

Decomposition of Term-Document Matrix Representation for Clustering Analysis

Jianxiong Yang

Graduate School of Information, Production and
Systems

Waseda University

Kitakyushu, Japan

leoworldplus@yahoo.co.jp

Junzo Watada

Graduate School of Information, Production and
Systems

Waseda University

Kitakyushu, Japan

junzow@osb.att.ne.jp

Abstract— Latent Semantic Indexing (LSI) is an information retrieval technique using a low-rank singular value decomposition (SVD) of term-document matrix. The aim of this method is to reduce the matrix dimension by finding a pattern in document collection with concurrently referring terms. The methods are implemented to calculate the weight of term-document in vector space model (VSM) for document clustering using fuzzy clustering algorithm. LSI is an attempt to exploit the underlying semantic structure of word usage in documents. During the query-matching phase of LSI, a user's query is first projected into the term-document space, and then compared to all terms and documents represented in the vector space. Using some similarity measure, the nearest (most relevant) terms and documents are identified and returned to the user. The current LSI query-matching method requires computing the similarity measure about the query of every term and document in the vector space. In this paper, the Maximal Tree Algorithm is used within a recent LSI implementation to mitigate the computational time and computational complexity of query matching. The Maximal Tree data structure stores the term and document vectors in such a way that only those terms and documents are most likely qualified as the nearest neighbor to the query will be examined and retrieved. In a word, this novel algorithm is suitable for improving the accuracy of data miners.

Keywords—Fuzzy clustering; data mining; LSI; SVD.

I. INTRODUCTION

Data mining [1] is a technique to obtain some patterns of knowledge from a depository or a data base. This technique enables us to find and describe structural patterns in data collection as a tool for helping to explain the data patterns and make predictions from them. Generally, data mining tasks are divided into two major steps: (1) predictive tasks which aim to forecast the value of a particular attribute based on the values of other attributes and (2) descriptive tasks which aim to derive patterns (correlations, trends, clusters, trajectories, and anomalies) [2]. Clustering is a method to re-organize automatically a large data collection by partitioning a set data, so the objects in the same cluster are more similar to one another than to objects in other clusters. Especially, document clustering aims to re-organize a large text data collection. In

the field of Information Retrieval (IR), document clustering is used to automatically group the document that belongs to the same topic in order to provide user's browsing of retrieval results [3]. Some experimental evidences show that IR application can benefit from the use of document clustering [4]. Document clustering has always been used as a tool to improve the performance of retrieving and navigating large data.

The clustering methods can be classified into two groups: hierarchical method and partitioning method. Partitioning clustering can be classified into hard clustering and overlapping clustering (fuzzy clustering). Any given document has a possibility to contain multiple subjects or categories. This issue aims to use a fuzzy clustering approach, which enables us to include a document in multiple clusters. The method is different from hard clustering method, which a document only belongs to one cluster, not more. The hard clustering assumes well defined boundary among the clusters.

While grouping or clustering of documents, the problem is a very huge number of terms or words are contained in the full text vector space. Many lexical matching at term grade are inaccurate. Sometimes, a word has ambiguously several meaning or some words have synonymously the same meaning, and these cause results in the selection of irrelevant documents. Therefore, this research uses an Information Retrieval approach of Latent Semantic Index (LSI). In this method, the documents are projected onto a small subspace of this vector space and are clustered. So, there is creation of new abstract vector space is capable of capturing contents of important documents in order [3].

The remaining of this paper is written as follows. Section 2 clarifies the research in the past. Section 3 is spent to give the preliminary knowledge on fuzzy clustering. In Section 4, a sufficient and necessary process is shown for the vector space model and LSI. In Section 5, we present an exemplified algorithm of this method. Finally, some concluding remarks are given in Section 6.

II. LITERATURE REVIEW

2.1 LSI Timeline

In 1964, H. Borko and M. Bernick [5] described and tested factor analysis technique first. They report the results of a series of experiments in the techniques of automatic document classifications. Two different classification schedules are compared along with two methods of automatically classifying documents into categories. It is concluded that, while there is no significant difference in the predictive efficiency between Bayesian and Factor Score methods. Automatic document classification is enhanced by the use of a factor-analytically-derived classification schedule. Approximately 55 percent of the documents was automatically and correctly classified. Their experiments provide the information on the effectiveness of techniques for automatic document classification. In the approach to the problem the authors' began by using the computer to help gather data about a representative set of technical documents. In the initial analysis all the content words used in the abstracts of the document were counted and arranged by frequency of occurrence. A limited number of high-frequency words were manually selected to constitute an authority list for use in automatically indexing documents. Every document in the sample was automatically indexed using these terms [5].

In 1988, Furnas *et al.* published a seminal paper on LSI technique. They introduced a new method for automatic indexing and retrieval, implicit higher-order structure in the association of terms with documents is modeled to improve estimates of term-document association, and therefore the detection of relevant documents on the basis of terms found in queries. Singular-value decomposition is used to decompose a large term by document matrix into 50 to 150 orthogonal factors from which the original matrix can be approximated by linear combination; both documents and terms are represented as vectors in a 50 to 150-dimensional space. Queries are represented as pseudo-documents vectors formed from weighted combinations of terms, and documents are ordered by their similarity to the query. Initial tests find this automatic method very promising [6].

Dumais and Nielsen first used LSI to assign articles to reviewers in 1992. The 117 manuscripts submitted for the Hypertext'91 Conference were assigned to members of the review committee, using a variety of automated methods based on information retrieval principles and Latent Semantic Indexing. Fifteen reviewers provided exhaustive ratings for the submitted abstracts, indicating how well each abstract matched their interests. The automated methods do a fairly good job of assigning relevant papers for review, but they are still somewhat poorer than assignments made manually by human experts and substantially poorer than an assignment perfectly matching the reviewers' own ranking of the papers. A new automated assignment method called "n of 2 n" achieves better performance than human experts by sending reviewers more papers than they actually have to review and then allowing them to choose part of their review load themselves [7].

Landauer *et al.* got patent granted for the cross-lingual application of LSI in 1994. This search was stimulated by the earlier examples of thermodynamics which arose from the

attempt to understand the limits of steam engines, and by Shannon's channel capacity theory. These examples suggested that the computer could be characterized in a similar way. The earlier prototypes had derived optimum performance limits in a way which transcended particular technologies and design choices. Scientists and engineers take pride in the ability to do back-of-the-envelope calculations, to quickly reach to the critical aspects without encumbering details. Yet, in this field of ultimate physical limits of information handling, the quick and dirty approaches have turned out to be wrong in a remarkably consistent way. All of the first answers have been misleading [8].

Foltz first use LSI for grading essays in 1995. While the CHI conference focuses on issues of the interactions between humans and computers, it also addresses issues of interactions between humans and more primitive information systems, such as conference proceedings. Over the past two years, the CHI conference committee has tried to improve the usability of the conference proceedings by improving the index. For CHI'92 and '93, the proceedings had no keyword index. This can be due to the fact that generating an index is both a time consuming process and difficult to do well. For CHI'94 and '95, an information retrieval method was used for indexing that permitted both automating the indexing and developing a better index to the CHI proceedings [9].

From 1998 through 2000, an R&D program was conducted to apply a key technology, Latent Semantic Indexing (LSI), to provide superior advanced search capabilities. During this period, several federal agencies were engaged to explore applications of this technology of particular interest to the US Intelligence Community. From 2001 through 2002, a complete software framework was implemented to facilitate the use of this technology in Text Analytics applications.

In 2005, EBSCO Publishing (EBSCO, Content Analyst Company) has announced the release of Executive Daily Brief (EDB), a result of a partnership with the Content Analyst Company. EDB combines Content Analyst's Concept Search with the variety of content offered by EBSCO, to create a customizable business information monitoring and alert service.

EDB is designed to enable users to search for conceptually-related content by entering phrases, paragraphs, or entire articles into a search box. EDB then intuitively finds and retrieves the relevant information. EDB actually "trains" itself as it operates, creating a more intelligent search with each use. Users set up customized folders on topics or projects of their own choosing. EDB monitors more than 10,000 business sources each day and automatically places relevant articles into the appropriate project folders for review, based upon example documents, emails, or articles that the user places in each folder.

EDB's content repository includes business journals, current articles, and real-time business news from a variety of information sources. Along with EDB's own content resources, users may customize EDB's content by exposing other sources of data to the Concept Search, such as pertinent company documents and intranet resources.

2.2 Challenges to LSI

Early challenges to LSI focused on scalability and performance. LSI requires relatively high computational performance and memory in comparison to other information retrieval techniques. However, with the implementation of modern high-speed processors and the availability of inexpensive memory, these considerations have been largely overcome. Real-world applications involving more than 30 million documents that were fully processed through the matrix and SVD computations are not uncommon in some LSI applications.

Another challenge to LSI has been the alleged difficulty in determining the optimal number of dimensions to use for performing the SVD. As a general rule, fewer dimensions allow for broader comparisons of the concepts contained in a collection of text, while a higher number of dimensions enable more specific (or more relevant) comparisons of concepts. The actual number of dimensions that can be used is limited by the number of documents in the collection. Research has demonstrated that around 300 dimensions will usually provide the best results with moderate-sized document collections (hundreds of thousands of documents) and perhaps 400 dimensions for larger document collections (millions of documents). However, recent studies indicate that 50-1000 dimensions are suitable depending on the size and nature of the document collection.

Checking the amount of variance in the data after computing the SVD can be used to determine the optimal number of dimensions to retain. The variance contained in the data can be viewed by plotting the singular values (SV) in a scree plot. Some LSI practitioners select the dimensionality associated with the knee of the curve as the cut-off point for the number of dimensions to retain. Others argue that some quantity of the variance must be retained, and the amount of variance in the data should dictate the proper dimensionality to retain. Seventy percent is often mentioned as the amount of variance in the data that should be used to select the optimal dimensionality for recomputing the SVD.

III. CLUSTERING ANALYSIS

Clustering is useful in several exploratory pattern-analysis, grouping, decision-making, and machine-learning situations, including data mining, document retrieval, image segmentation, and pattern classification. However, in many such problems, there is little prior information (e.g., statistical models) available about the data, and the decision-maker must make as few assumptions about the data as possible. It is under these restrictions that clustering methodology is particularly appropriate for the exploration of interrelationships among the data points to make an assessment (perhaps preliminary) of their structure.

The term “clustering” is used in several research communities to describe methods for grouping of unlabeled data. These communities have different terminologies and assumptions for the components of the clustering process and the contexts in which clustering are used. Thus, we face a dilemma regarding the scope of this survey. The production of a truly comprehensive survey would be a monumental task given the sheer mass of literature in this area. The accessibility

of the survey might also be questionable given the need to reconcile very different vocabularies and assumptions regarding clustering in the various communities.

Fuzzy clustering methods partition a set of data into a number of overlapping clusters based on the distance in a metric space between the data points and the cluster prototypes. Various clustering algorithms can be used depending on the assumed structure of the identification data and the model type one wants to obtain [11]. A clustering method that has proven suitable for the identification of TS fuzzy models is the GK fuzzy clustering algorithm [12]. Unlike the popular fuzzy - means algorithm [13], the GK algorithm employs an adaptive distance norm in order to detect clusters of different geometric shapes in the data set.

Each cluster in the product space of the input/output data represents a rule in the rule base. The goal is to establish the fuzzy antecedents A_i . These are defined by the fuzzy clusters found in the data. If desired, univariate membership functions μ_{ij} can be obtained by projections onto the various input variables x_j spanning the cluster space. The details are described by [11] and [14].

IV. VECTOR SPACE MODEL AND LSI

4.1 Latent Semantic Indexing

Latent Semantic Indexing (LSI) is an indexing and retrieval method that 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 texts. LSI is based on the principle that words tend to have similar meanings in the same context. It requires relatively high computational performance and large memory in comparison to other information retrieval techniques [15]. A key feature of LSI is its ability to extract the conceptual content of texts by establishing associations among those terms which occur in similar contexts.

Latent Semantic Indexing can be able to correlate semantically terms in a collection of texts. The method was first applied to texts at Bell Laboratories in the late 1980s, also called Latent Semantic Analysis (LSA), which can uncover the underlying latent semantic structure in the usage of words in a body of texts and extract the meaning of the text in response to user queries, commonly referred to as concept searches. Queries, or concept searches, depend on a set of documents which have undergone LSI will return results that are similar in meaning to the search criteria even if the results do not share specific words with the search criteria [16].

4.2 Singular Value Decomposition

In linear algebra, the singular value decomposition (SVD) is an important factorization of a rectangular real or complex matrix, with many applications in signal processing and statistics. Suppose $\mathbf{A}$ is an $m \times n$ matrix whose entries come from a space, which is either the space of real numbers or the space of complex numbers. Then there exists a factorization of the form

$$\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^T \quad (1)$$

where U is an $m \times m$ unitary matrix, the matrix A is an $m \times n$ diagonal matrix with nonnegative real numbers on the diagonal, and V^T is an $n \times n$ unitary matrix, denotes the conjugate transpose of V . Such a factorization is called the singular value decomposition of A . The diagonal entries Σ_{ij} of Σ are known as the singular values of A . It's shown in Figure 1:

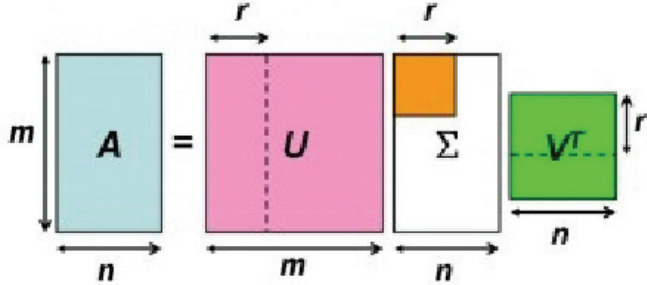


Figure 1. The singular value decomposition

For obtaining simplified vector space, we just need pick up the non-zero singular values of Σ . The rank- r is the amount of non-zero singular values of Σ .

A common usage is to place the singular values in descending order. In this case, the diagonal matrix Σ is uniquely determined by A .

So we obtain the followed new equation(2) and Figure 2:

$$A = U_r \Sigma_r V_r^T \quad (2)$$

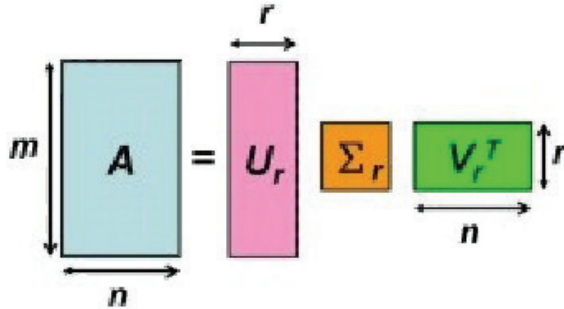


Figure 2. The simplified SVD

4.3 An Example of SVD

This example has terms 1, 2, 3, 4 and 5 as follows:
 Term 1 appear 1, 0, 0, 0 times in document 1, 2, 3, 4;
 Term 2 appear 0, 0, 0, 4 times in document 1, 2, 3, 4;
 Term 3 appear 0, 3, 0, 0 times in document 1, 2, 3, 4;
 Term 4 appear 0, 0, 0, 0 times in document 1, 2, 3, 4;
 Term 5 appear 2, 0, 0, 0 times in document 1, 2, 3, 4.

Factor matrix A , that is, term-document matrix A is shown in matrix 3:

$$A = \begin{pmatrix} t_1 & t_2 & t_3 & t_4 & t_5 \\ 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \end{pmatrix} \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{matrix} \quad (3)$$

Step 1:

Calculate the interactive matrix (4).

$$A \cdot A^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 16 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 4 \end{pmatrix} \quad (4)$$

Step 2:

We calculate the eigenvalues and singular vectors of $A \cdot A^T$. The non-zero singular eigenvectors are shown by descending as:

$$\lambda_1=16, \lambda_2=9, \lambda_3=5.$$

The non-zero singular values are obtained in the following:

$\delta_1=4, \delta_2=3, \delta_3=\sqrt{5}$, the rank- r of non-zero singular matrix Σ is 3.

So, the eigenvectors are in the following:

$$v_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, v_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, v_3 = \begin{pmatrix} \sqrt{0.2} \\ 0 \\ 0 \\ 0 \\ \sqrt{0.8} \end{pmatrix}.$$

So the singular value matrix Σ and the term decomposition matrix V are shown as:

$$\Sigma_r = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & \sqrt{5} \end{pmatrix};$$

$$V_r^T = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \sqrt{0.2} & 0 & 0 & 0 & \sqrt{0.8} \end{pmatrix}.$$

Step 3:

Calculating the document decomposition matrix U .

$$u_1 = \frac{1}{\delta_1} A v_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

$$u_2 = \frac{1}{\delta_2} A v_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},$$

$$u_3 = \frac{1}{\delta_3} A v_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$U_r = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

So if we want to analyze the attribute of document, we just need calculate the document decomposition matrix U_r .

V. ACCIDENT CASE CLUSTERING BASED ON LSI

Latent semantic analysis employs information retrieval to explore latent semantics and attributes of accident cases. So, we use latent semantic method to analyze a large number of accidents data. This analysis has the following steps: build the primary matrix, execute singular value decomposition, pursue similarity analysis of accident cases and obtain the results clustering.

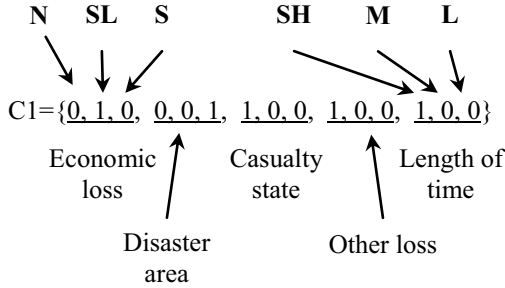


Figure 3. The description of accident attributes

TABLE I. SAMPLE DATA

No.	Attribute				
	Economic loss	Disaster area	Casualty state	Other loss	Length of time
C1	SL	S	N	N	SH
C2	N	N	S	S	SH
C3	SL	S	S	N	SH
C4	N	SL	S	S	SH
C5	S	S	S	N	SH
C6	S	SL	SL	SL	L
C7	SL	S	SL	N	M
C8	N	S	S	N	L

TABLE II. THE PRIMARY MATRIX

	C1	C2	C3	C4	C5	C6	C7	C8
Economic loss : N	0	1	0	1	0	0	0	1
Economic loss : SL	1	0	1	0	0	0	1	0
Economic loss : S	0	0	0	0	1	1	0	0
Disaster area : N	0	1	0	0	0	0	0	0
Disaster area : SL	0	0	0	1	0	1	0	0
Disaster area : S	1	0	1	0	1	0	1	1
Casualty state : N	1	0	0	0	0	0	0	0
Casualty state : SL	0	0	0	0	0	1	1	0
Casualty state : S	0	1	1	1	1	0	0	1
other loss : N	1	0	1	0	1	0	1	1
other loss : SL	0	0	0	0	0	1	0	0
other loss : S	0	1	0	1	0	0	0	0
Length of time : SH	1	1	1	1	1	0	0	0
Length of time : M	0	0	0	0	0	0	1	0
Length of time : L	0	0	0	0	0	1	0	1

5.1 Establishing Primary Matrix

In our accident database, all the data include 5 attributes: (1) economic loss, (2) disaster area, (3) casualty state, (4) other loss and (5) length of time. Every attribute has 3 serious grades which economic loss, disaster area, casualty state and other loss are **N**(nothing), **SL**(slight) and **S**(serious), and length of time is **SH**(short), **M**(middle) and **L**(long). For example, the accident case 1 has slight economic loss, serious disaster area, no casualty state, no other loss and short length of time. If the any attribute of an accident case has any grade, then this grade is "1", or else it is "0". It may denote the expression shown in Figure 3.

Table I recorded eight accident cases from the National Institute of Technology Evaluation (NITE). So we can transform it to a primary matrix as shown in Table II.

5.2 Singular Value Decomposition

In Table II, we have eight cases with five attributes. These create a 15x8 matrix, so we use Matlab to decompose singular value and obtain decomposed matrix U , Σ and V^T . We chose the non-zero singular values of Σ to build U_r , Σ_r and V_r^T which are shown in matrices 5, 6 and 7, respectively.

$$U_r = \begin{pmatrix} -0.39071 & 0.32158 \\ -0.29845 & -0.56435 \\ -0.46969 & 0.15712 \\ -0.30411 & -0.56933 \\ -0.43633 & 0.037314 \\ -0.10217 & -0.034567 \\ -0.30819 & 0.47413 \\ -0.38615 & -0.043407 \end{pmatrix} \quad (5)$$

$$\Sigma_r = \begin{pmatrix} 4.3642 & 0 \\ 0 & 2.8161 \end{pmatrix} \quad (6)$$

$$V_r = \begin{pmatrix} -0.22655 & -0.41799 \\ -0.26777 & 0.33835 \\ -0.12346 & 0.000975 \\ -0.068385 & -0.2004 \\ -0.093093 & -0.21445 \\ -0.4563 & 0.33619 \\ -0.089526 & 0.11419 \\ -0.094028 & 0.15609 \\ -0.43422 & -0.34894 \\ -0.4563 & 0.33619 \\ -0.02341 & -0.012275 \\ -0.13807 & -0.40257 \\ -0.43527 & -0.21933 \\ -0.070618 & 0.16837 \\ -0.11189 & -0.027689 \end{pmatrix} \quad (7)$$

5.3 The Clustering Of Accident Cases

At first, we calculate the cosine of any two accident cases in 2-Dimensional semantic space. If the cosine of these two cases is "1", the angle of two case vectors is 0-degree, that is, the two cases are the same else cosine is "0" these two cases are not the same. All of the values are shown in Table III.

From the data of Table III, we use the similarity measure data shown in Table III, to build the maximal tree. First, we pick all the vertices for which relation values are not less than 0.9. Thus, the vertices {1, 3}, {1, 7}, {2, 4}, {3, 5}, {3, 8}, {5, 6}, {5, 8}, and {6, 8} are selected output. We connect these

vertices by their similarity value to build the initial cluster tree, as shown in Figure 4.

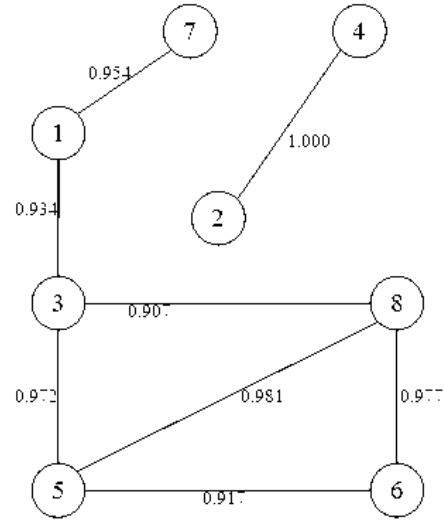


Figure 4. The initial cluster tree

The maximal tree cannot have a circuit because any two vertices have only one maximal relation. Thus, we must remove the edge of minimum value in a circuit. The edge {3, 8}, and {5, 6} should be removed. The next cluster tree is shown in Figure 5.

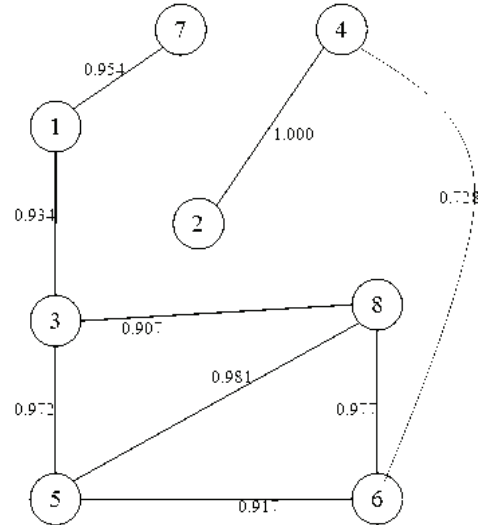


Figure 5. The clustering tree

TABLE III. FUZZY SIMILARITY DATA

	1	2	3	4	5	6	7	8
1	1							
2	-0.20082	1						
3	0.93383	0.16291	1					
4	-0.19676	0.99999	0.16699	1				
5	0.82342	0.39052	0.97192	0.39433	1			
6	0.52771	0.72615	0.79565	0.72899	0.91652	1		
7	0.95362	-0.4864	0.78282	-0.48278	0.61441	0.24753	1	
8	0.69629	0.56331	0.90698	0.56673	0.98062	0.97712	0.44792	1

However vertices 2 and 4 are not included in group {1, 3, 5, 6, 7, 8}. We thus need to connect add the two groups of vertices, {2,4} and {1,3,5,6,7,8} and connect vertices with maximum similarity value between the two groups. The similarity values are 0.782 for {4, 6}. So, all the vertices are connected. The cluster tree is shown in Figure 6.

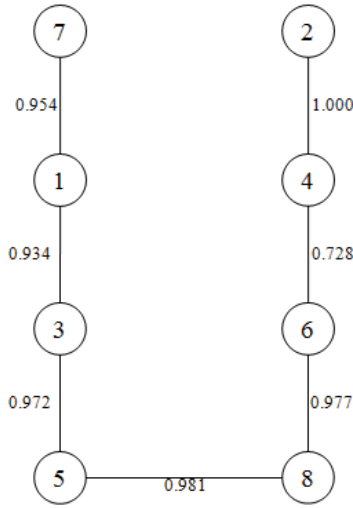


Figure 6. The Maximal Tree

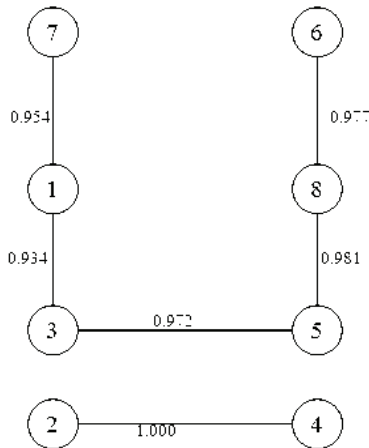


Figure 7. The clustering by $\lambda=0.900$

At last, we get a Maximal Tree, which each side has a weight " r_{ij} ". By the Threshold- λ , we remove all the sides which have weight $r_{ij} < \lambda$. In the remaining vertexes, any connected vertexes are the same cluster [10].

If we select a high similarity value of $\lambda=0.900$, based on the clustering tree, we delete all the sides whose similarity value of λ blew 0.900. So the cases 1, 3, 5, 6, 7 and 8 are similar in the same group, and the Cases 2 and 4 are similar in the same group. It is shown in Figure 7. In addition, if we wish to extract some accident cases, we just need to embed the equivalent example data of these cases in the data set. If any ones can be clustered to a group with the equivalent example data according to the threshold λ , it must be a useful data element which we want [10].

VI. APPLICATION OF LSI

LSI is used to realize automated document categorization. Several experiments have demonstrated a number of successful correlations between the way LSI and humans' process and effectively categorized text [17]. Document categorization is the assignment of documents to one or more predefined categories based on their similarity to the conceptual content of the categories [18]. LSI uses template documents to establish the conceptual basis for each category. During the process of categorization, the concepts contained in the documents categorized are compared to the concepts contained in the template items, and a category or categories are assigned to the documents based on the similarities between the concepts they contain and the concepts that are contained in the template documents.

LSI can realize effectively dynamic clustering based on the conceptual content of documents can also be accomplished using LSI. Clustering is a way to group documents based on their conceptual similarity to each other without using template documents to establish the conceptual basis for each cluster. This is useful when dealing with an unknown collection of unstructured text.

LSI is not restricted only to working with words. It can also process arbitrary character strings. Any object if it can be expressed in text can be represented in an LSI vector space. For example, texts with MEDLINE abstracts have shown that LSI is able to effectively classify genes based on conceptual modeling of the biological information contained in the titles and abstracts of the MEDLINE citations [19].

LSI has proven to be a useful solution to a number of conceptual matching problems [20][21]. The technique has been shown to capture key relationship information, including causal, goal-oriented, and taxonomic information [22].

VII. CONCLUSIONS

The latent semantic indexing (LSI) methodology for information retrieval has applied the singular value decomposition to identify an eigenfunction for a large matrix, whose cells represent the occurrence of terms(conditions) within documents. This methodology was used to rank text documents and data clustering, based on their relevance to a topic. In this paper, we proposed the method that integrated the Latent Semantic Indexing (LSI) concept to our document clustering. This involves the use of Singular Value Decomposition (SVD) which creates a new abstract and uses a way of finding simplified document collection in matrix representation, so that it could identify the terms and documents which are similar. The considered synthetic and real-world examples demonstrated the improved mining properties due to the befitting cluster and the algorithms capability of determining a suitable similarity degree of clusters in the data.

REFERENCES

- [1] Y. Lei, V. Uren, and E. Motta, "SemSearch: A Search Engine for the Semantic Web," Proc. 15th Int'l Conf. Managing Knowledge in a World of Networks (EKAW '06), pp. 238-245, 2006.
- [2] PN Tan, M.S., V Kumar, Introduction to Data Mining. Pearson International ed. 2006: Pearson Education, Inc.
- [3] M.A. Hearst, a.J.O.P. "Reexamining the cluster hypothesis". 1996: In Proceeding of SIGIR '96.
- [4] Jardine, N.a.v.R., C.J., The Use of Hierarchical Clustering in Information Retrieval. Information Storage and Retrieval. Vol. 7. 1971.
- [5] H. Borko, S. Monica, "Automatic Document Classification Part II . Additional Experiments", Journal of the ACM (JACM) JACM Homepage archive, Volume 11 Issue 2, pp. 138-151. April 1964.
- [6] G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter, and K. E. Lochbaum, "Information retrieval using a singular value decomposition model of latent semantic structure", In SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval (1988), pp. 465-480.
- [7] S.T. Dumais and J. Nielsen, "Automating the Assignment of Submitted Manuscripts to Reviewers", Proceedings of the ACM SIGIR'92 15th International Conference on Research and Development in Information Retrieval (Copenhagen, Denmark, 21-24 June 1992), pp.233-244.
- [8] R.Landauer, "Zig-zag path to understanding [physical limits of information handling]", Physics and Computation, 1994. PhysComp '94, Proceedings., Workshop on , Issue Date: 17-20, Nov 1994, pp. 54-59
- [9] Peter W. Foltz, Improving human-proceedings interaction: indexing the CHI index, CHI '95 Conference companion on Human factors in computing systems, pp.101-102.
- [10] J.X Yang, J. Watada; "Efficient data mining method based on Fuzzy Clustering", Proceedings, International Symposium on Management Engineering, Aug. 2010, Japan, Kitakyusyu, pp.242-249.
- [11] R. Kruse F, R. Höppner, F. Klawonn, and T. Runkler, Fuzzy Cluster Analysis. New York: Wiley, 1999.
- [12] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in Proc. IEEE CDC, San Diego, CA, 1979, pp. 761-766.
- [13] J. C. Bezdek, Pattern Recognition With Fuzzy Objective Functions. New York: Plenum, 1981.
- [14] R. Babu'ska, Fuzzy Modeling for Control. Boston, MA: Kluwer , 1998
- [15] Karypis, G., Han, E., "Fast Supervised Dimensionality Reduction Algorithm with Applications to Document Categorization and Retrieval", Proceedings of CIKM-00, 9th ACM Conference on Information and Knowledge Management, 2000, pp. 12 - 19.
- [16] JX Yang, J Watada; "Wise search engine based on LSI", 2010 Autonomous Agents and Multiagent Systems (AAMAS2010), May. 2010, Canada, Toronto, Workshops-ADMI, pp.127-137.
- [17] Landauer, T., et al., "Learning Human-like Knowledge by Singular Value Decomposition: A Progress Report", M. I. Jordan, M. J. Kearns & S. A. Solla (Eds.), Advances in Neural Information Processing Systems 10, Cambridge: MIT Press, 1998, pp. 45-51.
- [18] Bradford, R., An Empirical Study of Required Dimensionality for Large-scale Latent Semantic Indexing Applications, Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, California, USA, 2008, pp. 153-162.
- [19] Homayouni, Ramin, Heinrich, Kevin, Wei, Lai, Berry, Michael W., Gene Clustering by Latent Semantic Indexing of MEDLINE Abstracts, August 2004, pp. 104-115.
- [20] Ding, C., A Similarity-based Probability Model for Latent Semantic Indexing, Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 1999, pp. 59-65.
- [21] Bartell, B., Cottrell, G., and Belew, R., Latent Semantic Indexing is an Optimal Special Case of Multidimensional Scaling, Proceedings, ACM SIGIR Conference on Research and Development in Information Retrieval, 1992, pp. 161-167.
- [22] Graesser, A., and Karnavat, A., Latent Semantic Analysis Captures Causal, Goal-oriented, and Taxonomic Structures, Proceedings of CogSci 2000, pp. 184-189.