Available online at www.sciencedirect.com
ScienceDirect
Data & Knowledge Engineering 65 (2008) 108–125
DATA & KNOWLEDGE ENGINEERING
www.elsevier.com/locate/datak

# Augmenting the power of LSI in text retrieval: Singular value rescaling

Hua Yan [a], William I. Grosky [b,*], Farshad Fotouhi [c]

[a] CIS Department, Borough of Manhattan Community College, The City University of New York, New York, NY 10007, United States
[b] Department of Computer and Information Science, University of Michigan-Dearborn, Dearborn, MI 48128, United States
[c] Department of Computer Science, Wayne State University, Detroit, MI 48002, United States

Received 31 March 2006; received in revised form 13 August 2007; accepted 12 October 2007
Available online 20 November 2007

**Abstract**

This paper presents an analysis of several different LSI (latent semantic indexing) query approaches and proposes a novel rescaling technique, namely singular value rescaling (SVR). Experiments on a standardized TREC data set confirmed the effectiveness of SVR, showing an improvement ratio of 5.9% over the best conventional LSI query approach. In addition, we also compared SVR with another scaling technique in text retrieval called iterative residual rescaling (IRR). Experiments on TREC data set show that SVR performs better than IRR.


*Keywords:* Latent semantic indexing; Text retrieval; Singular value decomposition; Singular value rescaling

## 1. Introduction

The technique of latent semantic indexing (LSI, also known as latent semantic analysis or LSA) has been known to the information retrieval community since 1989 [7]. Since then, there have been volumes of research on this particular topic [11,3,5,30,24,25,9]. Through our work, we discovered that there are actually three different LSI query methods that are used without adequate distinction in the literature. This paper presents an analysis of these methods and also proposes a novel rescaling technique, namely singular value rescaling (SVR).

This paper is organized as follows: Section 2 introduces some background information about the SVD method, which is the mathematical engine behind the conventional LSI process. Section 3 gives a simple example to illustrate the use of LSI in text retrieval. Section 4 describes three commonly-used LSI query methods and their underlying philosophies. Section 5 provides an analysis of these three methods. Section 6 introduces our novel technique of singular value rescaling. Section 7 describes the experimental results on TREC data sets. Section 8 compares our SVR technique with another scaling technique used in text retrieval, namely

* Corresponding author.
*E-mail addresses:* hyan@bmcc.cuny.edu (H. Yan), wgrosky@umich.edu (W.I. Grosky), fotouhi@cs.wayne.edu (F. Fotouhi).

0169-023X/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2007.10.005

iterative residual rescaling (IRR). Section 9 provides a high-level conceptual explanation on how SVR works on top of the conventional SVD technique. Section 10 presents our conclusions.

## 2. Background

Most of the currently practiced LSI text retrieval techniques utilize the singular value decomposition (SVD) model, which works as follows. Assume that we have a collection of $d$ documents containing $t$ distinctive terms. Here, documents mean text sections and terms mean non-trivial words in these text sections. First, a term-by-document matrix, $A_0$, is constructed, where $A_0(i,j)$ corresponds to the frequency with which term $i$ occurs in document $j$, after any necessary term weighting has been performed [11]. $A_0$ then goes through the singular value decomposition process, where it is represented as the product of three matrices, $A_0 = U_0 S_0 V_0^{\mathrm{T}}$, where $A_0$ is a $t \times d$ matrix of rank $r$, $U_0$ is a $t \times r$ orthogonal matrix, $S_0$ is an $r \times r$ diagonal matrix with singular values $s_1 \geqslant s_2 \geqslant \cdots \geqslant s_r$ of $A_0$ in non-ascending order, and $V_0^{\mathrm{T}}$ is an $r \times d$ orthogonal matrix. Note that, in the context of text retrieval, document vectors can either refer to the column vectors $A_{0(:,i)}$ in $A_0$ or the column vectors $V_{0(:,i)}^{\mathrm{T}}$ in $V_0^{\mathrm{T}}$, and term vectors can either refer to the row vectors $A_{0(j,:)}$ in $A_0$ or the row vectors $U_{0(j,:)}$ in $U$.

The same nomenclature of document vectors and term vectors also applies to the dimensionally reduced model that is constructed in the next step: $A = USV^{\mathrm{T}}$, where $U$ is the first $k$ columns of $U_0$, $V^{\mathrm{T}}$ is the first $k$ rows of $V_0^{\mathrm{T}}$, and $S$ is a diagonal matrix with the $k$ largest singular values from $S_0$. Empirical evidence [7,8,11] shows that this dimensionally reduced model of $A$ better captures the hidden (latent) semantic structure of relations between terms and documents than does the original model, $A_0$. After these steps, we can now use the new model to compare two documents, two terms, or to determine the relationship between a term and a document [8]. The LSI model is a valuable information retrieval tool due to its capabilities for executing queries against the data it represents. Different researchers with different philosophies have come up with different ways of doing this, however, which we illustrate and analyze in the following sections.

## 3. An illustrative example

In Table 1, a term-by-document matrix $A_0$ is constructed using the highlighted terms shown in Fig. 1. If an element $A_0(i,j)$ has a value of $n$, this means that term $i$ occurs in document $j$ for a total of $n$ times.

Singular value decomposition is performed on $A_0 : \mathrm{SVD}(A_0) \rightarrow U_0 S_0 V_0^{\mathrm{T}}$, giving rise to the following three matrices: orthogonal matrix $U_0$ of dimension $11 \times 8$, diagonal matrix $S_0$ of dimension $8 \times 8$ with singular values $3.1262 \geqslant 2.1753 \geqslant 2.1225 \geqslant 2.0172 \geqslant 1.1260 \geqslant 1.0933 \geqslant 1.0000 \geqslant 0.6765$, and orthogonal matrix $V_0^{\mathrm{T}}$ of dimension $8 \times 9$. Now we construct the dimensionally reduced model. Note that so far there is no standard theory on how to choose the rank of the reduced dimension. Usually this step is done through trial and error. Choosing $k = 3$, we obtain the following matrices: $U$ of dimension $11 \times 3$, $S$ of dimension $3 \times 3$ with singular

Table 1
Term-by-document matrix $A_0$

| | Ontology topic | | | LSI topic | | | Image indexing topic | | |
|---|---|---|---|---|---|---|---|---|---|
| | *a1* | *a2* | *a3* | *b1* | *b2* | *b3* | *c1* | *c2* | *c3* |
| Ontology | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| RDF | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| XML | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Matrices | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| LSI | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| SVD | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Image | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| Indexing | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| Retrieval | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Introduction | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| Survey | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |