LBNL-41101
UC-405

ERNEST ORLANDO LAWRENCE
BERKELEY NATIONAL LABORATORY

On Updating Problems in Latent Semantic Indexing

Horst D. Simon and Hongyuan Zha
Computing Sciences Directorate

November 1997

LBNL-41101
UC-405

On Updating Problems in Latent Semantic Indexing

Horst D. Simon and Hongyuan Zha

Computing Sciences Directorate
Ernest Orlando Lawrence Berkeley National Laboratory
University of California
Berkeley, California 94720

November 1997

This work was supported by the Director, Office of Energy Research, Office of Laboratory Policy and Infrastructure Management, Office of Computational and Technology Research, Division of Mathematical, Information, and Computational Sciences, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098, and by the National Science Foundation under Grant No. CCR-9619452.

ON UPDATING PROBLEMS IN LATENT SEMANTIC INDEXING *

HORST D. SIMON † AND HONGYUAN ZHA ‡

Abstract. We develop new SVD-updating algorithms for three types of updating problems arising from *Latent Semantic Indexing* (LSI) for information retrieval to deal with rapidly changing text document collections. We also provide theoretical justification for using a reduced-dimension representation of the original document collection in the updating process. Numerical experiments using several standard text document collections show that the new algorithms give higher (interpolated) average precisions than the existing algorithms and the retrieval accuracy is comparable to that obtained using the complete document collection.

1. Introduction. Latent semantic indexing (LSI) is a concept-based automatic indexing method that tries to overcome the two fundamental problems which plague traditional lexical-matching indexing schemes: synonymy and polysemy [3].[1] Synonymy refers to the problem that several different words can be used to express a concept and the keywords in a user's query may not match those in the relevant documents while polysemy means that words can have multiple meanings and user's words may match those in irrelevant documents [7]. LSI is an extension of the vector space model for information retrieval [6, 9]. In the vector space model, the collection of text documents is represented by a *term-document* matrix $A = [a_{ij}] \in \mathcal{R}^{m\times n}$, where a_{ij} is the number of times term i appears in document j, and m is the number of terms and n is the number of documents in the collection. Consequently, a document becomes a column vector, and a user's query can also be represented as a vector of the same dimension. The similarity between a query vector and a document vector is usually measured by the cosine of the angle between them, and for each query a list of documents ranked in decreasing order of similarity is returned to the user. LSI extends this vector space model by modeling the term-document relationship using a *reduced-dimension representation* (RDR) computed by the singular value decomposition (SVD) of the term-document matrix A.[2] Specifically let

$$A = P\Sigma Q^T, \quad \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_{\min(m,n)}), \quad \sigma_1 \geq \ldots \geq \sigma_{\min(m,n)},$$

be the SVD of A. Then the RDR is given by the best rank-k approximation $A_k \equiv P_k\Sigma_k Q_k^T$, where P_k and Q_k are formed by the first k columns of P and Q, respectively, and Σ_k is the k-th leading principal submatrix of Σ. Corresponding to each of the k reduced dimensions is associated a pseudo-concept which may not have any explicit semantic content yet helps to discriminate documents [1, 3].

In rapidly changing environments such as the World Wide Web, the document collection is frequently updated with new documents and terms constantly being added. Updating the LSI-generated RDR can be carried out using a process called *fold-in*

* This work was supported by the Director, Office of Energy Research, Office of Laboratory Policy and Infrastructure Management, Office of Computational and Technology Research, Division of Mathematical, Information, and Computational Sciences, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098, and by NSF grant CCR-9619452.

† NERSC, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720.

‡ 307 Pond Laboratory, Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802-6103.

[1] LSI does a better job dealing with synonymy while polysemy still remains to be a problem unless word senses are used.

[2] Various weighting schemes can be applied to A before its SVD is computed [9]. Notice that alternative decompositions have also been used for LSI [5].

[3]. Fold-in is less expensive. However, since fold-in is based on the old RDR, it does not adjust the representation of existing terms and documents, and therefore retrieval accuracy may suffer. In [1, 8], three SVD-updating algorithms are derived focusing on the balance among memory usage, computational complexity and retrieval accuracy. The purpose of this paper is to point out an error in the derivation of the algorithms in [1, 8], and to show that better retrieval accuracy can be obtained with our new algorithms. In particular we show that no retrieval accuracy degradation will occur if updating is done with our new algorithms. The rest of the paper is organized as follows: In Section 2 we state the three types of updating problems in LSI and derive new algorithms for handling each of them. In Section 3 we provide theoretical justification for basing the updating process on the RDR of the old document collection. Section 4 presents several numerical experiments. Section 5 concludes the paper and points out some future research topics.

2. New Updating Algorithms. Let $A \in \mathcal{R}^{m\times n}$ be the original term-document matrix, and $A_k = P_k\Sigma_k Q_k^T$ be the best rank-k RDR of A. Following [1, 8], we specify three types of updating problems in LSI:

1. UPDATING DOCUMENTS. Let $D \in \mathcal{R}^{m\times p}$ be the p new documents. Compute the best rank-k approximation of
$$B \equiv [A_k, D].$$
2. UPDATING TERMS. Let $T \in \mathcal{R}^{q\times n}$ be the q new term vectors. Compute the best rank-k approximation of
$$C \equiv \begin{bmatrix} A_k \\ T \end{bmatrix}.$$
3. TERM WEIGHT CORRECTIONS. Let there be j terms that need term weight adjustment, $Z_j^T \in \mathcal{R}^{j\times n}$ specify the difference between the old weights and the new ones, $Y_j \in \mathcal{R}^{n\times j}$ be a selection matrix indicating the j terms that need adjusting. Compute the best rank-k approximation of
$$W \equiv A_k + Y_j Z_j^T.$$

Notice that in all the above three cases instead of the original term-document matrix A, we have used A_k, the best rank-k approximation of A as the starting point of the updating process. Therefore we may not obtain the best rank-k approximation of the true new term-document matrix. This replacement procedure needs to be justified and we will have more to say on this later in Section 3.

Now we present our new algorithms for the three types of updating problems mentioned above. During the presentation, we will also compare our approaches with those used in [1, 8].

UPDATING DOCUMENTS. Let the QR decomposition of $(I - P_kP_k^T)D$ be
$$(I - P_kP_k^T)D = \hat{P}_k R,$$
where $\hat{P}_k$ is orthonormal, and R is upper triangular. For simplicity we assume R is nonsingular.[3] It can be verified that
$$B \equiv [A_k, D] = [P_k, \hat{P}_k]\begin{bmatrix} \Sigma_k & P_k^T D \\ 0 & R \end{bmatrix}\begin{bmatrix} Q_k^T & 0 \\ 0 & I_p \end{bmatrix}.$$

[3] If $(I - P_kP_k^T)D$ is not of full column rank, R can be upper trapezoidal.

Notice that $[P_k, \hat{P}_k]$ is orthonormal. Now let the SVD of

$$\hat{B} \equiv \begin{bmatrix} \Sigma_k & P_k^T D \\ 0 & R \end{bmatrix} = [U_k, U_k^\perp] \begin{bmatrix} \hat{\Sigma}_k & \\ 0 & \hat{\Sigma}_p \end{bmatrix} [V_k, V_k^\perp]^T, \tag{2.1}$$

where U_k and V_k are of column dimension k, and $\hat{\Sigma}_k \in \mathcal{R}^{k\times k}$. Then the best rank-$k$ approximation of B is given by

$$B_k \equiv ([P_k, \hat{P}_k]U_k)\hat{\Sigma}_k \left(\begin{bmatrix} Q_k & 0 \\ 0 & I_p \end{bmatrix} V_k \right)^T.$$

In [1, 8], only $[\Sigma_k, P_k^T D]$ instead of $\hat{B}$ in (2.1) is used to construct the SVD of B. The R matrix in $\hat{B}$ is completely discarded. The SVD thus constructed is certainly not the *exact* SVD of B, and can not even be a good approximation of it if the norm of R is not small. This situation can happen when the added new documents alter the original low-dimension representation significantly. Numerical experiments in Section 4 bear this out.

Our approach is certainly more expensive than the less accurate alternative in [1, 8]: for one thing we need to compute the SVD of $\hat{B}$ instead of a submatrix of it; and also in order to form the left singular vector matrix of B we need to compute $[P_k, \hat{P}_k]U_k$ instead of $P_k\tilde{U}_k$, where $\tilde{U}_k$ is the left singular vector matrix of $[\Sigma_k, P_k^T D]$. However, if p, the number of documents added is relatively small, the added computational cost is not much.[4]

Our presentation for updating terms and for term weight corrections will be brief. The above comments regarding the algorithms in [1, 8] also apply in these two updating problems as well.

UPDATING TERMS. Let the QR decomposition of $(I - Q_kQ_k^T)T^T$ be

$$(I - Q_kQ_k^T)T^T = \hat{Q}_kL^T,$$

where L is lower triangular. Then

$$C \equiv \begin{bmatrix} A_k \\ T \end{bmatrix} = \begin{bmatrix} P_k^T & 0 \\ 0 & I_q \end{bmatrix} \begin{bmatrix} \Sigma_k & 0 \\ TV_k & L \end{bmatrix} [Q_k, \hat{Q}_k]^T.$$

Now let the SVD of

$$\hat{C} \equiv \begin{bmatrix} \Sigma_k & 0 \\ TV_k & L \end{bmatrix} = [U_k, U_k^\perp] \begin{bmatrix} \hat{\Sigma}_k & \\ 0 & \hat{\Sigma}_q \end{bmatrix} [V_k, V_k^\perp]^T,$$

where U_k and V_k are of column dimension k, and $\hat{\Sigma}_k \in \mathcal{R}^{k\times k}$. Then the best rank-$k$ approximation of C is given by

$$C_k \equiv \left(\begin{bmatrix} P_k & 0 \\ 0 & I_q \end{bmatrix} U_k \right)^T \hat{\Sigma}_k([Q_k, \hat{Q}_k]V_k)^T.$$

TERM WEIGHT CORRECTIONS. Let the QR decomposition of $(I - P_kP_k^T)X_j$ and $(I - Q_kQ_k^T)Y_j$ be

$$(I - P_kP_k^T)X_j = \hat{P}_kR_P, \quad (I - Q_kQ_k^T)Y_j = \hat{Q}_kR_Q,$$

[4] We will have more to say in Section 4 and Section 5 and for the case when p is large.

with R_P and R_Q upper triangular. Then it can be verified that

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

Notice that both $[P_k, \hat{P}_k]$ and $[Q_k, \hat{Q}_k]$ are orthonormal. Let the SVD of

$$\hat{W} \equiv \begin{bmatrix} \Sigma_k & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} P_k^T X_j \\ R_P \end{bmatrix} \begin{bmatrix} Q_k^T Y_j \\ R_P \end{bmatrix}^T = [U_k, U_k^\perp] \begin{bmatrix} \hat{\Sigma}_k & \\ 0 & \hat{\Sigma}_j \end{bmatrix} [V_k, V_k^\perp]^T,$$

where U_k and V_k are of column dimension k, and $\hat{\Sigma}_k \in \mathcal{R}^{k \times k}$. Then the best rank-$k$ approximation of W is given by

$$C_k \equiv ([P_k, \hat{P}_k] U_k) \hat{\Sigma}_k ([Q_k, \hat{Q}_k] V_k)^T.$$

3. Justification for the Use of A_k. We will concentrate on the DOCUMENT UPDATING PROBLEM in what follows. Notice that in updating we use the matrix $[A_k, D]$ instead of using the true new term-document matrix $[A, D]$ as would have been the case in traditional SVD updating problems. So it is a critical issue whether the replacement of A by its best rank-k approximation is justified for there is always the possibility that this process may introduce unacceptable error in the updated RDR. To proceed we introduce some notation: for any matrix $A \in \mathcal{R}^{m \times n}$, we will use $\text{best}_k(A)$ to denote its best rank-k approximation, and its singular values are assumed to be arranged in nonincreasing order,

$$\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_m(A).$$

Our first result compares the singular values of $[\text{best}_k(A), D]$ and $[A, D]$. As a convention when we compare the singular values of two matrices with the same number of rows but different number of columns we will count the singular values according to the number of rows. We now state two simple results without proof.

LEMMA 3.1. *Let $A \in \mathcal{R}^{m \times n}$. Let V be orthonormal. Then*

$$\sigma_i(AV^T) = \sigma_i(A), \quad i = 1, \ldots, m.$$

LEMMA 3.2. *Let $A = [A1, A2]$. Then $\sigma_i(A_1) \leq \sigma_i(A)$, $\quad i = 1, \ldots, m$.*

THEOREM 3.3. *Let $A \in \mathcal{R}^{m \times n}$ be the original term-document matrix, and let $D \in \mathcal{R}^{m \times p}$ represent the newly-added document vectors. Then*

$$\sigma_i([\text{best}_k(A), D]) \leq \sigma_i([A, D]), \quad i = 1, \ldots, m.$$

Proof. Let the SVD of A be

$$A = [P_k, P_k^\perp] \, \text{diag}(\Sigma_k, \Sigma_k^\perp) [Q_k, Q_k^\perp]^T$$

Then we have for $i = 1, \ldots, m$,

$$\begin{aligned} \sigma_i([A, D]) &= \sigma_i([[P_k, P_k^\perp] \, \text{diag}(\Sigma_k, \Sigma_k^\perp), D]) \\ &= \sigma_i([P_k \Sigma_k, D, P_k^\perp \Sigma_k^\perp]) \\ &= \sigma_i([P_k \Sigma_k Q_k^T, D, P_k^\perp \Sigma_k^\perp]) \text{ (by Lemma 3.1)} \\ &= \sigma_i([[\text{best}_k(A), D], P_k^\perp \Sigma_k^\perp]) \end{aligned}$$

Noticing that $[\text{best}_k(A), D]$ is a submatrix of $[[\text{best}_k(A), D], P_k^{\perp}\Sigma_k^{\perp}]$ we obtain the result by invoking Lemma 3.2. □

It is rather easy to find examples for which the strict inequalities hold in the above Theorem. Next we investigate under what conditions replacing A by $\text{best}_k(A)$ has no effect on the computed RDR. We first state the following result without proof.

LEMMA 3.4. *Let the SVD of $A \in \mathcal{R}^{m\times n}$ be $A = \sum_{i=1}^{m} \sigma_i u_i v_i^T$ with u_i and v_i the i-th left and right singular vector, respectively. Then for $p \geq k$ we have*

$$\text{best}_k(A) = \text{best}_k(A - \sum_{i=p+1}^{m} \sigma_i u_i v_i^T).$$

THEOREM 3.5. *Let $\hat{B} = [A, D]$,[5] $B = [\text{best}_k(A), D]$, where $A \in \mathcal{R}^{m\times n}$ and $D \in \mathcal{R}^{m\times p}$ with $m \geq (n+p)$. Moreover assume that*

$$\hat{B}^T\hat{B} = X + \sigma^2 I, \quad \sigma > 0,$$

where X is symmetric and positive semi-definite with $\text{rank}(X) = k$. *Then*

$$\text{best}_k(\hat{B}) = \text{best}_k(B).$$

Proof. The general idea of the proof is to show that what is discarded when A is replaced by $\text{best}_k(A)$ will also be discarded when $\text{best}_k(\hat{B})$ is computed from $\hat{B}$. To this end write

$$\hat{B}^T\hat{B} - \sigma^2 I = \begin{bmatrix} A^TA - \sigma^2 I & A^TD \\ D^TA & D^TD - \sigma^2 I \end{bmatrix}.$$

Since $\text{rank}(X) = k$, it follows that $\text{rank}(A^TA - \sigma^2 I) \leq k$ and $\text{rank}(D^TD - \sigma^2 I) \leq k$. Let the eigendecompositions of

$$A^TA - \sigma^2 I = V_A \,\text{diag}(\Sigma_A^2, 0)V_A^T, \quad D^TD - \sigma^2 I = V_D \,\text{diag}(\Sigma_D^2, 0)V_D^T,$$

where $\Sigma_A \in \mathcal{R}^{k_1\times k_1}, \Sigma_D \in \mathcal{R}^{k_2\times k_2}$ are nonsingular with $k_1 \leq k, k_2 \leq k$. We can write the SVD of A and D as follows:

$$A = U_A \,\text{diag}(\Sigma_A, \sigma I_{t_1})V_A^T = [U_A^{(1)}, U_A^{(2)}] \,\text{diag}(\Sigma_A, \sigma I_{t_1})[V_A^{(1)}, V_A^{(2)}]^T, \tag{3.2}$$

$$D = U_D \,\text{diag}(\Sigma_D, \sigma I_{t_2})V_D^T = [U_D^{(1)}, U_D^{(2)}] \,\text{diag}(\Sigma_D, \sigma I_{t_2})[V_D^{(1)}, V_D^{(2)}]^T, \tag{3.3}$$

where $U_A^{(1)} \in \mathcal{R}^{m\times k_1}, U_D^{(1)} \in \mathcal{R}^{m\times k_2}$, and $t_1 = n - k_1, t_2 = p - k_2$, respectively. Now write $V_A^TA^TDV_D$ in a partitioned form as

$$V_A^TA^TDV_D = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}, \quad S_{11} \in \mathcal{R}^{k_1\times k_2}. \tag{3.4}$$

Since $X = \hat{B}^T\hat{B} - \sigma^2 I$ is symmetric positive semi-definite and $\text{rank}(X) = k$, it follows that $S_{12} = 0, S_{21} = 0, S_{22} = 0$ and $k_1 + k_2 = \text{rank}(X) = k$. Using the SVD of A and D in (3.2) and (3.3), Equation (3.4) becomes

$$[U_A^{(1)}\Sigma_A, \sigma U_A^{(2)}]^T[U_D^{(1)}\Sigma_D, \sigma U_D^{(2)}] = \begin{bmatrix} S_{11} & 0 \\ 0 & 0 \end{bmatrix},$$

[5] The $\hat{B}$ defined here is different from that in (2.1).

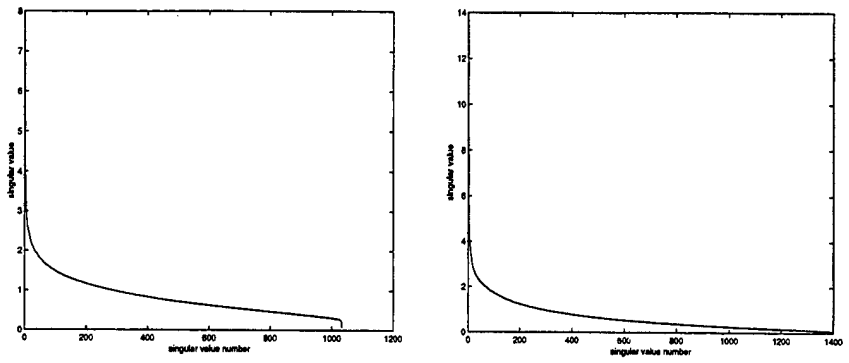

FIG. 1. *Singular value distributions:* 3681 × 1033 *term-document matrix of* MEDLINE *Collection (left) and* 2331 × 1400 CRANFIELD *Collection (right)*

which leads to[6]

$$U_A^{(1)} \perp U_D^{(2)}, \quad U_A^{(2)} \perp U_D^{(1)}, \quad U_A^{(2)} \perp U_D^{(2)}.$$

Let $\hat{U}$ be an orthonormal basis of

$$\mathcal{R}([U_A^{(1)}, U_D^{(1)}]) \cap \mathcal{R}([U_A^{(2)}, U_D^{(2)}])^{\perp},$$

where we have used $\mathcal{R}(\cdot)$ to denote the column space of a matrix, and $\mathcal{R}(\cdot)^{\perp}$ the orthogonal complement of the column space. Then we can write

$$[A, D] = [\hat{U}, U_A^{(2)}, U_D^{(2)}] \operatorname{diag}(\tilde{B}, \sigma I_{t_1}, \sigma I_{t_2}) \begin{bmatrix} (V_A^{(1)})^T & 0 \\ 0 & (V_D^{(1)})^T \\ (V_A^{(2)})^T & 0 \\ 0 & (V_D^{(2)})^T \end{bmatrix},$$

where $\tilde{B} \in \mathcal{R}^{k \times k}$ with all its singular values greater than σ. Therefore,

$$\hat{B} = [A, D] = [\hat{U}, U_D^{(2)}] \operatorname{diag}(\tilde{B}, \sigma I_{t_2}) \begin{bmatrix} (V_A^{(1)})^T & 0 \\ 0 & (V_D^{(1)})^T \\ 0 & (V_D^{(2)})^T \end{bmatrix} + \sigma U_A^{(2)} [(V_A^{(2)})^T, 0].$$

the right hand side of the above is easily seen to be $B + \sigma U_A^{(2)} [(V_A^{(2)})^T, 0]$, and the relation $\text{best}_k(\hat{B}) = \text{best}_k(B)$ then follows from Lemma 3.4. □

The matrix $\hat{B}^T \hat{B}$ in Theorem 3.5 has a so-called *low-rank-plus-shift* structure, a concept that has been used in sensor array signal processing [11, 12]. We now assess how well this structure can fit the term-document matrices of some standard document collections. Figure 1 plots the singular value distributions of two term-document matrices, one from the MEDLINE collection, the other CRANFIELD collection [2] used in the next section as well. We compute a low-rank-plus-shift approximation of a term-document matrix A in the following way: Let the SVD of A be $A = P_k \Sigma_k Q_k^T + P_k^{\perp} \Sigma_k^{\perp} (Q_k^{\perp})^T$, and let σ be the mean of the diagonal elements of $\Sigma_k^{\perp}$. Then the approximation is taken to be $A^{(k)} \equiv P_k \Sigma_k Q_k^T + \sigma P_k^{\perp} (Q_k^{\perp})^T$. For the MEDLINE collection we have $\|A - A^{(100)}\|_F / \|A\|_F = 0.2909$ and for the CRANFIELD collection we have $\|A - A^{(200)}\|_F / \|A\|_F = 0.2786$.

[6] we use $S \perp T$ to denote $S^T T = 0$.

TABLE 1
Comparison of average precisions for MEDLINE *collection*

p	100	200	300	400	500	600	700
s	933	833	733	633	533	433	333
Meth_1	65.36	65.52	66.58	67.16	66.98	66.56	66.48
Meth_2	64.26	64.40	58.48	50.78	46.90	44.04	44.97
Increm	65.36	65.61	65.61	66.33	66.65	66.58	66.75

4. Numerical Experiments. In this section we use several examples to illustrate the algorithms developed in Section 2 and compared them with those in [1, 8]. In all of the examples, we use the weighting scheme `lxn.bpx` [5, 9]. The partial SVD of the original term-document matrix is computed using Lanczos process with *one-sided* reorthogonalization scheme proposed in [10]. For each method and the corresponding parameters, we tabulate the average precision in percentage which is computed using the 11-point interpolated formula [4, 5]. All the computations are done on a Sun Ultra I workstation using MATLAB 5.0.

EXAMPLE 1. We use the MEDLINE text collection [2]. The term-document matrix is 3681×1033 and the number of queries is 30. The RDR is computed using a two-step method based on updating: for a given s we compute a rank-k approximation of the first s columns of the term-document matrix using the Lanczos SVD process, and then we add the remaining documents to produce a new rank-k approximation using updating algorithms. In Table 1, $k = 100$, p is the number of new documents added, Meth_1 is the updating algorithm in Section 2 and Meth_2 is that used in [1, 8]. Row 3 and row 4 of the table gives the average precisions in percentage. As is expected Meth_1 performs much better than Meth_2 for those seven comninations of p and s. What is surprising is that Meth_1 performs even better than rank-k approximation using the whole term-document matrix for which the average precision is 65.50%.

Instead of updating a group of p new documents all at once, we also carry out a test by breaking these p new documents into subgroups of 100 documents each, and use the updating algorithms to update one subgroup at a time. Row 5 of Table 1 gives the computed average precisions for $k = 100$ for our updating algorithm. Since the algorithms in [1, 8] always discard the R matrix in (2.1) therefore it makes no difference to the updated low-rank approximation whether it is computed with all the new documents all at once or incrementally with each subgroup at a time.

EXAMPLE 2. We repeat the tests in Example 1 for the CRANFIELD collection [2]. The term-document matrix is 2331×1400 and the number of queries is 225. Table 2 gives the results of the computations. For this example, the dimension for the RDR is chosen to be $k = 200$. In the incremental method we again update a subgroup of 100 documents at a time.

EXAMPLE 3. We use the 4322×11429 term-document matrix from the NPL

TABLE 2
Comparison of average precisions for CRANFIELD *collection*

p	100	200	300	400	500	600	700	800	900
s	1300	1200	1100	1000	900	800	700	600	500
Meth_1	41.53	41.26	41.70	41.38	41.81	41.53	41.58	41.48	41.36
Meth_2	41.89	41.65	42.08	41.03	39.24	37.58	34.65	32.11	29.38
Increm	41.53	41.30	41.63	41.57	41.43	41.36	41.14	41.30	41.36

TABLE 3
Comparison of average precisions for NPL *collection*

p	200	400	600	800	1000	1200	1400
s	4122	3922	3722	3522	3322	3122	2922
Meth_2	22.34	20.87	19.72	19.16	17.88	17.72	17.60
Increm	22.66	22.37	22.32	22.47	22.11	22.04	22.16

collection [2]. The number of queries is 100. We apply the TERM-UPDATING algorithm in Section 2. Since the original term-document matrix has the terms sorted in nonincreasing document frequency, we apply a random permutation to the rows of the term-document matrix before we extract any submatrix. For a given s we compute a rank-k approximation of the first s rows of the permuted term-document matrix using the Lanczos SVD process, and then we add the remaining terms to produce a new rank-k approximation. For both Meth_2 and INCREM we add 100 document at a time.

5. Concluding Remarks. We showed that better average precisions can be obtained using the updating algorithms developed in this paper. We also provided theoretical justification for basing the updating procedures on the RDR of the original document collection. We have only presented a result assuming exact low-rank-plus-shift structure. In future research we will consider the case when the low-rank-plus-shift structure only holds approximately. We also have used an incremental approach to handle the case when the number of new documents is large. Another approach will be first to find the RDR of the set of new documents and then merge it with the RDR of the original document collection. These issues will be discussed in a forthcoming paper.

REFERENCES

[1] M.W. Berry, S.T. Dumais and G.W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37:573-595, 1995.

[2] Cornell SMART System, `ftp://ftp.cs.cornell.edu/pub/smart`.

[3] S. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas and R.A. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41:391–407, 1990.

[4] D. Harman. TREC-3 conference report. NIST Special Publication 500-225, 1995.

[5] T.G. Kolda and D.P. O'Leary. A semi-discrete matrix decomposition for latent semantic indexing in information retrieval. Technical Report UMCP-CSD CS-TR-3724, Department of Computer Science, University of Maryland, 1996.

[6] G. Kowalski. Information Retrieval System: Theory and Implementation. Kluwer Academic Publishers, Boston, 1997.

[7] R. Krovetz and W.B. Croft. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems*, 10:115–141, 1992.

[8] G.W. O'Brien. Information Management Tools for Updating an SVD-Encoded Indexing Scheme. M.S. Thesis, Department of Computer Science, Univ. of Tennessee, 1994.

[9] G. Salton. Automatic Text Processing. Addison-Wesley, New York, 1989.

[10] H.D. Simon and H. Zha. Low rank matrix approximation using the Lanczos bidiagonalization process with applications. Technical Report CSE-97-008, Department of Computer Science and Engineering, The Pennsylvania State University, 1997.

[11] G. Xu and T. Kailath. Fast subspace decompsotion. *IEEE Transactions on Signal Processing*, 42:539–551, 1994.

[12] G. Xu, H. Zha, G. Golub, and T. Kailath. Fast algorithms for updating signal subspaces. *IEEE Transactions on Circuits and Systems*, 41:537–549, 1994.

Report Number (14) LBNL--41101

Publ. Date (11) 199711

Sponsor Code (18) DOE/ER; NSF, XF

UC Category (19) UC-405; UC-000, DOE/ER

DOE