A Novel Approach for Compact Document Clustering

Rachitha Sony Krotha, Madhavi Dabbiru
Department of Information Technology
GMRIT
Rajam, A.P, India.
Email: rachithagmr@gmail.com

Abstract: **Clustering is the problem of discovering "meaningful" groups in given data. The first and common step in the process of Partitional Clustering is to decide the best value of K, the number of partitions. The clustering solution varies with K. Instead of clustering the data by guessing K value, in this paper we propose to cluster the data based on their similarity to obtain more meaningful clusters. Other characteristics of our clustering approach are (1) It deals with outliers (2) It deals the problem of clustering heterogeneous data (3) It reduces the high dimensionality of the term document matrix (4) It outperforms in accuracy the well- known clustering algorithm K-Means.**

Keywords: **Clustering, Latent Semantic Indexing, Text Preprocessing**, **Term document matrix**.

I. INTRODUCTION

Clustering algorithms are mainly used to retrieve patterns from a large dataset. A database or a data warehouse can contain several dimensions or attributes. Many clustering algorithms are good at handling low dimensional data, involving only two to three dimensions. It is a challenge to cluster data objects in high dimensional space, especially when data is very sparse and highly skewed. To overcome these problems there are some high-dimensionality reduction techniques. Among these Latent Semantic Indexing (LSI) is an efficient method. LSI is an indexing and retrieval method that uses a mathematical technique called Singular Value Decomposition (SVD) to reduce high-dimensionality of vector space model. This work involves preprocessing of text documents, document clustering and high-dimensionality reduction of the generated clusters in order to handle heterogeneous documents. Recent studies indicate that SVD is mostly useful for small homogeneous data collections. For large heterogeneous datasets, the performance of the SVD based text retrieval technique may deteriorate. In this paper, we construct a large heterogeneous text dataset by merging popular text datasets of moderate size. We then use a similarity based clustering technique to cluster the dataset into a few compactly structured datasets. The truncated SVD is performed on the clustered small datasets individually. Our experimental results show that the clustered SVD strategies may enhance the retrieval accuracy on large scale data collections and reduce the SVD computation time and storage space.

This paper is organized as follows: Section two describes related literature to the present research work in the areas of text mining, and section three discusses the methodology for preprocessing text documents in brief. Section four describes the methodology for forming meaningful clusters and discusses the metrics for evaluating the clustering quality. Section five titled "Experimentation and results" shows experimentation on various synthetic and real data sets and presents results in the form of graphs in support of the proposed algorithms. Comparative performance analysis is done for similarity based clustering and the popular K-Means algorithm. Section six gives the conclusions and future scope for extension followed by references.

II. RELATED WORK

Latent semantic indexing (LSI) has emerged as a competitive text retrieval technique (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990). LSI is a variant of the vector space model in which a low-rank approximation to the vector space representation of the database is computed (Berry, Drmac, & Jessup, 1999). Retrieval is performed using the databases of singular values and vectors obtained from the truncated SVD, not on the original term–document matrix. In addition, for large datasets the SVD computation may be too expensive to be carried out on conventional computers. Also, the dense data structure of the truncated SVD matrices poses a huge challenge for both disk and memory spaces of conventional computers (Gao & Zhang, 2003). However, some of these reports do not provide convincing experimental results on large heterogeneous datasets.

III. PREPROCESSING TEXT DOCUMENTS

In our preprocessing approach we collect all the stop words, which are commonly available and are eliminated. This proposal is incorporated into a porter algorithm for stemming that gives effective preprocessing of document. The Porter stemmer is divided into five steps, step1 removes the i-suffixes and step 2 to 4 removes the d-suffixes. The basic idea is to represent each document as a vector of certain keyword word frequencies. In order to do so, the following steps are needed.

1. The documents for which document clustering is to be performed are extracted.
2. Stop words like a, an, the and so on which do not have any content based on ASCII values are eliminated.
3. In the next step (i.e. stemming), the “root” word is found eliminating plurals, tenses, prefixes, and suffixes.
4. The frequency occurrences of each word for every document are counted.
5. Using information-theoretic criteria non-content-bearing “high-frequency” and “low-frequency” words are eliminated. The high frequency words are called keywords.
6. The sorted keywords in each document are now represented using the vector-space model. In this model, each document, *d*, is considered to be a vector, *d*, in the term-space (set of document “words”)
7. Clustering of keywords in each document is done by using hierarchical clustering.

IV. METHODOLOGY

a) Terminology

Vector space model: (or *term vector model*) is an algebraic model for representing text documents as vectors of identifiers, such as, for example, index terms. It is used in information filtering, information retrieval, indexing and relevancy rankings [wiki]

Latent semantic analysis: (LSA) is a technique in natural language processing, in particular in vector semantics, of analyzing relationships between a set of documents and the terms they contain, by producing a set of concepts related to the documents and terms. A matrix containing word counts per document (rows represent unique words and columns represent each document) is constructed from a large piece of text and a mathematical technique called singular value decomposition (SVD) is used to reduce the number of columns while preserving the similarity structure among rows. Words are then compared by taking the cosine of the angle between the two vectors formed by any two rows. Values close to 1 represent very similar words while values close to 0 represent very dissimilar words

Rank lowering: After the construction of the occurrence matrix, LSA finds a low-rank approximation to the term-document matrix. There could be various reasons for these approximations:

- The original term-document matrix is presumed too large for the computing resources.
- The original term-document matrix is presumed *noisy*.
- The original term-document matrix is presumed overly sparse relative to the "true" term-document matrix. (synonymy).

It also mitigates the problem with polysemy, since components of polysemous words that point in the "right" direction are added to the components of words that share a similar meaning.

*Term Document Matrix***:** LSI begins by constructing a term-document matrix, **A**, to identify the occurrences of the *m* unique terms within a collection of *n* documents. In a term-document matrix, each term is represented by a row, and each document is represented by a column, with each matrix cell, a_{ij}, initially representing the number of times the associated term appears in the indicated document, tf_{ij}. This matrix is usually very large and very sparse.

Once a term-document matrix is constructed, local and global weighting functions can be applied to it to condition the data. The weighting functions transform each cell, a_{ij} of *A*, to be the product of a local term weight, l_{ij}, which describes the relative frequency of a term in a document, and a global weight, g_i, which describes the relative frequency of the term within the entire collection of documents.

Cosine similarity is a measure of similarity between two vectors by measuring the cosine of the angle between them. The cosine of 0 is 1, and less than 1 for any other angle; the lowest value of the cosine is -1. The cosine of the angle between two vectors thus determines whether two vectors are pointing in roughly the same direction. The cosine of two vectors:

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos\theta$$

Given two vectors of attributes, *A* and *B*, the cosine similarity, *θ*, is represented using a dot product and magnitude as

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

The resulting similarity ranges from −1 meaning exactly opposite, to 1 meaning exactly the same, with 0 usually indicating independence, and in-between values indicating intermediate similarity or dissimilarity.

Rank-Reduced Singular Value Decomposition: A rank-reduced, Singular Value Decomposition is performed on the matrix to determine patterns in the relationships between the terms and concepts contained in the text. The SVD forms the foundation for LSI. It computes the term and document vector spaces by transforming the single term-frequency matrix, *A*, into three other matrices, a term-concept vector matrix, *T*, a singular values matrix, *S*, and a concept-document vector matrix, *D*, which satisfy the following relations:

$A = TSD^T$
$T^TT = D^T D = I_r \quad TT^T = I_m \quad DD^T = I_n$
$S_{1,1} \geq S_{2,2} \geq ... \geq S_{r,r} > 0 \quad S_{i,j} = 0$ where $i \neq j$

In the formula, *A*, is the supplied m by *n* weighted matrix of term frequencies in a collection of text where *m* is the number of unique terms, and *n* is the number of documents. *T* is a computed *m* by *r* matrix of term vectors where *r* is the rank of *A*, a measure of its unique dimensions ≤ min(m,n). *S* is a computed *r* by *r* diagonal matrix of decreasing singular values, and D is a computed *n* by *r* matrix of document vectors. The LSI modification to a standard SVD is to reduce the rank or truncate the singular value matrix S to size *k* << *r*, typically on the order of a k in the range of 100 to 300 dimensions, effectively reducing the term and document vector matrix sizes to m by k and *n* by k respectively. This reduced set of matrices is often denoted with a modified formula such as:

$$A \approx A_k = T_k \, S_k \, D_k^T$$

Efficient LSI algorithms only compute the first *k* singular values and term and document vectors as opposed to computing a full SVD and then truncating it.

b) Step by step procedure of proposed Document Clustering.

1. Generate Term Document Matrix.
2. Normalize the matrix.
3. Find the document similarity matrix using cosine similarity measure.
4. Find key documents and cluster the documents using closeness feature as described in the clustering algorithm given below.
5. Each cluster consists of a set of document vectors. It can be formed as a matrix(set of document vectors)
6. Find the SVD for each cluster.
7. Thus each cluster is reduced in size.
8. Given a query word it is compared with its appropriate cluster and all the documents which are matched are retrieved.

Clustering Approach

1. A document which is similar to more number (user given threshold) of document vectors is called a popular document.
2. Select a few popular documents from the entire pool of documents.
3. Now assign document vectors to each popular document based on the similarity measure and form clusters.
4. If there exists document vectors which do not comply with any of the popular documents (outliers) then assign them into a special group called Miscellaneous cluster.
5. Repeat steps 3 and 4 until all the document vectors are exhausted.

c) Metrics for evaluating clustering quality.

Clustering is the problem of discovering “meaningful” groups in given data. There exists a huge variety of clustering quality functions.

Silhouette refers to a method of interpretation and validation of clusters of data. The technique provides a succinct graphical representation of how well each object lies within its cluster. It was first described by Peter J. Rousseeuw in 1986.

For each datum i, let a(i)be the average dissimilarity of ‘i’ with all other data within the same cluster. We can interpret a(i)as how well matched ‘i’ is to the cluster it is assigned. Then find the average dissimilarity of ‘i’ with the data of another single cluster. Repeat this for every cluster of which ‘i’ is not a member. Denote the lowest average dissimilarity to ‘i’ of any such cluster by b(i). The cluster with this lowest average dissimilarity is said to be the "neighbouring cluster" of ‘i’ as it is, aside from the cluster ‘i’ is assigned, the cluster in which ‘i’ fits best. We now define:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

This can be written as:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

From the above definition it is clear that

$$-1 \le s(i) \le 1$$

For s(i) to be close to 1 we require a(i)<<b(i). As a(i) is a measure of how dissimilar ‘i’ is to its own cluster, a small value means it is well matched. Furthermore, a large s(i) implies that ‘i’ is badly matched to its neighboring cluster. Thus an s(i) close to one means that the datum is appropriately clustered. If s(i) is close to -1, then by the same logic we see that ‘i’ would be more appropriate if it was clustered in its neighboring cluster. An s(i) near zero means that the datum is on the border of two natural clusters.

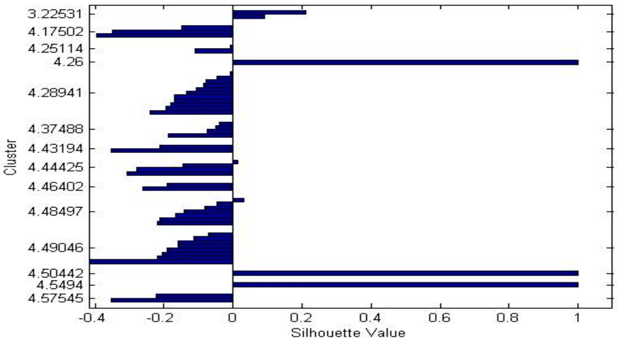


Figure 1: Silhouette plot for Document similarity based clustering.

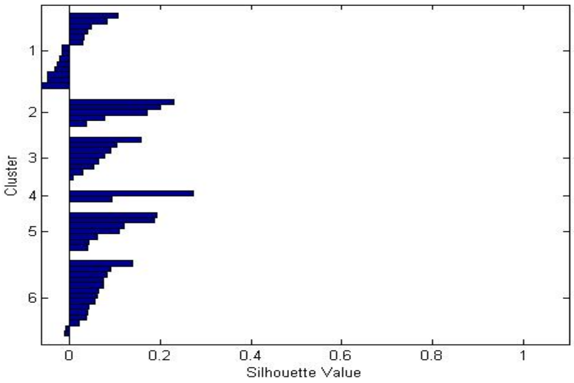


Figure 2: Silhouette plot for K-Means based clustering.

From figures 1 and 2, we observe that the silhouette values of document similarity based clustering (figure 1) are close to 1. This indicates a good clustering quality when compared with the traditional partitional algorithm K-Means (figure 2).

V.EXPERIMENTATION AND RESULTS

We developed three programs for this work. The first one is to preprocess the text documents. The second one is to perform similarity based clustering and the third one is to calculate precision and recall to measure the quality of the clustering solutions. All of our programs are implemented using java on intel P III processor with 512MB RAM.

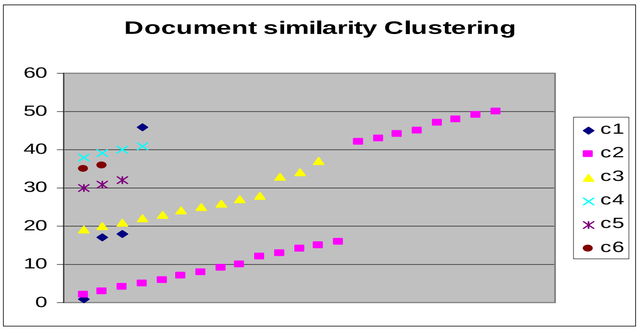


Figure 3: Clustering based on closeness feature.

We used MATLAB 7.0 for computation of SVDs and Silhouette plot. We also used the readymade clustering function K-Means to compare the efficiency and accuracy of our proposed clustering method. We have taken documents of different concepts like medicine, computers, education, and administration. Our test data contains 50 document vectors. Each document consists of 100 to 1000 terms.

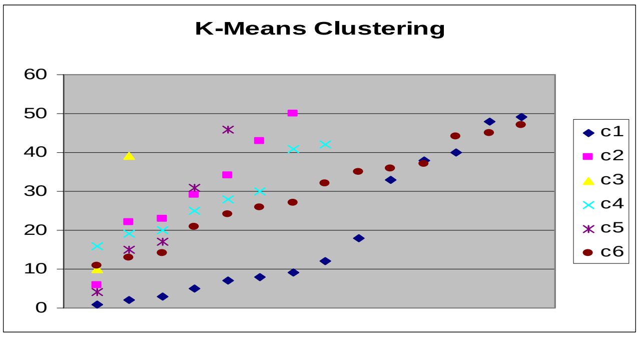


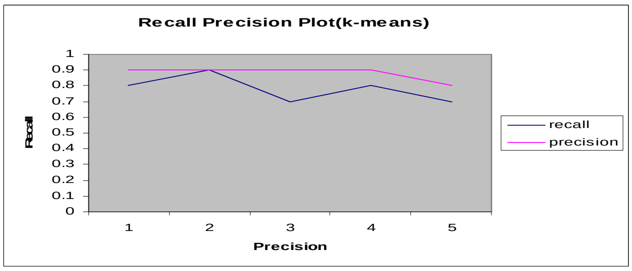
Figure 4: Clustering based on K-Means

Figures 3 and 4 depict the clustering solution of Document similarity based clustering and K-Means clustering solution. Of the two methods, Document similarity based clustering generates more appropriate and meaningful clusters when compared to that of K-Means. The precision and recall values are plotted in figure 5. C1 to C6 are clusters generated. We then calculate the recall and precision of that cluster for each given class. for cluster *j* and class *i*,

$$\text{Recall}(\ i,\ j\ ) = n_{ij}\ /\ n_i$$

$$\text{Precision}(\ i,\ j\ ) = n_{ij}\ /\ n_j$$

where n_{ij} is the number of members of class *i* in cluster *j*, n_j is the number of members of cluster *j* and n_i is the number of members of class *i*.



.
Figure 5: Precision as a function of recall for Clustering based on document similarity.

We can observe from figure 5 that precision deteriorates with the decrease in recall. But the drop is small.

VI. CONCLUSIONS

The similarity based document clustering that we proposed has experimentally demonstrated to produce consistent clustering solutions. The clusters are more relevant and accurate when compared to the traditional K-Means clustering. We also handled heterogeneous data by first clustering the term document matrix and then performed dimensionality reduction using LSI. We also handled outliers in the data.

With recent studies, we came to know that LPI can also be used to reduce the high dimensionality of data. We extend our research by applying locality preserving indexing for dimensionality reduction and also experiment on different sizes of data.

VII. REFERENCES

[1] S.Deerwester, Dumais, Furnas, Landauer, & Harshman, “Indexing by Latent semantic Analysis”, Journal of American Society for information Science and Technology, 1990. 41:p.391-407.

[2] Berry, Drmac, & Jessup, “Matrices,vector spaces and Information Retrieval SIAM Review Vol. 41.No.2 pp 335-362. (1999).

[3] Jing Gao, Jun Zhang, “Sparsification strategies in Latent Semantic Indexing”, Technical report no.368-03, Department of Computer Science, University of Kentucky, Lexington KY, 2003.

[4] G.Nagamani and Madhavi Dabbiru, “Text Mining based on Utility Mining” (2012), 30th-31st March, GIET, Rajahmundry, A.P, India.

[5] K.Rachitha Sony and M.RamaKrishna Murthy, “Reducing High Dimensionality by Latent Semantic Indexing for Document Clustering”(2012), 2nd -3rd March, GMRIT, Rajam, A.P, India.

[6] Deng Cai, Xiaofei He, Jiawei Han, “Tensor Space Model for Document Analysis”, SIGIR’06, August 6–11, 2006, Seattle, Washington, USA.

[7] Jiawei Han and Micheline Kamber., “Data Mining Concepts and Techniques”, Elsevier Pubications.

[8] Rajan chatamvelli “Data Mining Methods”, Narosa publishing house.

[9] Manu Konchady, “Text Mining Application Programming”, Cengage Learning.

[10] Michael W. Berry and Malu Castellanos, “Survey of Text Mining: Clustering, Classification, and Retrieval”, Second Edition September 30, 2007 Springer.

[11] Herbert Schildt, “Java 2 The Complete Reference”, Osborne.

[12] Stephen J.Chapman, “MATLAB Programming for Engineers”, Thomson Learning third edition.

[13] wikipedia.org

[14] www.mathworks.com

AUTHORS PROFILE

K.Rachitha Sony received her M.sc. Degree in Computer Science from Annamalai University, Annamalai and M.Tech(CSE) from JNTUK . Presently she is working as an Assistant Professor in the Department of Information Technogy, GMRIT, Rajam, Srikakulam AndhraPradesh, India. Her areas of interest include Data Mining, Information Retrieval, and Data base Management Systems.

D.Madhavi received her Ph.D. Degree in Computer Science and Engineering from Acharya Nagarjuna University.and M.E. Degree in Computer Engineering with distinction from Andhra University. She is presently working as a Professor in the Department of Information Technology, GMRIT, Rajam, Andhra Pradesh, India. Her areas of interest include Data Mining, Information Retrieval, Data base Management Systems and Software Engineering.