ISSN:2320-0790

An International Journal of Advanced Computer Technology

# Cluster Tree Based Hybrid Document Similarity Measure

M. Varshana Devi

PG Student, Department of Computer Science and Engineering,
V.S.B. Engineering College, Karur
varshuccet@gmail.com

**Abstract:** Cluster tree based hybrid similarity measure is established to measure the hybrid similarity. In cluster tree, the hybrid similarity measure can be calculated for the random data even it may not be the co-occurred and generate different views. Different views of tree can be combined and choose the one which is significant in cost. A method is proposed to combine the multiple views. Multiple views are represented by different distance measures into a single cluster. Comparing the cluster tree based hybrid similarity with the traditional statistical methods it gives the better feasibility for intelligent based search. It helps in improving the dimensionality reduction and semantic analysis.

**Keywords: Dimensionality reduction, semantic analysis, cluster tree, hybrid similarity, term association**

## I. INTRODUCTION

The evolution of human languages has been expedited by the use of the Internet. We see a growing demand for semantic representation that includes the term associations and spatial distributions. Another demand is to find low-dimensional semantic expressions of documents, while preserving the essential statistical relationships between terms and documents. Some usages of low-dimensional representation are extremely useful for facilitating the processing of large document corpora and the handling of various data mining tasks, such as classification, retrieval, plagiarism, etc. However, the main challenge for document analysis knows how to locate the low-dimensional space with the fusion of local information, which conveys term associations and spatial distributions, in a unified framework. Here, we introduce a new model for in-depth document analysis, named multidimensional latent semantic analysis (MDLSA). It starts by partitioning each document into paragraphs and establishing a term affinity matrix. Each component in the matrix reflects the statistics of term cooccurrence in a paragraph. It is worth noting that the document segmentation can be implemented in a finer manner, for example, partitioning into sentences. Thus, it allows us to perform an in-depth analysis in a more flexible way. We then conduct a 2-D principal component analysis (2DPCA) with respect to the term affinity matrix. This analysis relies on finding the leading eigenvectors of the sample covariance matrix to characterize a lower dimensional semantic space. According to our empirical study, we find that using only a 1-D projection to represent each document is sufficient to achieve marked results. Moreover, a hybrid document similarity measure is designed to further improve the performance of this framework. In comparison with the traditional "Bag of Words" (BoW) models such as the latent semantic indexing (LSI) and the principal component analysis (PCA), MDLSA aims to mine the in-depth document semantics, which enables us to not only capture the global semantics at the whole document level, but also to deliver the semantic information from local data-view regarding the term associations at the paragraph level. The problems in these methods are overcome by the cluster tree. The results corroborate that the proposed technique is accurate and computationally efficient for performing various document applications.

## II. EXTRACTING GLOBAL FEATURES

In this section, we introduce the common procedures of document feature extraction, such as preprocessing, vocabulary construction, forming a weighted term vector, which is regarded as a global representation of a document, and dimensionality reduction.

### 2.1 Vocabulary Construction

First, we introduce the common document feature extraction procedures. The preprocessing works by first separating the main text contents from documents, for example, HTML formatted documents. We then extract words from all the documents in a dataset and apply stemming to each

word. Stems are often used as basic features instead of original words. Thus, "program," "programs," and "programming" are all considered as the same word. We remove the stop words (a set of common words like "a," "the," "are," etc.) and store the stemmed words together with the information of the tf, $f^t_u$(the frequency of the *u*th word in all documents), and the document frequency, $f^d_u$ (the number of documents the *u*-th word appears). Forming a histogram vector for each document requires the construction of a word vocabulary each histogram vector can refer to. Based on the stored *tf* and document frequency, we use the well-known *tf-idf* term-weighting measure to calculate the weight of each word

$$W_u = f^t_u \cdot idf$$

where *idf* denotes the inverse-document-frequency that is given by idf = log2 ($n/f^d_u$ ), and *n* is the total number of documents in a dataset. It is noted that this term-weighting measure can be replaced by other feature selection criteria. The words are then sorted in descending order according to their weights. The first *m* words are selected to construct the vocabulary *M*. According to the empirical study using all the words in the dataset to construct the vocabulary is not necessarily expected to deliver the improvement of performance because some words may be noisy features for some topics. We have conducted detailed experiments to evaluate the performance in terms of different options of the vocabulary size, i.e., the value of *m*.

### 2.2 Dimensionality Reduction

A document set can be represented by $X = [x_1, x_2, \ldots, x_n] \in R^{m \times n}$, which is a rectangular matrix of terms and documents. The desire of latent semantic analysis is to produce a set *Y* , which is an accurate representation of *X*, but resides in a lower dimensional space. *Y* is of dimension *d*, with $d << m$, and it is produced by the form

$$Y = V^T_g X$$

where *Vg* is an $m \times d$ linear transformation matrix. Thus, it is straightforward to replace each document *xi* by its projection $y_i = V^T_g x_i$ such that we can make between or within comparisons facile in the lower dimensional latent semantic space. There are a number of ways to accomplish this projection. The transformation matrix *Vg* can be obtained by traditional techniques such as the PCA, the LSI, or other dimensionality reduction approaches. In this study, we use the classical PCA to determine the matrix *Vg*. The PCA is a well-known technique in the category of dimensionality reduction. In the PCA, the determination of *Vg* is given by maximizing the variance of the projected vectors, which is in the format of

$$\max \sum_{i=1}^{n} ||yi - \frac{1}{n} \sum_{i=1}^{n} yi||^2_2.$$

It has been shown that the matrix *Vg* is the set of eigenvectors of the sample covariance matrix associated with the *d* largest eigenvalues. Keep this in mind, as we will use this set of global representations $\{y_1, y_2, \ldots, y_n\}$ to formulate a hybrid similarity of two documents.

## III. WORD AFFINITY GRAPH

This section introduces a scheme to produce an in-depth document representation. First, we segment each document into paragraphs. Second, we build a word affinity graph, which describes the local information of each document.

### 3.1 Document Segmentation

As we mentioned before, the major drawback of the traditional modeling methods such as the PCA and the LSI is that they lack the description of term associations and spatial distribution information over the reduced space. In this study, we propose a new document representation that contains this description. First, each document is segmented into paragraphs. Since we only considered the HTML documents in this paper, a Java platform was developed to implement the segmentation. For the HTML format document, we can use the HTML tags to identify paragraphs easily. Before document segmentation, we first filter out the formatted text that appears within the HTML tags. The text is not accounted for in word counts or document features. The overall document partitioning process can be summarized as follows 1) Partition a document into blocks using the HTML tags: "<p>," "<br\>," "<li>," "</td>," etc. 2) Merge the subsequent blocks to form a new paragraph until the total number of words of the merged blocks exceeds a paragraph threshold (set at 50). 3) The new block is merged with the previous paragraph if the total number of words in a paragraph exceeds the minimum threshold (set at 30). For the HTML documents, it is noted that there is no rule for minimum/maximum number of words for paragraphs. Setting a threshold for word counts, however, still enables us to control the number of paragraphs flexibly in each document and remove the blocks, which contain only a few words (e.g., titles), by being attached to the real paragraph blocks. It is worth pointing out that we are able to further partition each paragraph into sentences by marking periods (the tag "/.") to form a finer structure such that more semantics can be included.

### 3.2 Word Affinity Graph

Building a word affinity graph for each document is to represent the frequency of term cooccurrence in a paragraph. Consider a graph

denoted by a matrix $Gi \in R^{m\times m}$, in which each element $g_{i,u,v}(u, v = 1, 2, \ldots, m)$ is defined by

$$g_{i,u,v}=\begin{cases} Fu,v \cdot log2(\frac{n}{DFu,v})/||Gi||, u \neq v \\ ftu \cdot \frac{\log\left(\frac{n}{fud}\right)}{||Gi||}, u = v \end{cases}$$

where $||\cdot||2$ is the Frobenius norm, $Fu,v$ is the frequency of the cooccurrence in a paragraph associated with the terms $u$ and $v$ in the $i$th document, $DFu,v$ is the document frequency that the terms $u$ and $v$ co appear in a document, and notations of $f^t_u$ and $f_{u.}^{\ d}$ cooccurrence in paragraphs, i.e., let $gi,u,v = 0$ (for $u \_= v$), the affinity graph $Gi$ becomes a diagonal matrix with the elements corresponding to the global feature vector $xi$ shown in (2) (the NORM weighting). By definition, the graph $Gi$ is a symmetric matrix. This graph contains the local semantic information of a document in a way that we can design an efficient semantic representation including term interconnections and distributions in a unified framework.

## IV. MULTIDIMENSIONAL LATENT SEMANTIC ANALYSIS

This section presents a new model,MDLSA, which considers word affinity graphs and maps them onto a low-dimensional latent semantic space. First, we introduce the objective of the MDLSA model. Second, we learn a semantic subspace by using the 2DPCA algorithm. Third, we further process and select the semantic projections. We summarize the MDLSA algorithm in the end.

### 4.1 Semantic Projection

Despite the capability of delivering more semantics, a word affinity graph is usually of large size and sparseness. It is computationally demanding if we simply rely on these graphs to make between or within comparisons. Besides, assembling the similarity between two matrices is another demanding issue. On the other hand, without further processing, these graph representations contain a large quantity of noises, which spread out the original term distributional space. As a result, these noises cause degradation of performance. Therefore, it is important to design an efficient dimensionality reduction technique, which is able to compress the graph in a principled manner and form an accurate representation in a lower dimensional space. The proposed MDLSA model is just this. Given a word affinity graph $G$ of size $m \times m$ , the goal of MDLSA is to produce a projection $\tilde{} Z$ of size $d \times d$ ($d << m$) resided in a lower dimensional semantic space. We then use a matrix $Z$ of size $d \times k$ ($k \leq d$), which is constructed by a smaller number of columns of $\tilde{} Z$. In linear algebra, the projection $\tilde{}Z$ can be obtained by

$$\tilde{}Z=V^TGV$$

where $V$ is an $m \times d$ linear transformation matrix. The problem comes to finding an optimal transformation $V$ for this dimensionality reduction.

### 4.2 Learning a Semantic Subspace

To acquire the optimal transformation matrix $V$ , we use the 2DPCA method, which has been successfully implemented in a face recognition system. For completeness, the process of calculating the matrix $V$ is summarized here, and the details can be found in the article reported by Yang *et al*. [25]. Let $\{G1,G2, \ldots, Gn\}$ be a set of training documents. By representing the word affinity graph $Gi$ associated with the $i$th document, the graph covariance (or scatter) matrix $C$ can be written by

$$C=\frac{1}{n}\sum_{i=1}^{n}(Gi - \bar{G})T(Gi-\bar{G})$$

where $\bar{G}$ denotes the average graph of all the training samples.Similar to PCA, 2DPCA introduces this total scatter of the projected samples to measure the discriminatory power of a transformation matrix $V$ . In fact, the total scatter of the samples in a training set can be characterized by maximizing the criterion.

$$J(v)=v^TCv$$

where $v$ is a unitary column vector, which is called the optimal mapping axis by maximizing the above quantity. In general, it is not sufficient to have only one optimal mapping axis. It is required to find a set of mapping axis, $v_1, v_2, \ldots, v_d$ , subject to the orthogonal constraints and maximizing the criterion $J(V)$ by the form

$$\{v_1, v_2, \ldots, v_d\}= \arg\max J(v)$$
$$\text{Subject to } v_j^Tv_l=0(j\neq l,j,l=1,2,\ldots,d).$$

According to linear algebra, the optimal mapping axes, $v_1, v_2, \ldots, v_d$ , are the orthogonal eigenvectors of $C$ associated with the first largest $d$ eigenvalues. If we denote these mapping axes by $V = [v_1, v_2, \ldots, v_d]$, the projection $\tilde{} Z$ of a word affinity graph $G$ will be acquired easily by the product of the resulting matrices Here, note that we take advantage of the symmetry of the affinity graph $G$.

### 4.3 Selection of the Semantic Projections

Actually, we can use another matrix $Z$ of size $d \times k$ ($k \leq d$), which is a sub matrix of $\tilde{} Z$, to represent the original graph $G$ for optimal approximate fit by discovering lower dimensional space. In practice, only using the first column of $\tilde{} Z$ is sufficient to achieve remarkable results. Thus, the matrix $Z$ is of size $d \times k$ (here, $k = 1$) and turns

out to be a column vector like $y_i$ produced by the traditional PCA corresponding to the global feature $x_i$ . We also conducted an empirical study on the selections of value of $k$. To avoid confusion, in the following context, let $z_i$ be the first column of ˜ $Z_i$ , which denotes the projection matrix of the $i$th affinity graph $Gi$ . Alternatively, the local information from the $i$th training document can be represented by the column vector $z_i$ . This is a very promising property of MDLSA by delivering three important advantages. First, in comparison with 2DPCA it does not need an assembled metric to conduct direct matrix comparison such that MDLSA is easier to make between comparisons. Second, much less time is required to compare two documents because MDLSA does not need the many-to-many matching compared with the MLM method. Third, MDLSA contains local semantic information of documents compared with the PCA and the LSI.

## V. HYBRID DOCUMENT SIMILARITY

Many document applications rely on the calculation of similarity between two documents. In order to further improve the performance of our framework, we develop a hybrid similarity measure to synthesize the information from a global data-view and local data-view. In this study, we have extracted two sets of features from each document: a feature vector $x_i$ containing global information (i.e., *tf*) and an affinity graph $G_i$ delivering local information (i.e., term associations). We then use dimensionality reduction techniques to map these features onto the latent semantic space, which is of lower dimension. Intuitively, combining these two information sources may bring performance gain. Therefore, we design a hybrid similarity associated with both the global and local information. Given two documents $p$ and $q$, let $y_p$ be the latent representation of document $p$ associated with the global feature $x_p$ , and $z_p$ the latent representation of document $p$ produced from the local source $G_p$ . Likewise, let $yq$ be the latent representation of document $q$ associated with the global feature $x_q$ , and $z_q$ the latent representation of document $q$ produced from the local source $G_q$ . We work by a combined similarity measure in the form, which involves the *cosine* distance criterion

$$S(p, q) = \mu Sg\,(p, q) + (1 - \mu) Sl\,(p, q)$$
$$Sg\,(p, q) = y_p \cdot y_q / \|z_p\|2\|z_q\|2,$$
$$Sl(p,q) = z_p \cdot z_q / \|z_p\|2\|z_q\|2.$$

where $S_g$ $(p, q)$ represents the global similarity, $S_l$ $(p, q)$ denotes the local similarity, and $\mu(0 \leq \mu \leq 1)$ is a weight parameter used to balance the importance of the global and local similarity. Thus, the system provides users flexibility to select the value of $\mu$ to balance this hybrid measure according to their expectations. In this study, we also include the effect study of the parameter $\mu$ in experiments. Note that the local similarity $S_l$ $(p, q)$ is associated with the features produced by only the MDLSA method, while the global similarity $S_g$ $(p, q)$ relies on the features obtained by the PCA.

## VI. CLUSTER TREE

When combining multiple views, we want to preserve cluster structures that are strongly suggested by the individual views. The idea is that if there was a strong separation between the data points in one of views, that separation should not be lost while combining the information from other views. In this it is proposed that building a hierarchical tree in a top-down fashion that uses the best view available at each split point in the tree.

## VII. CONCLUSION

In this paper we mainly focus on two common incapability of traditional statistics based semantic similarity measures for social tagging systems, e.g., unable to evaluate similarities among tags not co-occurred and unable to reflect the structural influence of the network of tag co-occurrence. Firstly, we propose a cluster tree based measure to evaluate the semantic similarity among random pair of tags. Secondly, we combine the cluster tree based measure and the statistics based measures into a hybrid one which can better reflect the structural influence of the network of tag co-occurrence.

## VIII. REFERENCES

[1] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.

[2] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. 22nd Annu. Int. SIGIR Conf.*, 1999, pp. 50–57. [6] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[3] N. Bouguila, "Clustering of count data using generalized Dirichlet multinomial distributions," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 4, pp. 462–474, Apr. 2008.

[4] M. Welling, M. Rosen-Zvi, and G. Hinton, "Exponential family harmoniums with an application to information retrieval," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, vol. 17, pp. 1481–1488.

[5] P. Gehler, A. Holub, and M. Welling, "The rate adapting Poisson model for information retrieval and object recognition," in *Proc. 23rd Int. Conf. Mach. Learn.*, Pittsburgh, PA, 2006, pp. 337–344.

[6] H. Zhang, T. W. S. Chow, and M. K. M. Rahman, "A new dualing
harmonium model for document retrieval," *Pattern Recognit.*, vol.42,
no. 11, pp. 2950–2960, 2009.

[7] A. Schenker, M. Last, H. Bunke, and A. Kandel, "Classification of web documents using graph matching," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 18, no. 3, pp. 475–496, 2004.

[8] M. Fuketa, S. Lee, T. Tsuji, M. Okada, and J. Aoe, "A document classification method by using field association words," *Inf. Sci.*, vol. 126, no. 1–4, pp. 57–70, 2000.

[9] C. M. Tan, Y. F.Wang, and C. D. Lee, "The use of bigrams to enhance text categorization," *Inf. Process. Manag.*, vol. 38, no. 4, pp. 529–546, 2002.

[10] M. L. Antonie and O. R. Zaiane, "Text document categorization by term association," in *Proc. IEEE Int. Conf. Data Mining*, 2002, pp. 19–26.

[11] P. Kanerva, J. Kristoferson, and A. Holst, "Random indexing of text samples for latent semantic analysis," in *Proc. 22nd Annu. Conf. Cognit. Sci. Soc.*, 2000, pp. 103–106.

[12] E. Gabrilovich and S.Markovitch, "Computing semantic relatedness using Wikipedia-based explicit semantic analysis," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2007, pp. 1606–1611.