An approach to look up documents in a library using singular value decomposition

Nguyen Thon Da†

†Faculty of Information Systems, HCMC University of Economics and Law, VNU, Ho Chi Minh, Vietnam

Summary
Building an search engine with high effect is really essential. The intent of this paper is to introduce the reader to SVD reduced dimension applied for looking up documents in the library under University of Economics and Law, HCMC, Vietnam. Using method LSI (Latent Semantic Indexing, we will compute a concept-based query vector and find out concept-based document vectors. We consider every query is a document. The problem given is finding out the relationship among us. Finally, We will compute the cosine between concept-based query vector with every of concept-based document vectors. The cosines which close to 1 the best is ones we need. From the cosines we choose, we will be easy to indicate relevant documents.

Key words:
SVD, singular value decomposition, information retrieval, text mining, searching document.

1. Introduction

1.1 Motivation

As we know, many retrieval systems match words in the user's queries with words in the text of documents. And one running in my school too. The disadvantage of matching words method are :The problem is that users want to retrieve on the basis of conceptual content, and individual words provide unreliable evidence about the conceptual topic or meaning of a document; there are usually many ways to express a given concept, so the literal terms in a user's query may not match those of a relevant document. Moreover, most words have multiple meanings, so terms in a user's query will literally match terms in documents that are not of interest to the user [12]. However, there is a better method using Latent Semantic Indexing (LSI). This method performs the match based on the concepts. In order to perform concept mapping, Singular Value Decomposition (SVD) is used. This is a concept-based retrieval method which overcomes many of the problems evident in today's popular word-based retrieval systems. For these reasons, we choose LSI model with a number of improvements to resolve the searching digital data in my school.

1. 2. Related works

A number of other researchers are using LSI information retrieval and classification work :
The reference [2] describes a method of LSI : The document database in the CACM test collection consists of all the 3204 articles. From this database, They built a term-by-document matrix of size 8258 × 2437. They evaluated the presented methodologies in terms of retrieval efficiency and accuracy, and compared it with the truncated SVD based method.
The reference [3] introduced a new intelligent approach to enhance the efficiency of IR system. The experiments that were conducted gave most promising results showing the superiority of their approach over traditional methods with VSM and LSI. The paper have given results : LSI with Traditional Approach with Average Precision is 59.06, LSI with Intelligent Approach with Average Precision is 60.99. According the authors, this system can be deployed in information intensive applications such as digital libraries.

2. Theoretical Consideration

2.1. Singular value decomposition

SVD is based on a theorem from linear algebra which says that a rectangular matrix A can be broken down into the product of three matrices - an orthogonal matrix U, a diagonal matrix S, and the transpose of an orthogonal matrix V . The theorem is usually presented something like this:
A = USVT where UTU = I; VTV = I; the columns of U are orthonormal eigenvectors of AAT , the columns of V are orthonormal eigenvectors of ATA, and S is a diagonal matrix containing the square roots of eigenvalues from U or V in descending order [12]. Figure 3 decribes the singular value decomposition of a m×n matrix.

Manuscript received May 5, 2016
Manuscript revised May 20, 2016

$$\begin{bmatrix} A \end{bmatrix} = \begin{bmatrix} U \end{bmatrix} \bullet \begin{bmatrix} S \end{bmatrix} \bullet \begin{bmatrix} V^T \end{bmatrix} \quad n < m$$

$$\begin{bmatrix} A \end{bmatrix} = \begin{bmatrix} U \end{bmatrix} \bullet \begin{bmatrix} S \end{bmatrix} \bullet \begin{bmatrix} V^T \end{bmatrix} \quad n \geq m$$

Figure 1 : The singular value decomposition of a m×n matrix

Source : [16]

The SVD of SST can therefore be used to recover an approximation to the SVD of A, while being computed from the SVD of S. SVD on S is only O(n2m) (assuming n < m), thus giving a fast approximation ([11])

2.2. Algorithms for the Singular Value Decomposition

According to [10], there are 12 algorithms of Singular Value Decomposition :

Algorithm1 is presented in three parts. It is analogous to the QR algorithm for symmetric matrices. Algorithm 1a is a Householder reduction of a matrix to bidiagonal form. Algorithm 1c is a step to be used iteratively in Algorithm 1b. Algorithm 2 computes the singular values and singular vectors of a bidiagonalmatrix to high relative accuracy. Algorithm 3 gives a "Squareroot-free" method to compute the singular values of a bidiagonal matrix to high relative accuracy. Algorithm 4 computes the singular values of an n×n bidiagonalmatrix by the bisection method, which allows k singular values to be computed in O(kn) time. By specifying the input tolerance tol appropriately, Algorithm 4 can also compute the singular values to high relative accuracy. Algorithm 5 computes the SVD of a bidiagonal by the divide and conquer method All of the above mentioned methods first reduce the matrix to bidiagonal form. The following algorithms iterate directly on the input matrix. Algorithms 6 and 7 are analogous to the Jacobi method for symmetric matrices. Algorithm 6 — also known as the "one-sided Jacobimethod for SVD" . Algorithm 7 begins with an orthogonal reduction of the m × n input matrix so that all the nonzeros lie in the upper n × n portion.

2.3. An approach SVD based-document retrieval

2.3.1 Term-document matrix

According to [1] :

Start with a dictionary of terms T1,T2,...,Tm. Terms are usually single words, but sometimes a term may contain more that one word such as "landing gear." It's up to you to decide how extensive your dictionary should be, but even if you use the entire English language, you probably won't be using more than a few hundred-thousand terms, and this is within the capacity of existing computer technology. Each document (or web page) Dj of interest is scanned for key terms (this is called indexing the document), and an associated document vector dj =(freq1j , freq2j ,..., freqmj)T is created in which freqij =number of times term Ti occurs in document Dj . After a collection of documents D1, D2,..., Dn has been indexed, the associated document vectors dj are placed as columns in a term-by-document matrix :

$$A_{mxn} = \begin{array}{c} \\ T_1 \\ T_2 \\ \vdots \\ Tm \end{array} \begin{array}{c} \begin{array}{cccc} d_1 & d_2 & \ldots & d_n \end{array} \\ \begin{bmatrix} freq_{11} & freq_{12} & \ldots & freq_{1n} \\ freq_{21} & freq_{22} & \ldots & freq_{2n} \\ \vdots & \vdots & & \vdots \\ freq_{m1} & freq_{m2} & \ldots & freq_{mn} \end{bmatrix} \end{array}$$

2.3.2. LSI algorithm

A pictorial representation of the SVD of input matrix A and the best rank-k approximation to A can be seen in Figure 4[14]

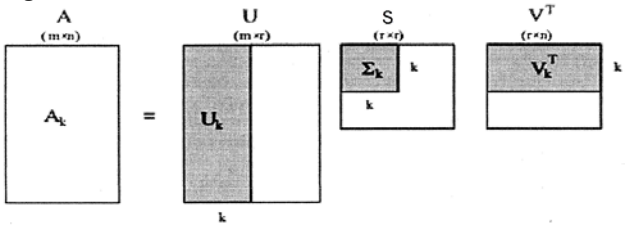

Figure 2 : Diagram of the truncated SVD

Source : [14]

In [7], Dr Garcia given 6 basic steps for LSI Algorithm. We added information from [4,17,18] to improve this algorithm :

Step 1: Score term weights and construct the term-document matrix A and query matrix.

Step 2: Decompose matrix A matrix and find the U, S and V matrices, where A = USVT

Step 3: Implement a Rank k Approximation by keeping the first columns of U and V and the first columns and rows of S.

Step 4: Find the new term vector coordinates in this reduced k-dimensional space.

What is the best rank, or k-dimensional vector space, to choose remains a question. Empirical testing suggests k = 100 – 300 dimensions is optimal for large datasets [4]

How many dimensions to retain? Total energy is sum of squares of singular values .

$$E = \sum_{i=1}^{n} \delta_{ii}^2$$

Retain dimensions such that p% of the energy is retained

$$E_k = \sum_{i=1}^{k} \delta_{ii}^2$$

$$E_k / E \geq p$$

Generally, is between 85% to 90% [17,18]
Step 5: Find the new query vector coordinates in the reduced k-dimensional space using

$$q = q^T U_k S_k^{-1}$$

Step 6: Group terms into clusters.

2.3.3 Similarity measures

The vector representation is given as $\vec{q}_i = (q_{1i}, q_{2i}, \ldots, q_{ti})^T$, where qzi is the weight of term z in the representation of the query $\vec{q}_i$. We estimate the similarity between a document and a query by the equation in figure 5.

$$\mathrm{Sim}_{\mathrm{VSM}}(\vec{q}_i, \vec{d}_j) = \frac{\sum_{z=1}^{t} (q_{zi} w_{zj})}{\sqrt{\sum_{z=1}^{t} q_{zi}^2} \sqrt{\sum_{z=1}^{t} w_{zj}^2}}.$$

Figure 3: The similarity between a document and a query

Source : [2]
After doing 6 step above, we applied the Similarity measures to indicate the documents we need to look up.

3. Experimental Consideration

3.1. Description of works

We created and set up a website including a search engine with details as below :
Configuration Information : Server :Starter VPS-Linux; Hardware : Intel Exon Lynnfield Quad-Core X3440, HDD : 40GB, 512MB DDR2 Centos 5; Software : Apache 2.2, PHP5, MySQL5.
Preparing Data : We survey actual state in the library of University of Economics and Law, this is base in order to we build the database of documents. In this research, we tested approximately 2200 materials. Our main goal is getting n documents relating to the query we input (suppose that n = 10). Thus, the result of our project is 10 documents which similar to the query. Information that we input contains one or more of ideals about title name, author name, and year published. The results we get is documents relating to what we inputted

3.2. Experiments

Suppose we need to look up the document as below :

Atlas learning centered communication vol.2 David Nunan 1995

Figure 4 : Some documents got after the query “Atlas learning centered communication vol.2 David Nunan 1995”was inputed

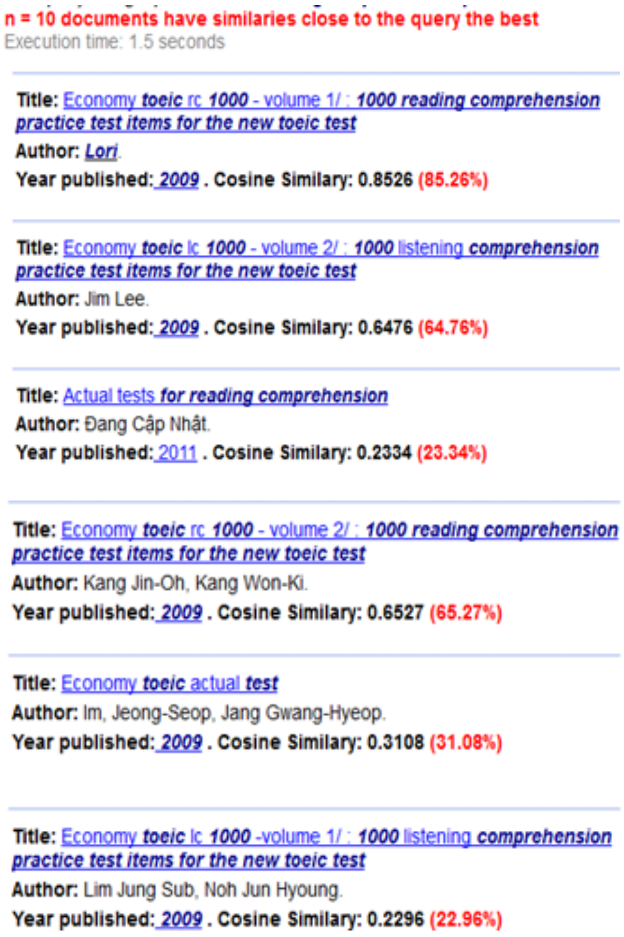

Figure 5 : Some documents got after the query “1000 reading comprehension practice test items for the new toeic test Lori 2009”was inputted.

4. Conclusion

In this paper, we collected all data from our database of school library – University of Economics and Law, HCMC, Vietnam. Different from other previous papers, those authors still confirmed the k-dimension is open question, we used k that we found out in [6]. After that, before applying SVD algorithm and using LSI, we had pre-process stages to remains potential documents. After calculating and testing, we gave results that could be acceptable in fact. Finally, base on the results got, we evaluated precision as done above.

Acknowledgment

The authors would like to express their cordial thanks to Prof., Dr. Bui Doan Khanh for his valuable advice.

References

[1] Carl D. Meyer, “Matrix Analysis and Applied Linear Algebra “, 2001
[2] Cherukuri Aswani Kumar, Suripeddi Sprivas, "Latent Semantic Indexing Using Eigenvalue Analysis for Efficient Information Retrieval", Int. J. Appl. Math. Comput. Sci, Vol. 16, No. 4, 2006, 551–558
[3] Ch. Aswani Kumar, Ankush Gupta,Mahmooda Batool, Shagun Trehan, “Latent Semantic Indexing-Based Intelligent Information Retrieval System for Digital Libraries”, Journal of Computing and Information Technology - CIT 14, 2006, 3, 191–196
[4] Dian I. Martin, "Mathematical and Computational Foundations For Latent Semantic Analysis", Small Bear Technical Consulting, LLC, www.SB-TC.com
[5] Dr. Arnab Bhattacharya, "Indexing and Searching Techniques in Databases", Course : CS618 (Indexing and Searching techniques in databases), 2011
[6] Dr. Edel Garcia, "Latent Semantic Indexing (LSI) A Fast Track Tutorial", September 6, 2006
[7] Dr. Edel Garcia, "Singular Value Decomposition (SVD), A fast track tutorial", September 6, 2006
[8] Dumais, S. T., Firnas, G. W., Landauer, T. K., Deerwester, S. and Harshman, R., “Using Latent Semantic Analysis to Improve Access to Textual Information”, Proceedings of the Conference on Human Factors in Computing Systems, Chicago, US, 1988
[9] Kirk Baker,” Singular Value Decomposition Tutorial”. March 29, 2005
[10] Leslie Hogben, Richard Brualdi, Anne Greenbaum and Roy Mathias, “Handbook of Linear Algebra”, 2007
[11] Michael P. Holmes, Alexander G. Gray and Charles Lee Isbell, Jr, ”Fast SVD for Large-Scale Matrices”, 2007
[12] M.W.Berry, S.T Dumais & G.W. O'Brien, "Using Linear Algebra for Intelligent Information Retrieval", December 1994
[13] Ravina Rodrigues, Kavita Asnani, "Concept Based Search Using LSI and Automatic Keyphrase Extraction", International Journal of Emerging Trends in Engineering and Technology (IJETET), Vol. I No. 1, 2011, ISSN 2248-9592
[14] Thomas K Landauer, Danielle S. McNamara, Simon Dennis, Walter Kintsch, "Handbook of Latent Semantic Analysis", 2007
[15] http://www.phpmath.com/build02/JAMA/
[16] http://www.exelisvis.com/docs/LA_SVD.html
[17] http://nptel.iitm.ac.in/courses/106104021/pdf_lecture/
[18] http://nptel.iitm.ac.in/courses/106104021/lecture29/

Nguyen Thon Da received the MSc degree from University of Information Technology, VNU in 2013. He used to work as a soildier (military service – from 2/2005 to 09/2006). Now he is working as an assistant teacher and an IT Lab employee (from 2007) in University of Economics and Law. His research interest includes data mining, SEO, information retrieval.