	0.5565 0.5565	0.4331 0.3646	0.3175 0.3175	0.5493 0.5493
2	Data Set2	100	0.6617 0.6595	0.3498 0.4821	0.2515 0.2515	0.6099 0.6099
3	Data Set3	120	0.5907 0.6511	0.3304 0.3078	0.2665 0.2665	0.5345 0.5345

Table 5. RUNTIME IN SECONDS FOR CLUSTERING DIFFERENT DIMENSIONALITY

S.No	Data Set	LSI: Number of factors	Run time(s)				
			K-Means	Spherical K-Means	PDDP	FCM	No reduction technique
1	Data Set1	40	0.001773	**0.001591**	0.023015	0.045474	0.930744
2	Data Set2	100	0.004124	**0.003849**	0.060597	0.047801	0.892202
3	Data Set3	120	0.004155	**0.003961**	0.049493	0.098193	3.394892

7. CONCLUSION

With the advancement of Web, huge amount of text document collections are available. In order to analyze and provide better relevant documents while searching, several techniques have been developed. In this paper, a comparative study on text document clustering algorithms based on latent semantic indexing is presented. Reuter 21578 dataset is used for experimentation and the clustering performance of the four clustering algorithms for text document clustering was effectively analyzed. The performance of spherical k-means and PDDP text documents clustering in terms of time, f-measure, and purity were improved. The FCM algorithm gave better results than k-means text document clustering algorithm. A set of non-overlapping partitions using latent semantic indexing is obtained. In future the ontology based methodology and stochastic optimization algorithms for clustering can be studied and implemented.

REFERENCES

[1] K. Cios, W. Pedrycs, & R. Swiniarski, (1998) *Data Mining Methods for Knowledge Discovery*, Boston: Kluwer Academic Publishers.

[2] W.B. Frakes & R. Baeza-Yates, (1992), *Information Retrieval: Data Structures and Algorithms*, Englewood Cliffs, N.J.: Prentice Hall.

[3] Jiawei han & Michelin Kamber , (2010), *Data mining concepts and techniques*, Elsevier.

[4] R. Baeza-Yates & R. Ribeiro-Neto, (1999) ,*Modern Information Retrieval*, Addison Wesley, ACM Press, New York.

[5] Wei Song,Soon Cheol Park. (2009), "Genetic Algorithm for text clustering based on latent semantic indexing", Computers and Mathematics with applications, Vol.57, pp.1901-1907.

[6] Poncelet, Pascal, Maguelonne Teisseire & Florent Masseglia. (2008), "Data Mining Patterns: New Methods and Application", Information Science Reference, Hershey PA, pp. 120-121.

[7] Reuters-21578 Distribution 1.0, http://www.daviddlewis.com/resources/testcollections/reuters21578.

[8] D. Zeimpekis & E. Gallopoulos. (2005), TMG: A MATLAB Toolbox for Generating Term-Document Matrices from Text Collections.

[9] I. S. Dhillon & D. M. Modha. (2001), "Concept Decompositions for Large Sparse Text Data using Clustering", Machine Learning, Vol.42, No.1, pp. 143-175.

[10] D. 1. Boley. (1998), "Principal Direction Divisive Partitioning", Data Mining and Knowledge Discovery, Vol.2, No.4, pp. 325-344.

[11] Jiayin KANG & Wenjun ZHANG. (2011), "Combination of Fuzzy C-means and Harmony Search Algorithms for Clustering of Text Document", Journal of Computational Information Systems , Vol.7, No. 16, pp. 5980-5986.

[12] Sadaaki Miyamoto & H.I , Katsuhiro Honda. (2008), "Algorithm for Fuzzy Clustering. Methods in C-Means Clustering with Applications", ed. S.i.F.a.S Computing, Vol. 229. Osaka, Japan: Scientific Publishing Services Pvt Ltd., Chennai, India.

[13] Hung, MC & Yang D-L. (2001), " An Efficient Fuzzy C-Means Clustering Algorithm", *"Proceedings of 2001 IEEE International Conference on Data Mining"*, San Jose, California.

[14] Shaik sharmila & Bodapati Prajna. (2016), "The Assessment of a Text Document Clustering and Classification using Fuzzy C-Means Clustering Algorithm", International Journal of Computer Science and Technology, Vol.7, No.4, pp. 82-86.

[15] Stuti Karol & Veenu Mangat, (2013), "Evaluation of text document clustering approach based on particle swarm optimization", Central European Journal of Computer Science, Vol**.** 3, No. 2, pp. 69-90.