Improving Latent Semantic Indexing with Concepts Mapping Based on Domain Ontology

Jingmin HAO, Lejian LIAO and Xiujie DONG

Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology

Beijing 100081, PRC

{Haojingmin & Liaolj & xiujie198}@bit.edu.cn

Abstract:

"Curse of dimensionality" is a common problem in the area of information retrieval. It was verified that points in a vector space are projected to a random subspace of suitably high dimension, and then the distances between the points are approximately preserved. Although such a random projection can be used to reduce the dimension of the document space, it does not bring together semantically related documents. Latent Semantic Indexing (LSI) projects documents to lower dimensional LSI space from higher dimensional term space with singular-value decomposition (SVD) for the purpose of reducing the dimensions of the document space and bringing together semantically related documents. But the computation time of SVD is a bottleneck because of the higher dimensions of documents. In this paper, a novel method of dimension reduction for improving LSI is provided. A term-to-concept projection matrix based on domain ontology was created in this method. This way documents were projected to lower dimensional concept space by the projection matrix. LSI pre-computation was performed not on the original term by document matrix, but on the lower dimensional concept by document matrix at great computational savings. Experiments indicate that this method improves the efficiency of LSI. And the similarity judgment between documents is not disturbed.

Keywords:

Latent Semantic Indexing, LSI, dimension reduction, domain ontology

1. Introduction

With the great success of the current World-Wide-Web, huge repositories of textual data from web documents have become available to a large public. "Curse of dimensionality" is a common problem in the area of information retrieval. Higher dimensions index structures make the efficiency of information retrieval systems rapidly decrease.

Clustering text documents into different category groups is an important step in indexing, retrieval, management and mining of abundant text data on the Web or in corporate information systems. Among others, the challenging problems of text clustering are big volume, high dimensionality and complex semantics. It was verified that points in a vector space are projected to a random subspace of suitably high dimension, and then distance between the points are approximately preserved.

In modeling a collection of documents for information access applications based on classical vector space model (VSM), the documents are often represented as a "bag of words", i.e., as term vectors composed of the terms and corresponding counts for each document. VSM assumes the independence of terms of documents. When directly using these representations, synonyms and polysemous terms, are not handled well. Methods for smoothing the term distributions through the use of latent classes have been shown to improve the performance of a number of information access tasks [13].

Latent Semantic Indexing (LSI) [1] is an approach to automatic indexing and information retrieval that attempts to overcome these problems by mapping documents as well as terms to a representation in the so called latent semantic space. In the latent semantic space, LSI tries to overcome the problems of lexical matching by using statistically derived conceptual indices instead of individual words for retrieval.

LSI usually takes the (high dimensional) vector space representation of documents based on term frequencies as a starting point and applies a dimension reducing linear projection [2]. The specific form of this mapping is determined by a given document collection and is based on a Singular Value Decomposition (SVD) [10] of the corresponding term by document matrix. The general claim is that similarities between documents or between documents and queries can be more reliably estimated in the reduced latent space representation than in the original representation. The rationale is that documents which share frequently co-occurring terms will have a similar representation in the latent space, even if they have no terms in common.

Although LSI has been applied with remarkable success in different domains, in real-time applications

978-1-4244-2780-2/08/$25.00 ©2008 IEEE

computation time is a bottleneck because of the calculation of SVD being quite difficult. We can perform the LSI pre-computation not on the original term document matrix but on a low dimensional projection at great computational savings and no great loss of accuracy [6].

Ontology defines concepts and the relationship of concepts. Domain ontology defines a common vocabulary for researchers who need to share information in a domain. In this paper, we provide a novel method to reduce the dimensions of documents vector space, which take advantage of the mapping from terms to concepts based on domain ontology. Experiments indicate that this method improves the efficiency of LSI and has no great loss of accuracy.

This paper is organized as follow: in section 2, we review related work in the area. In section 3, we describe the background of LSI. Section 4 gives a basic description of domain ontology about soccer domain. in this section we present how to use domain ontology instead of random projection to reduce the dimensionality of documents vector. Section 5 is experimental design, Section 6 gives the conclusions.

2. Related work

In the last decades, some work has been done about how to improve the performance of LSI. Literature [5] showed that projecting documents via LSI and truncation offers a dramatic advantage over full profile clustering in terms of time efficiency.

In [5], Thomas Hofmann presented Probabilistic Latent Semantic Indexing which is based on a statistical latent class model for factor analysis of count data. In contrast to standard Latent Semantic Indexing (LSI) by Singular Value Decomposition, the probabilistic variant has a solid statistical foundation and defines a proper generative data model.

A result by Johnson and Lindenstrauss [3] states that if points in a vector space are projected to a random subspace of suitably high dimension, then the distances between the points are approximately preserved. Although such a random projection can be used to reduce the dimension of the document space, it does not bring together semantically related documents. LSI on the other hand seems to achieve the latter, but its computation time is a bottleneck. This naturally suggests the following two-step approach [6]:

1. Apply a random projection [12] to the initial corpus to ℓ dimensions, for some small $\ell > k$, to obtain, with high probability, a much smaller representation, which is still very close (in terms of distances and angles) to the original corpus.

2. Apply rank O (k) LSI (because of the random projection, the number of singular values kept may have to be increased a little).

In [6], Christos H. Papadimitriou et al have researched on projection techniques for projecting the term by document matrix on a completely random low dimensional subspace, then, with high probability we have a distance preservation property akin to that enjoyed by LSI. This suggests that random projection may yield an interesting improvement on LSI. We can perform the LSI pre-computation not on the original term by document matrix, but on a low dimensional projection at great computational savings and no great loss of accuracy.

Random projection can be seen as an alternative to (and a justification of) sampling in LSI. The result of suggests a more elaborate (and computationally intensive) approach—projection on a random low-dimensional subspace—which can be rigorously proved to be accurate [6].

Pavel Moravec et al [4] offer a replacement of LSI with a projection matrix created from WordNet hierarchy and compare it with LSI. The conclusion of [4] shown that using WordNet ontology instead of random projections obtained better results and is even comparable with classical LSI. This could make the dimension reduction feasible even for huge document collections.

In contrast to [4, 6], we propose another novel method, using domain ontology instead of random projection and WordNet to make dimension reduction for domain special documents. It is based on the assumption that domain documents must share many domain special concepts, such that we can map term of documents onto domain special concept to make the dimension reduction feasible.

3. LSI background

3.1 What is LSI?

Latent Semantic Indexing (LSI) [9] is a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text. LSI is an algebraic extension of classical vector model [7, 8]. The key idea of LSI [1] is to map documents (and by symmetry terms) to a vector space of reduced dimensionality (using SVD to reduce the dimensions), the latent semantic space, attempts to solve the synonomy and polysemy problems that plague automatic information retrieval systems.

3.2 Description of LSI

A corpus is a collection of documents. Each document is a collection of terms from a universe of n terms. Each document can thus be represented as a vector in $\Re_n$ where each axis represents a term. The ith coordinate of a vector represents some function of the number of times the ith term occurs in the document represented by the vector. First, the term to document association is presented as a term by document matrix.

Let $\boldsymbol{A}$ be an $n \times m$ matrix of rank r whose rows represent terms and columns represent documents.

Theorem1. (Singular value decomposition)[10]: Let $\boldsymbol{A}$ is an $n \times m$ rank r matrix. Let the singular values of $\boldsymbol{A}$ (the eigenvalues of a matrix $\boldsymbol{AA}^{\mathrm{T}}$) be $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$ .Then there exist orthogonal matrices $U=(u_1,\cdots,u_r)$ and $V=(v_1,\cdots,v_r)$, whose column vectors are orthonormal, and a diagonal matrix $\sum = diag(\sigma_1,\Lambda,\sigma_r)$. The decomposition $\boldsymbol{A}=\boldsymbol{U}\sum\boldsymbol{V}^{\mathrm{T}}$ is called singular decomposition of matrix $\boldsymbol{A}$. Columns of U (or V) are called left (or right) singular vectors of matrix $\boldsymbol{A}$.

LSI works by omitting all but the k largest singular values in SVD, for some appropriate k; here k is the dimension of the low-dimensional space alluded to in the informal description in section one. It should be small enough to enable fast retrieval and large enough to adequately capture the structure of the corpus (in practice, k is in the few hundreds, compared with r in the many thousands). Let $\sum_k = diag(\sigma_1,\Lambda,\sigma_k)$ and $U_k=(u_1,\Lambda,u_k), V_k=(v_1,\Lambda,v_k)$, then

$$A_k = U_k \sum_k V_k^T .$$

$\boldsymbol{A}_{\mathrm{k}}$ is a matrix of rank k. The rows of $\boldsymbol{V}_{\mathrm{k}}\sum_{\mathrm{k}}$ above are then used to represent the documents. In other words, the column vectors of matrix $\boldsymbol{A}$ (documents) are projected to the k-dimensional space spanned by the column vectors of $\boldsymbol{U}_{\mathrm{k}}$. We sometimes call this space the LSI space of $\boldsymbol{A}$. LSI represents terms and documents in a new vector space with smaller dimensions that minimize the distance between the projected terms and the original terms, i.e., one obtains the approximation

$$A = U\sum V^T \approx U_k \sum_k V_k^T = A_k .$$

How good is this approximation? The following well known theorem gives us some idea (the subscript **F** denotes the Frobenius norm)

Theorem 2 [11]: Among all $n \times m$ matrices $\boldsymbol{C}$ of rank at most k, $\boldsymbol{A}_{\mathrm{k}}$ is the one that minimizes

$$\|A-C\|_F^2 = \sum_{i,j}(A_{i,j}-C_{i,j})^2 ,$$

Therefore LSI preserves (to the extent possible) the relative distances (and hence, presumably, the retrieval capabilities) in the term by document matrix while projecting it to a lower-dimensional space.

3.3 Updating of LSI

Suppose an LSI-generated database already exists. That is, a collection of text objects has been parsed, a term-document matrix has been generated, and the SVD of the term by document matrix has been computed. If more terms and documents must be added, two alternatives for incorporating them currently exist: recomputing the SVD of a new term by document matrix or *folding-in* the new terms and documents.

Updating refers to the general process of adding new terms and/or documents to an existing LSI-generated database. Updating can mean either folding-in or SVD-updating. *SVD-updating* is the new method of updating developed in [16]. Folding-in terms or documents is a much simpler alternative that uses an existing SVD to represent new information. Recomputing the SVD is not an updating method, but a way of creating an LSI-generated database with new terms and/or documents from scratch which can be compared to either updating method. Recomputing the SVD of a larger term-document matrix requires more computation time and, for large problems, may be impossible due to memory constraints. Recomputing the SVD allows the new p terms and q documents to directly affect the latent semantic structure by creating a new term-document matrix $\boldsymbol{A}_{(\mathrm{m+p})\times(\mathrm{n+q})}$, computing the SVD of the new term-document matrix, and generating a different $\boldsymbol{A}_{\mathrm{k}}$ matrix. In contrast, folding-in is based on the existing latent semantic structure, the current $\boldsymbol{A}_{\mathrm{k}}$, and hence new terms and documents have no effect on the representation of the pre-existing terms and documents. Folding-in requires less time and memory but can have deteriorating effects on the representation of the new terms and documents.

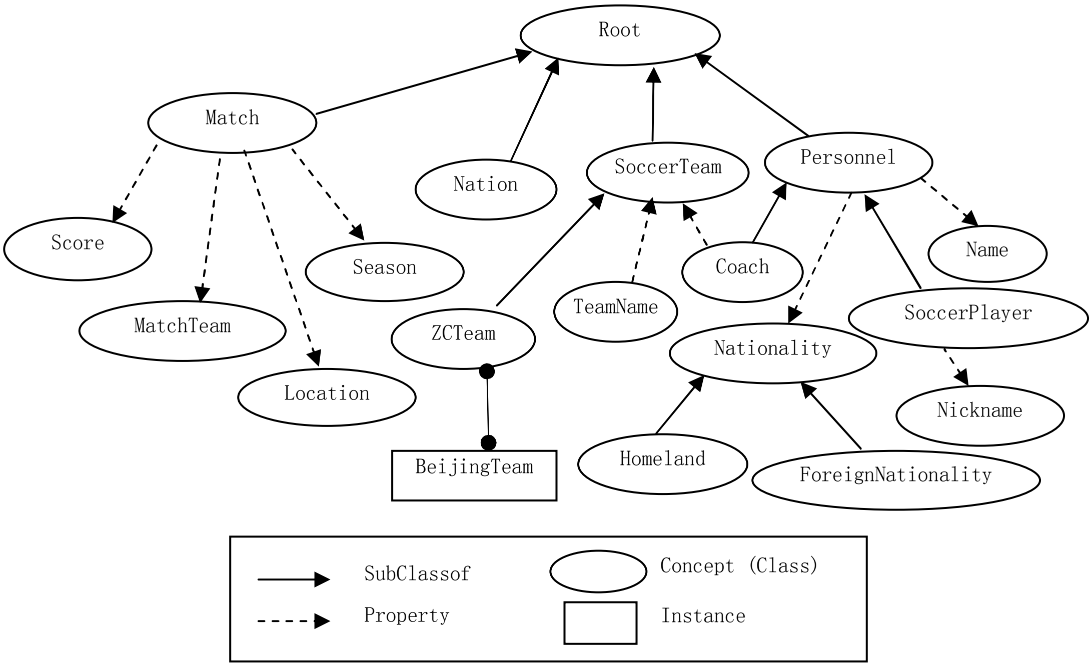


Figure 1. Portion of diagram of soccer domain ontology structure

Ideally, the most robust way to produce the best rank-*k* approximation (A_k) to a term by document matrix which has been updated with new terms and documents is to simply compute the SVD of a reconstructed term by document matrix, say $\tilde{\mathbf{A}}$. Updating methods which can approximate the SVD of the larger term by document matrix $\tilde{\mathbf{A}}$ become attractive in the presence of memory or time constraints. As discussed in [16], the the accuracy of SVD-updating approaches can be easily compared to that obtained when the SVD of $\tilde{\mathbf{A}}$ is explicitly computed.

4. Using domain ontology instead of random projection

Ontology is a formal specification of a shared conceptualization of a domain of interest to a group of users [17] [18]. The core of using ontology is to have shared meaning. Domain ontology usually describes terms, concepts and relationships widely used for a particular application domain. Domain ontology defines a common vocabulary for researchers who need to share information in a domain.

4.1 Ontology definition

Domain ontology can be defined by:

Definition 1 (domain Ontology): A domain ontology is 5-tuple *Ont*: = (*L, F, C, H, ROOT*), where

- *L* is a lexicon that contains a set of terms.
- *C* is a set of concepts: $C = \{c_1, c_2, \dots, c_n\}$.
- The reference function *F* with *F*: $2^L \mapsto 2^C$.

F links set of terms $\{L_i\} \subset L$ to the set of concepts they refer to. The inverse of *F* is F^{-1}.

- *H* is concepts hierarchy. Concepts are taxonomically related by the directed, acyclic, transitive, reflexive relation *H*, ($H \subset C \times C$). *H* (SoccerPlayer , Personnel) means that SoccerPlayer is a *subconcept* of Personnel.
- A *top concept* ROOT $\in$ *C*. For all $C \in C$ it holds: *H*(*C*, ROOT)

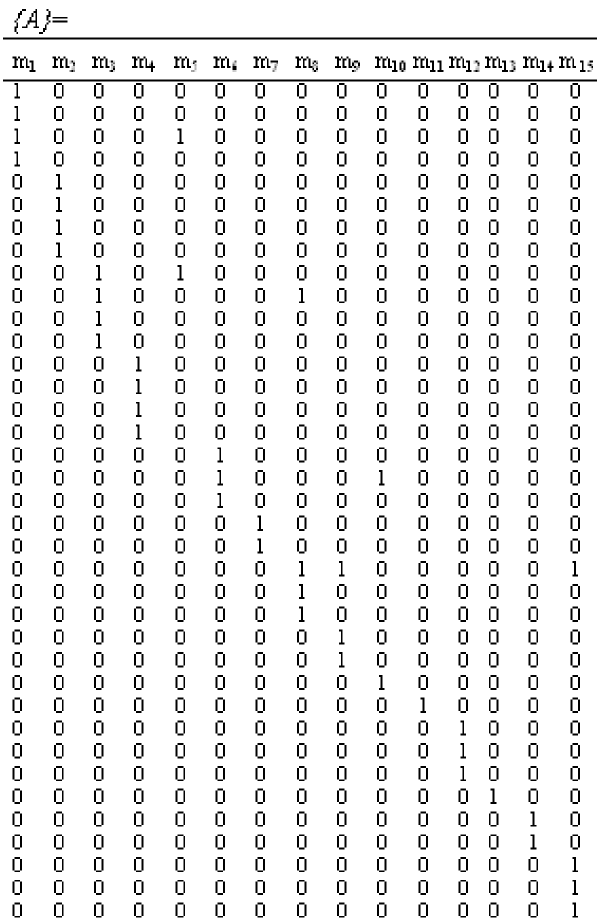
In this paper, we use domain ontology of soccer domain as the example. Figure 1 is the diagram of soccer domain ontology structure.

4.2 Create concept by document matrix

As mentioned above, the calculation of SVD is quite difficult and since the resulting matrices ***U*** and ***V*** are dense, memory can be exhausted quite quickly. So we face a question, how to create concepts space for given document collection.

One possibility is the usage of Papadimitriou's two-step algorithm [6] combining random projection [12] with LSI. Simply said, we first create a pseudoconcept-document matrix $\boldsymbol{A}_0$ with a suitable number of pseudoconcepts by multiplication of a zero-mean unit-variance projection matrix and the term by document matrix ***A***. In this step, we propose to use domain concepts from domain ontology instead of pseudoconcepts to create a concept by document matrix $\boldsymbol{A}_0$. Projection matrix is created with the mapping from term to concept in domain ontology. Our method realizes direct, visualized mapping from term to concept, which is viewed as semantics based mapping, and coincides with the spirit of LSI. Presumably, it should give a more good approximation of rank-*k* LSI of original matrix *A* than random projection.

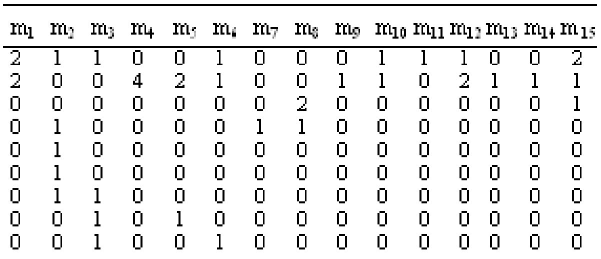
According to the definition of domain ontology, within a domain ontology, the reference function *F* links set of terms to the set of concepts they refer to. We can create projection matrix (from terms to concepts) based on reference function *F.* For example, *BeijingTeam* can be mapped to *ZCTeam.* Then, we create the concept by document matrix based on the projection matrix.



{A}=

m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	m_{11}	m_{12}	m_{13}	m_{14}	m_{15}
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 2. Original word-by-document matrix ***A***



{A0}=

m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	m_{11}	m_{12}	m_{13}	m_{14}	m_{15}
2	1	1	0	0	1	0	0	0	1	1	1	0	0	2
2	0	0	4	2	1	0	0	1	1	0	2	1	1	1
0	0	0	0	0	0	0	2	0	0	0	0	0	0	1
0	1	0	0	0	0	1	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0	0	0	0	0

Figure 3. the concept-by-document matrix $\boldsymbol{A}_0$ *after* projection

5. Experimental design

Here is a small example that gives the flavor of the analysis and demonstrates what our method accomplishes. In this experiment we use as text message the titles of fifteen web documents in Chinese about soccer news crawled from WWW. Thus the original matrix has fifteen columns, and we have given it 37 rows, each corresponding to a content word or Named Entity used in the titles. Because the titles wrote in Chinese we can not list them here. The corresponding original word-by-document matrix A is shown in Figure 2. Figure 3 is the reduction matrix A0 after projection. As we can see, dimensionality of document vectors in $\boldsymbol{A}_0$ is 9, dimensionality of document vectors in $\boldsymbol{A}$ is 37, the effect of dimension reduction is very obvious. Limited with space, we can not show the similarity matrix of $\boldsymbol{A}$ and $\boldsymbol{A}_0$. Figure 4 is a two-dimensional plot of the documents for the original term-document matrix; Figure 5 is a two-dimensional plot of the documents for the concept-document matrix after projection. According to what this two plot show, correlations between the fifteen texts semantically related objects became closer after reducing the dimensions of original term by document matrix based on the soccer domain ontology. That is, the method of dimension reduction based on concept mapping improves the performance of LSI in terms of text clustering.

5.1 Computational savings

What are the computational savings achieved by

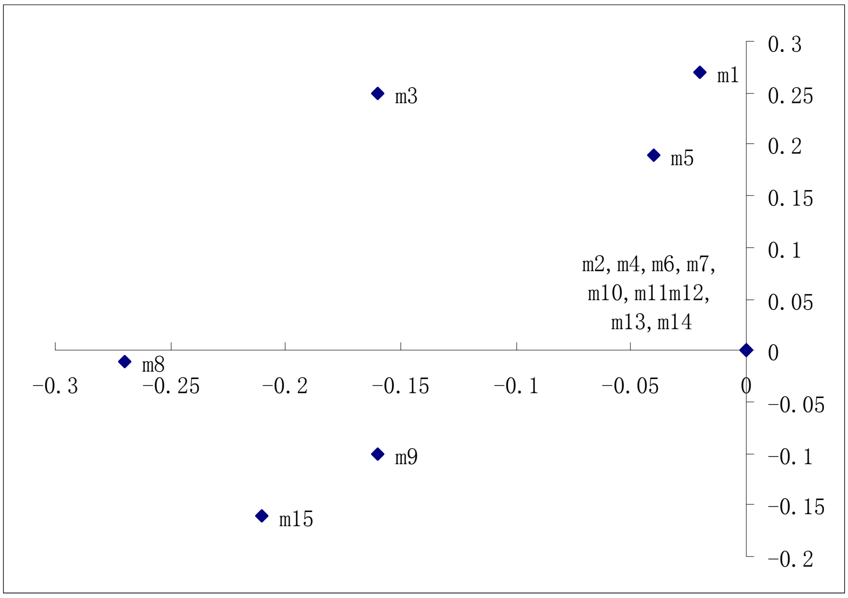


Figure 4. Two-dimensional plot of the documents for the original term-document matrix.

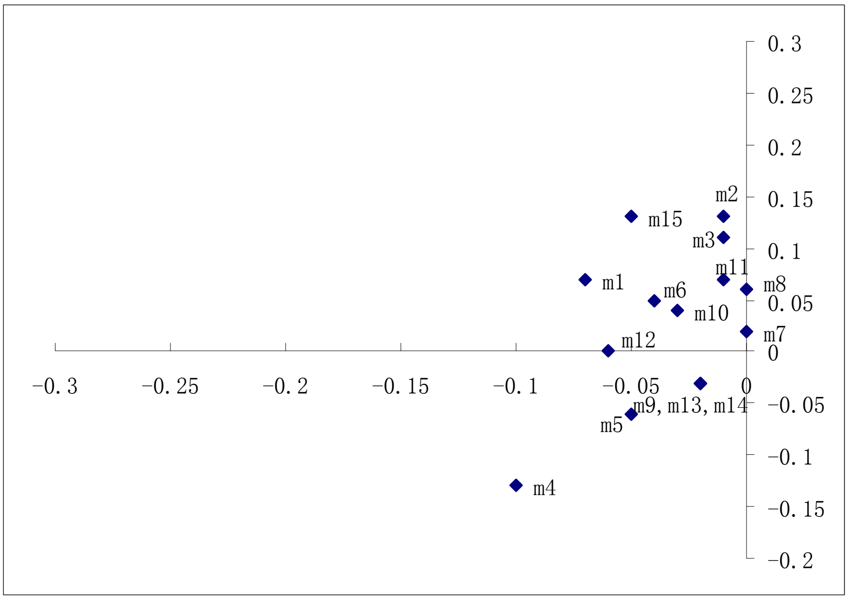


Figure 5. Two-dimensional plot of the documents for the reduction concept-document matrix.

dimension reduction. Let $\boldsymbol{A}$ be an $n \times m$ matrix. The time to compute LSI is O (mnc) if $\boldsymbol{A}$ is sparse with about c non-zero entries per column (i.e., c is the average number of terms in a document). The time needed to compute the projection to ℓ dimensions is O $(mc\ell)$. After the projection, the time to compute LSI is O $(m\ell)$. So the total time is O (m ℓ $(\ell + c)$). To obtain an approximation we need ℓ to be $O(\frac{\log n}{c^2})$. Thus the running time of our method is asymptotically superior: $O(m(\log^2 n + c\log n))$ compared to $O(mnc)$.

6. Conclusions

In this paper, we provide a novel method of dimension reduction for term by document matrix in LSI. We create the projection matrix based on domain ontology. Experiment show encouraged results. This method overcomes the problem of long time consume for SVD calculation, and preserve the correlation between text objects term by document matrix. So our method can improve the performance of LSI with a simple way in contrast to random projection.

Acknowledgements

This paper is supported by Beijing Key Discipline.

References

[1] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, Indexing by Latent Semantic Indexing, Journal of the American Society for Information Science, 41 (1990), pp. 391-407.

[2] Salton, G., and McGill, M. J. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.

[3] P. Frankl and H. Maehara. The Johnson-Lindenstrauss Lemma and the Sphericity of some graphs. J. Comb. Theory B 44 (1988), 355-362

[4] Pavel Moravec, Michal Kolovrat, Václav Snásel: LSI vs. WordNet Ontology in Dimension Reduction for Information Retrieval. DATESO 2004: 18-26

[5] Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In Proceedings of SIGIR-99, pages 35–44.

[6] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In Proocedings of the ACM Conference on Principles of Database Systems (PODS), pages 159–168, 1998.

[7] M. Berry and M. Browne. Understanding Search Engines, Mathematical Modeling and Text Retrieval. Siam, 1999.

[8] M. Berry, S. Dumais, and T. Letsche. Computation Methods for Intelligent Information Access. In Proceedings of the 1995 ACM/IEEE Supercomputing Conference, 1995.

[9] Landauer, T. K. & Dumais, S. T. (1997). A solution to Plato's problem: The Latent Semanctic Analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review, 104, 211-140

[10] G. Golub and C. Van Loan. Matrix Computations. Johns-Hopkins, Baltimore, Maryland, second edition, 1989.

[11] G. Golub and C. Reinsch. Handbook for matrix computation II, Linear Algebra. Springer-Verlag, New York, 1971.

[12] D. Achlioptas. Database-friendly random projections. In Symposium on Principles of Database Systems, 2001.

[13] David Guo, Michael Berry, Bryan Thompson, and Sidney Balin. Knowledge-enhanced latent semantic indexing. Information Retrieval, 6(2):225–250. 2003.

[14] Stanislaw Osifiski. Dimensionality Reduction Techniques for Search Results Clustering [D]. MSc. thesis, University of Sheffield, UK, 2 004.

[15] A Hotho, A Macdehe, S Staab. Onoglogy-based Text Clustering [A]. IJCAI-2001 Workshop.

[16] G. W. O'Brien, Information Management Tools for Updating an SVD-Encoded Indexing Scheme, Master's thesis, The University of Knoxville, Tennessee, Knoxville, TN, 1994.

[17] GRUBER CTR. A translation approach to portable ontologies[J]. Knowledge Acquistion. 1993, 5 (2):199-220.

[18] BORST WN. Construction of Engineering Ontologies for Knowledge Sharing and Reuse[D]. PHD thesis, University of Twente. Enschede, 1997.