인터넷기반 정보 검색을 위한 LSI 활용 - QR 분해를 이용한 LSI 향상

박 유 진, 송 만 석
연세대학교 컴퓨터과학과 한글정보처리연구실
전화 : 02-2123-2711

LSI-Updating Application for Internet-based Information Retrieval - LSI Improvement Using QR Decomposition

You-Jin Park Man-Suk Song
Dept. of Computer Science, Yonsei University
Email : {romey, mssong}@december.yonsei.ac.kr

Abstract

This paper took advantage of SVD (Singular Value Decomposition) techniques of LSI(Latent Semantic Indexing) to grasp easily terminology distribution. Existent LSI did to static database, propose that apply to dynamic database in this paper. But, if dynamic applies LSI to database, updating problem happens.

Existent updating way is Recomputing method, Folding-in method, SVD-updating method. Proposed QR decomposition method to show performance improvement than existent three methods in this paper.

I. 서론

정보 검색 시스템(information retrieval system)은 사용자가 필요로 하는 정보를 수집하여 내용을 분석한 뒤 찾기 쉬운 형태로 조직하여, 정보에 대한 요구가 발생했을 때 해당 정보를 찾아 제공해 주는 시스템을 말한다. 즉 사용자가 요구하는 정보를 효과적으로 검색해주는 것을 목적으로 하고 있다. 이와 관련해서 다양한 한글 정보 검색 시스템과 응용 시스템이 개발되고 상용화되고 있다. 이전에 연구되어진 키워드 기반(keyword-based)의 검색 시스템 연구는 질의어(query)가 포함되어 있는 문서만을 검색하여, 검색하고자 하는 문서의 용어를 정확히 알지 못하는 경우에는 관련성이 적은 문서들이 많이 검색되고 있다. 키워드 기반에서의 단점은 사용자가 부여한 탐색어와 시스템이 문서를 인덱스한 색인어가 서로 일치하지 않아 동의어와 다의어 문제를 일으킨다. 이로 인해서 사용자는 부적당한 정보를 검색하거나 원하는 정보를 찾지 못하는 용어 문제가 발생한다.

본 논문에서는 용어 분포를 쉽게 파악하기 위하여 LSI(Latent Semantic Indexing : 잠재적 의미 색인)의 SVD(Singular Value Decomposition)기법을 이용하였다. 기존의 LSI는 정적인 데이터베이스만을 대상으로 하였으나, 본 논문에서 동적인 데이터베이스에 적용하는 것을 제안한다. 그러나 동적이 데이터베이스에 LSI를 적용을 하면 updating 문제가 발생하게 된다. 기존의 updating방식으로는 Recomputing방법, Folding-in 방법, SVD-updating방법이 있다. 본 논문에서는 기존의 세 가지 방법보다 성능 향상을 보이기 위해서 QR 분해(decomposition) 방법을 제안했다.

II. LSI(Latent Semantic Indexing) Updating

2.1 LSI(잠재적 의미 색인) 개념

LSI의 중요한 특징은 용어간의 상호 관련성이 자동

적으로 유도되고 검색효율을 향상시키는데 사용할 수 있다. LSI는 관련성을 모델화하기 위해서 SVD라는 통계적 기법을 사용하여 용어-문서 행렬을 k 값의 집합체로 분해한다. 각각의 용어와 문서는 k-차원의 LSI 공간에 벡터로써 표현되어지고 유사한 내용의 문서에 사용된 용어들이 이들 공간에서 유사한 값을 가지게 된다.

SVD를 거치면 설명력이 높은 순서로 원하는 수만큼의 고유벡터를 얻을 수 있다. 이 때 설명력이 작은 축은 잡음(noise)으로 간주함으로써 데이터의 차원을 줄이는 효과를 얻게 된다. 이 고유벡터들의 원래의 데이터 행렬에는 드러나 있지 않았던 의미구조를 나타나게 하므로 보다 심층적인 수준에서의 의미 분석을 가능하게 한다.

SVD는 문서간의 의미 구조를 파악하기 위해, 용어-문서(m*n)행렬에 SVD를 적용하여 k개의 벡터를 생성한다. k개로 분해된 벡터는 동일한 의미 공간상에 문서와 용어를 표현하는데 사용한다.

n개의 문서와 m개의 용어를 m*n(m≥n)으로 나타내고, k는 인자(factor)의 수, r은 A의 범위(rank)인 행렬 A_k라고 한다면, 행렬 A_k에 대한 SVD는 3가지 행렬의 곱으로 표현한다.

$$A_k = U\Sigma V^T \qquad (1)$$

U와 V는 각각 직교(orthonormal)하기 때문에 $U^TU=V^TV=I_n$가 성립한다. $\Sigma=diag(\sigma_1, \cdots, \sigma_n)$는 singular value들의 대각(diagonal)행렬인데 $1\le i\le r$ 사이에서는 양수값($\sigma_i>0$)을 갖고, $i\ge r+1$에서는 영($\sigma_i=0$)이다.

직교행렬인 U와 V의 처음 r개의 열들은 각각 AA^T와 A^TA의 r개의 고유값(eigenvalue)들과 연관된 직교 고유벡터(orthonormal eigenvector)를 정의한다. A의 singular value는 AA^T의 n개의 고유값들을 제곱한 양수들인 Σ의 대각 요소들로 정의된다. 이 표현을 그림으로 나타낸 것이 그림 1이고, 요소에 대한 설명은 표 1에 나타나 있다.

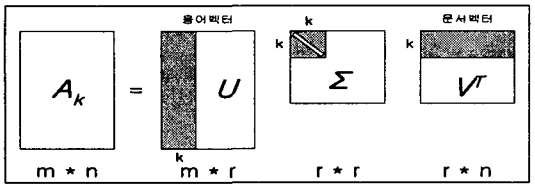

그림 1. 용어-문서 행렬(A_k)의 SVD 표현

A_k	Best rank-k approximation to A
U	Term Vectors
Σ	Singular Values
V	Document Vectors
m	Number of term
n	Number of documents
k	Number of factors
r	Rank of A

표 1. LSI의 SVD요소

2.2 LSI에 용어 및 문서 Updating

용어-문서 행렬이 생성되고, 용어-문서 행렬의 SVD가 계산되어진 LSI 데이터 베이스가 있다. 이 데이터베이스에 용어와 문서들이 추가되었을 때, 추가된 것을 기존에 존재하는 데이터베이스와 통합하는 방법에는 3가지가 있다.

첫째 - 새로운 용어-문서 행렬의 SVD를 Recomput-ing하는 방법.
둘째-새로운 용어나 문서를 Folding-in하는 방법.
셋째-SVD-updating을 하는 방법.

Updating은 새로운 용어나 문서 행렬을 이미 생성된 LSI 데이터베이스에 추가하는 과정을 말한다.

(1) Recomputing 방법

행렬 A에 새로운 행렬이 추가될 경우, 더 커진 용어-문서 행렬의 SVD를 Recomputing하는 것은 더 많은 계산 시간이 걸리고 메모리 제약이 뒤따른다. 새롭게 추가된 용어와 문서를 가지고 새로운 용어-문서 행렬을 만들어서 SVD를 구하기 때문에 잠재적 의미 구조에 직접적인 영향을 끼친다.

(2) Folding-in 방법

용어와 문서를 Folding-in하는 것은 기존에 잠재적 의미 구조에 기초를 두고 있기 때문에 존재하고 있는 용어-문서 표현에 영향을 주지 않는다.

① 문서(document) folding-in 방법

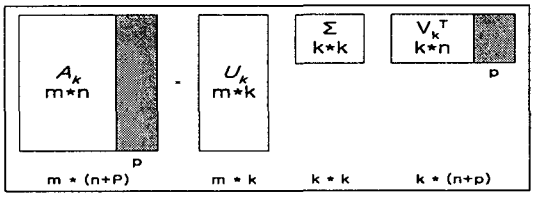

그림 2. p개의 문서를 Folding-in하는 수학적 방법

$$\hat{d} = d^T U_k \sum_k^{-1} \quad (2)$$

② 용어(term) folding-in 방법

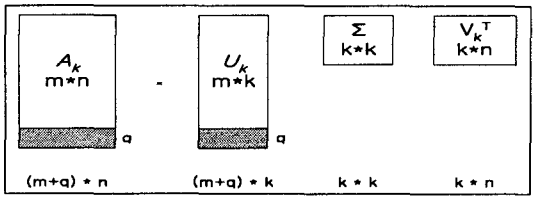

그림 3. q개의 용어를 Folding-in하는 수학적 방법

$$\hat{t} = tV_k \sum_k^{-1} \quad (3)$$

적은 시간과 메모리를 필요로 하지만, 새로 추가된 용어와 문서의 표현을 나쁘게 할 수 있다.

(3) SVD-Updating 방법

SVD-updating은 첫째-새로운 문서를 추가하는 단계, 둘째-새로운 용어를 추가하는 단계, 셋째-용어의 가중치의 변화를 교정하는 단계로 세 단계를 요구한다.

① 문서 추가 단계(Document updating)

$$U_B = U_k U_F,\ V_B = \begin{pmatrix} V_k & 0 \\ 0 & I_p \end{pmatrix} V_F,\ \Sigma_F = \Sigma_B \quad (4)$$

여기서 U_B와 V_B는 $m \times k$와 $(n+p) \times (k+p)$행렬로 나타낸다.

② 용어 추가 단계(Term updating)

$$U_C = \begin{pmatrix} U_k & 0 \\ 0 & I_q \end{pmatrix} U_H,\ V_C = V_k V_H,\ \Sigma_H = \Sigma_C \quad (5)$$

여기서 U_C와 V_C는 $(m+q) \times (k+q)$와 $n \times k$ 행렬로 나타낸다.

③ 용어 가중치 교정 단계

$$U_W = U_k U_Q,\ V_W = V_k V_Q \quad (6)$$

왜냐하면, $(U_Q U_k)^T W V_k V_Q = \Sigma_Q = \Sigma_W$ 이기 때문이다.

Ⅲ. QR 분해를 이용한 방법

QR 분해(decomposition)를 이용해서 SVD-updating 보다 더 정확한 수학적 모델을 제시하고, 새로운 추가(updating) 방법을 사용해서 더 좋은 검색 정확률을 보여준다.

3.1 QR 분해를 이용한 문서 추가 방법

$(I - P_k P_k^T)D$ 의 QR 분해식을 아래의 식과 같이 나타낼 수 있다.

$$(I - P_k P_k^T)D = \hat{P}_k R \quad (7)$$

위 식에서, $\hat{P}_k$는 orthonormal하고, R은 상 삼각(upper triangular) 행렬이다. 이 과정은 새로운 문서를 기존의 왼쪽 잠재적 의미구조에 투사(projects)하기 위해서이다.

3.2 QR 분해를 이용한 용어 추가 방법

$(I - Q_k Q_k^T)T^T$ 의 QR 분해식을 아래의 식과 같이 나타낼 수 있다.

$$(I - Q_k Q_k^T)T^T = \hat{Q}_k L^T \quad (8)$$

위 식에서, L은 하 삼각(lower triangular) 행렬이다. 이 과정은 새로운 문서를 기존의 오른쪽 잠재적 의미구조에 투사(projects)하기 위해서이다.

3.3 QR 분해를 이용한 용어 가중치 교정 방법

$(I - P_k P_k^T)X_j$와 $(I - Q_k Q_k^T)Y_j$ 의 QR 분해식을 아래의 식 (28)과 같이 나타낼 수 있다.

$$(I - P_k P_k^T)X_j = \hat{P}_k R_P,\ (I - Q_k Q_k^T)Y_j = \hat{Q}_k R_Q \quad (9)$$

위 식에서, R_P와 R_Q는 상 삼각(upper triangular) 행렬이다. 그리고 그것을 식 (10)과 같이 증명할 수 있다.

$$W = A_k + X_j Y_j^T = [P_k, \hat{P}_k] \left(\begin{bmatrix} \Sigma_k & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} P_k^T X_j \\ R_P \end{bmatrix} \begin{bmatrix} Q_k^T Y_j \\ R_Q \end{bmatrix} \right) [Q_k, \hat{Q}_k]^T \quad (10)$$

위 식에서, $[P_k, \hat{P}_k]$와 $[Q_k, \hat{Q}_k]$는 orthonormal이므로, 아래의 식 (11)과 같이 SVD 표현으로 나타낼 수 있다.

$$W = \begin{bmatrix} \Sigma_k & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} P_k^T X_j \\ R_P \end{bmatrix} \begin{bmatrix} Q_k^T Y_j \\ R_Q \end{bmatrix}^T = [U_k, U_k^{\perp}] \begin{bmatrix} \hat{\Sigma}_k & \\ & \hat{\Sigma}_n \end{bmatrix} [V_k, V_k^{\perp}]^T \quad (11)$$

위 식에서, U_k와 V_k는 k-차원의 열을 나타내고, $\hat{\Sigma}$는 k-차원의 특이치 벡터이다. W의 최상의 rank-k의 근사치는 아래의 식 (12)에 의해서 주어진다.

$$C_k = \left([P_k, \hat{P}_k]U_k\right)\hat{\sum}_k\left([Q_k, \hat{Q}_k]V_k\right)^T \qquad (12)$$

Ⅳ. 실험

본 논문에서는 인터넷에서 LSI Updating 문제를 해결하기 위해 기존의 LSI Updating 방법인 Folding-in 방법과 SVD-updating 방법을 이용해서 실험한 결과와 본 논문에서 제안하는 QR 분해(decomposition) 방법을 비교 분석해 보았다.

KTset95의 문서집합은 많은 문서를 포함하고 있기 때문에 모든 문서와 용어를 고려한 용어-문서 벡터를 구성하기는 사실상 힘들다. 그리고 구성을 했다고 하더라도 용어-문서 벡터를 SVD 계산하는데 걸리는 시간이나 시스템 부하가 너무 크기 때문에 제안한 방법이 효과적인지를 평가하기 위해서 KTset95의 문서집합을 다시 재구성하여 문서집합을 만들어 평가에 사용했다. 성능 평가를 하기 위해서 문서를 10개를 추출해 내고, 가중치를 부여한 색인어 12개를 추출해서 초기 LSI기법을 적용을 하였고, Updating을 하기 위해서 3개의 문서를 가지고 KTset95에서 가지고 와서 성능 평가를 하였다.

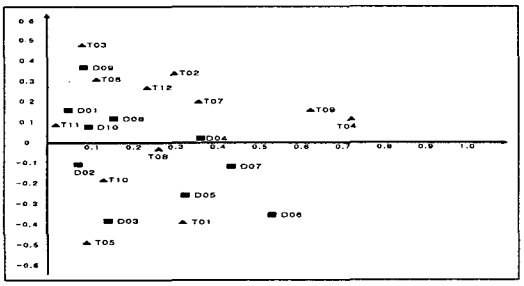

그림 4. Folding-in 방법 적용

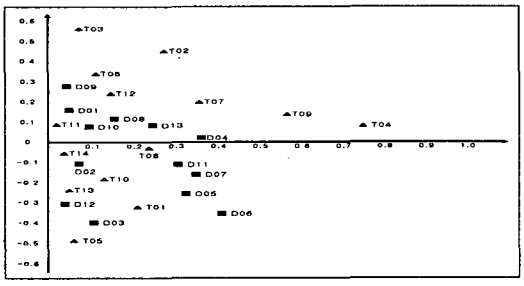

그림 5. SVD-Updating 방법

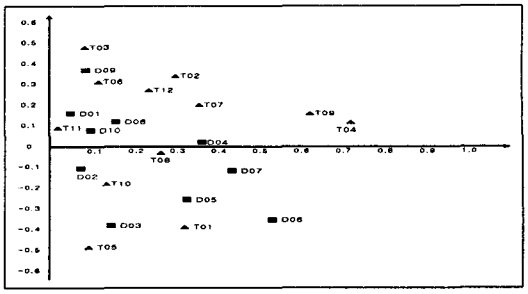

그림 6. 기존 문서의 2차원 표현

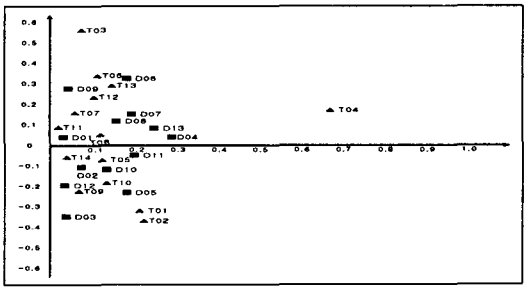

그림 7. QR 분해(decomposition) 방법 적용

Ⅴ. 결론

그림4에서부터 그림 7까지는 2차원 표현으로 용어-문서들의 벡터를 도식화한 것이다. 본 논문에서 LSI updating 방법들의 정확도를 측정하기 위해서 Recomputing 방법에 의해 구해진 U_2, V_2와 본 논문에서 다루고 있는 다른 방법들에 의해서 구해진 U_2, V_2 사이의 absolute distance를 구하였다.

	절 대 거 리(absolute distance)	
	용어 : 용어	문서 : 문서
Folding-in 방법	6.4361	7.0326
SVD-updating 방법	2.8047	2.2754
QR 분해 방법	0.3691	0.5238

표 2. 절대거리 비교(정확률 비교)

표의 결과를 보면 기존의 방법들 보다 QR 분해 방법이 정확도가 더 높은 것을 알 수가 있다.

참고문헌

[1] Todd. A. Letsche, "Toward Large-Scale Information Retrieval Using Latent Sementic Indexing", 1996

[2] M. Berry, S. Dumais, and G. O'Brien, "Using Linear Algebra for Intelligent Information Retrieval", *SIAM*, p. 573-595, 1995.