Extended Lanczos Bidiagonalization for Dimension Reduction in Information Retrieval

Xuansheng Wang *†§, François Glineur*†, Paul Van Dooren* and Linzhang Lu‡§
*ICTEAM Institute, Université catholique de Louvain, B-1348 Louvain-La-Neuve, Belgium
†CORE, Université catholique de Louvain, B-1348 Louvain-La-Neuve, Belgium
‡School of Mathematics and Computer Science, Guizhou Normal University, Guiyang 550001, P. R. China
§School of Mathematical Science, Xiamen University, Xiamen 361005, P. R. China

***Abstract*—We describe an extended bidiagonalization scheme designed to compute low-rank approximations of very large data matrices. Its goal is identical to that of the truncated singular value decomposition, but it is significantly cheaper. It consists in an extension of the standard Lanczos bidiagonalization that improves its approximation capabilities, while keeping the computational cost reasonable. This low-rank approximation yields much cheaper computations of the matrix-vector products that are central in many information retrieval tasks. We demonstrate effectiveness of this approach on applications in face recognition and latent semantic indexing.**

***Index Terms*—: Dimension Reduction, Latent Semantic Indexing, Eigenfaces, Singular Value Decomposition, Principal Components Analysis, Lanczos Bidiagonalization**

I. Introduction

The problem of dimension reduction has received a lot of attention in areas such as databases, data mining, machine learning, and information retrieval [8], [10], [12]. Latent semantic indexing (LSI) and principal component analysis (PCA) are for instance two well-known applications that make use of some form of dimension reduction.

Solving the dimension reduction problem consists in finding a good $r-$dimensional approximation of a given subspace. When that subspace is the span of a given matrix, this is equivalent to finding a good low-rank approximation of that matrix. It is well-known that the truncated singular value decomposition provides the best possible low-rank approximation in the Frobenius norm sense, but is expensive to compute because its complexity is cubic in the matrix dimensions.

When r is large, the Lanczos bidiagonalization method [3] can often provide a good rank-r approximation at a much lower cost, even though it is not optimal. When r is small, though, the quality of the approximation can be very bad compared to that of the optimal low-rank approximation. To remedy this problem, we propose in this paper a novel method, which we call *extended Lanczos bidiagonalization*, which is almost as cheap as Lanczos bidiagonalization, but provides a much better $r-$dimensional approximation, even if r is small.

The rest of this paper is organized as follows. In section 2, we review briefly the applications of latent semantic indexing and of classification based on eigenfaces. In section 3, we then recall the Lanczos bidiagonalization algorithm and give some of its properties. In section 4, we give a detailed description of the extended Lanczos bidiagonalization algorithm and in section 5, we describe numerical experiments to show its effectiveness. We end with a brief section of concluding remarks.

II. Latent Semantic Indexing and Principal Component Analysis

In this section, we briefly recall the techniques of latent semantic indexing in information retrieval and of principal component analysis in face recognition.

A. Latent Semantic Indexing (LSI)

Latent semantic indexing (LSI) [12] has become a standard technique to retrieve from a very large database of documents, those documents that correspond to a particular set of words (called a *query*). The method is based on *the assumption that there is some underlying latent semantic structure in the data ... that is corrupted by the wide variety of words used [6] and that this semantic structure can be discovered and enhanced by projecting the data (the term-document matrix) onto a lower-dimensional space.*

The term-document matrix A is a very large matrix where $A(i,j)$ indicates how many times word i occurs in document j, so that each of its columns a_j corresponds to a document. The method of choice for the projection of A on the set of low-rank matrices, is the Singular Value Decomposition (SVD):

$$A = U\Sigma V^T, \tag{1}$$

yielding a low-rank approximation

$$A_r = U_r\Sigma_r V_r^T, \tag{2}$$

where U_r (respectively, V_r) consists of the first r columns of U (respectively, V) and Σ_r is the rth principal submatrix of diagonal Σ. This matrix A_r is a best rank-r approximation of A in the 2-norm and in the Frobenius norm sense [3]. In latent semantic indexing, the columns of U_r live in the document space and form an orthogonal basis that we use to approximate the documents. If we write $H_r = \Sigma_r V_r^T$ in terms of its column vectors : $H_r = [h_1, h_2, \cdots, h_n]$ and if we assume $A \approx U_r H_r$, then we have $a_i = U_r h_i$, which means that column i of H_r gives the coordinates of document i in terms of the orthogonal basis U_r. The term-document matrix is thus represented by its rank-r approximation $A_r = U_r H_r$. Therefore, in query matching, where we need to compute the inner product of

a given vector q with each document, we compute $q^T A_r = q^T U_r H_r = (U_r^T q)^T H_r$ rather than computing $q^T A$. Thus, we compute the coordinates of the query in terms of the new document basis and compute the cosines from

$$\cos\theta_i = \frac{q_r^T h_i}{||q_r||_2||h_i||_2}, \quad q_r = U_r^T q. \tag{3}$$

This means that the query matching is performed in an r-dimensional space. We refer the reader to [12] for more details.

B. Principal Component Analysis (PCA)

Principal component analysis is a technique to reduce the dimension of problems involving large covariance matrices of sets of data vectors $a_i \in R^m$, $i = 1, 2, \ldots n$. For simplicity, we will assume these vectors to have zero mean :

$$\sum_{i=1}^{n} a_i = 0. \tag{4}$$

If the original dataset does not satisfy this constraint, we can simply subtract the mean of each dimension from the original dataset. If we call A the data matrix containing these vectors as columns :

$$A = [a_1, a_2, \cdots, a_n],$$

then one can view each row of A as a particular variable observed over the n samples. The covariance matrix C of these m data is then given by the $m \times m$ symmetric positive definite matrix

$$C = AA^T,$$

but for some applications this can be a very large matrix. In the so-called *eigenfaces method* [11] this is typically the case since the dimension m is the number of pixels in the images of the different faces, and for standard images this number can be of the order of millions. Since C is a very large matrix, it is more efficient to replace it by a low-rank approximation which can be obtained from its eigendecomposition :

$$C = U\Lambda U^T \tag{5}$$

and then truncating it to its r leading eigenvalues :

$$C_r = U_r \Lambda_r U_r^T, \tag{6}$$

where U_r consists of the first r columns of U and Λ_r is the rth principal submatrix of diagonal Λ. It is well known that the best rank-r approximation given in (2) of the data matrix A with SVD (1), yields in fact the *same basis* U_r and that $\Lambda_r = \Sigma_r^2$. Moreover, this matrix C_r is a best rank-r approximation of C in the 2-norm and in the Frobenius norm sense [3]. It is also known (see [7]) that the basis U_r minimizes the mean square reconstruction error over all choices of r orthonormal basis vectors [7]. Such a set of eigenvectors that defines a new uncorrelated coordinate system for the training set matrix A is known as the set of *principal components*.

In the context of face recognition, the columns of U_r are frequently called *eigenfaces* [9]. This shows that principal component analysis is equivalent to the singular value decomposition of the underlying (centered) data matrix. Given U_r, we can "project" any vector a_i onto this r-dimensional subspace using

$$y_i = U_r^T a_i \tag{7}$$

and the reconstructed vector $\tilde{a}_i$ can be approximated as

$$\tilde{a}_i = U_r y_i. \tag{8}$$

These approximations can be used for the purpose of classification, recognition, clustering and so on. We will use them to perform (approximate) calculation of distances in the classification application (eigenfaces) described in section 5.

III. Lanczos Bidiagonalization and Krylov subspaces

In [3], Golub and Kahan describe a bidiagonalization procedure which is a variant of the Lanczos tridiagonalization algorithm and which is widely used in numerical linear algebra. Starting from an arbitrary vector b, the Golub-Kahan algorithm constructs two orthogonal bases U and V that bidiagonalize the given matrix A, by successively enlarging the corresponding Krylov subspaces at each step.

LANBI Algorithm
Inputs: $A \in R_+^{m,n}$, $b \in R_+^m$ and $0 < r < \min(m, n)$.
Outputs: orthogonal matrices U, V, diagonal matrix Σ_B.
Start with $u_1 = b/||b||$, $\beta_1 = 0$;
Step 1
for $i = 1, \cdots, r$, do
  Compute α_i, v_i such that $\alpha_i v_i = A^T u_i - \beta_i v_{i-1}$;
  Compute β_{i+1}, u_{i+1} such that $\beta_{i+1} u_{i+1} = A v_i - \alpha_i u_i$;
end
Step 2
Perform singular value decomposition on bidiagonal matrix $B_{r,r} = \text{bidiag}\{\alpha_1, \beta_2, \alpha_2, \cdots, \beta_r, \alpha_r\}$, i.e. compute U_B, Σ_B, V_B such that $B_{r,r} = U_B \Sigma_B V_B^T$;
Step 3
Compute $U = [u_1, \cdots, u_r]U_B$, $V = [v_1, \cdots, v_r]V_B$.

In the standard Lanczos bidiagonalization process, α_i and β_i are usually chosen such that $||u_i||_2 = ||v_i||_2 = 1$, but here we prefer to choose α_i and β_i positive. If we define

$$U_{r+1} = [u_1, \cdots, u_{r+1}], \quad V_r = [v_1, \cdots, v_r], \tag{9}$$

$$B_{r+1,r} = \begin{pmatrix} \alpha_1 & & \\ \beta_2 & \alpha_2 & \\ & \ddots & \alpha_r \\ & & \beta_{r+1} \end{pmatrix} \tag{10}$$

then after r steps (without breakdowns) we obtain the identities

$$A^T U_r = V_r B_{r,r}^T, \quad A V_r = U_{r+1} B_{r+1,r}.$$

It follows from this that the columns of U_r form an orthonormal basis of the Krylov subspace

$$\text{Span}[U_r] = K_r(AA^T, b) \subseteq \text{Im}([A\ b]). \tag{11}$$

Similarly, the columns of V_r then form an orthonormal basis of the Krylov subspace

$$\mathrm{Span}[V_r] = K_r(A^TA, A^Tb) \subseteq \mathrm{Im}([A^T]). \tag{12}$$

We can see that Lanczos bidiagonalization process builds up two orthogonal bases of the Krylov sequence of vectors produced by repeated application of the matrices A and A^T to a starting vector b. However, in practice, a potential complication arises: even if mathematically the recurrences guarantee orthogonal bases, rounding errors will destroy their orthogonality. Thus, one has to apply some kind of reorthogonalization strategy, which we will describe in the next section.

IV. Extended Lanczos Bidiagonalization Algorithm

In this section, we propose a new method, which we call the *extended Lanczos bidiagonalization* algorithm. Compared with the Lanczos bidiagonalization algorithm, we add a small number of iterations in order to obtain an improved low-rank approximation. Similar to section 3, we give a pseudocode of our algorithm.

E-LANBI Algorithm:
Inputs: $A \in R_+^{m,n}$, $b \in R_+^m$, $0 < r < s < \min(m, n)$.
Outputs: orthogonal matrices U, V, diagonal matrix Σ_B.
Start with $u_1 = b/||b||$, $\beta_1 = 0$;
Step 1
for $i = 1, \cdots, r+\mathbf{s}$, do
Compute α_i, v_i such that $\alpha_i v_i = A^T u_i - \beta_i v_{i-1}$;
Compute β_{i+1}, u_{i+1} such that $\beta_{i+1}u_{i+1} = Av_i - \alpha_i u_i$;
end
Step 2
Perform singular value decomposition on bidiagonal matrix $B_{r+s,r+s} = \mathrm{bidiag}\{\alpha_1, \beta_2, \alpha_2, \cdots, \beta_{r+s}, \alpha_{r+s}\}$, i.e. compute U_B, Σ_B, V_B such that $B_{r+s,r+s} = U_B \Sigma_B V_B^T$;
Step 3
Compute $U = [u_1, \cdots, u_{r+s}]U_B(:, 1:r)$
and $V = [v_1, \cdots, v_{r+s}]V_B(:, 1:r)$
(where $M(:, 1:r)$ denotes the first r columns of matrix M).

Here again, α_i and β_i are chosen to be positive. Let

$$U_{r+s+1} = [u_1, \cdots, u_{r+s+1}], \quad V_{r+s} = [v_1, \cdots, v_{r+s}], \tag{13}$$

$$B_{r+s+1,r+s} = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & & \alpha_{r+s} \\ & & & \beta_{r+s+1} \end{pmatrix} \tag{14}$$

then after $r + s$ steps we have the identities

$$A^TU_{r+s} = V_{r+s}B_{r+s,r+s}^T, \quad AV_{r+s} = U_{r+s+1}B_{r+s+1,r+s}.$$

The columns of U_{r+s} then form an orthonormal basis of the Krylov subspace

$$\mathrm{Span}[U_{r+s}] = K_{r+s}(AA^T, b) \subseteq \mathrm{Im}([A\ b]). \tag{15}$$

Similarly, the columns of V_{r+s} form an orthonormal basis of the Krylov subspace

$$\mathrm{Span}[V_{r+s}] = K_{r+s}(A^TA, A^Tb) \subseteq \mathrm{Im}([A^T]). \tag{16}$$

From the above, we know that

$$\mathrm{Span}[U] \subseteq \mathrm{Span}[U_{r+s}] = K_{r+s}(AA^T, b) \subseteq \mathrm{Im}([A\ b]),$$

and

$$\mathrm{Span}[V] \subseteq \mathrm{Span}[V_{r+s}] = K_{r+s}(A^TA, A^Tb) \subseteq \mathrm{Im}([A^T]),$$

If $s = 0$, $\mathrm{Span}[U] = \mathrm{Span}[U_{r+s}]$ and $\mathrm{Span}[V] = \mathrm{Span}[V_{r+s}]$; otherwise, the inclusions $\mathrm{Span}[U] \subset \mathrm{Span}[U_{r+s}]$ and $\mathrm{Span}[V] \subset \mathrm{Span}[V_{r+s}]$ are strict.

Similar to the Lanczos bidiagonalization process, the extended Lanczos bidiagonalization constructs two orthogonal bases of the Krylov sequence of vectors produced by repeated application of the matrices A and A^T to a starting vector b. To get a rank-r approximation of the original A, we need to apply $r + s$ Lanczos bidiagonalization steps followed by the SVD of a small matrix B_{r+s}. Hence, the extended Lanczos bidiagonalization algorithm constructs the best rank-r subspace in an extended subspace $\mathrm{Span}[U_{r+s}]$. It is therefore no surprise that we obtain a better rank-r approximation than the one obtained after r steps of the standard Lanczos bidiagonalization algorithm.

Notice that for LSI, the input matrix is typically sparse, while for PCA applied to image databases, the input matrix is typically dense. The complexity of our E-LANBI method will depend on sparsity. If we denote by α the average number of nonzero elements per column in A, then we have the following complexity table

	dense A	sparse A
Step 1	$O(mn(r+s))$	$O(\alpha n(r+s))$
Step 2	$O(r+s)^3$	$O(r+s)^3$
Step 3	$O(r(r+s)(m+n))$	$O(r(r+s)(m+n))$

which shows that E-LANBI gives a better complexity than the typical cost of $O(mn \cdot \min\{m, n\})$ for the truncated SVD.

A. Reorthogonalization

It is well known that the theoretical orthogonality of the computed Lanczos vectors u_i and v_i is quickly lost in practice. This is triggered by the convergence of one or more singular vectors [1], and many papers discuss this problem [1], [4], [5]. In this paper, since $m > n$, we combine full reorthogonalization for v_i and partial reorthogonalization for u_i. Concretely, we add the following line of pseudocode after the second line of step 1 of the algorithms in sections 3 and 4:

$$v_i = v_i - \sum_{j=1}^{i-1} \langle v_i, v_j \rangle v_j,$$

and the following line of pseudocode after the third line of step 1 of the algorithms in sections 3 and 4:

$$u_i = u_i - \sum_{j=i-l}^{i-1} \langle u_i, u_j \rangle u_j.$$

Here, l is a small integer (such as $l = 5$) and we start our reorthogonalization as soon as $i > l$. So, these two reorthogonalization steps increase the computational cost with $O(in) + O(lm)$ at every Lanczos step.

B. Convergence

In [2], [13], it was shown that the Lanczos algorithm is convergent as r increases. In the notation of this paper and using $C := AA^T$, Saad gives in [13] a bound on the angle between the jth eigenvector ϕ_j of C and the span of U_r:

$$\frac{||(I - U_r U_r^T)\phi_j||}{||U_r U_r^T \phi_j||} \leq \frac{K_j}{T_{r-j}(\gamma_j)} \frac{||(I - U_1 U_1^T)\phi_j||}{||U_1 U_1^T \phi_j||}, \quad j \leq r, \tag{17}$$

where

$$\gamma_j = 1 + 2\frac{\lambda_j - \lambda_{j+1}}{\lambda_{j+1} - \lambda_n}, \quad K_1 = 1, \quad K_j = \prod_{i=1}^{j-1} \frac{\lambda_i - \lambda_n}{\lambda_i - \lambda_j} \quad j \neq 1,$$

λ_j is the jth eigenvalue of C and $T_k(\cdot)$ is the Chebyshev polynomial of the first kind of degree k. Assuming ϕ_j has norm 1 and letting $c_j = K_j ||Q_r Q_r^T \phi_j|| \frac{||(I - Q_1 Q_1^T)\phi_j||}{||Q_1 Q_1^T \phi_j||}$, one then obtains a simple inequality of the form:

$$||(I - U_r U_r^T)\phi_j|| \leq c_j T_{r-j}(\gamma_j)^{-1}. \tag{18}$$

This inequality shows that the angle between any unit eigenvector ϕ_j and the subspace Span(U_r) decays at least as $T_{r-j}(\gamma_j)^{-1}$, which decreases exponentially as $r - j$ increases [13]. So, if we can get a better subspace Span(U_r), a smaller bound will result from it. Let us consider the subspace span($\hat{U}_r$) obtained by Algorithm E- LANBI. If $s = 0$, we know that span($\hat{U}_i$)=span(U_i) and hence the convergence is the same as for Algorithm LANBI. If $s = \min\{m-r, n-r\}$, we know that $\hat{U}_r$ spans the leading r eigenvectors $\phi_j, j = 1, \cdots, r$ of C, and hence $||(I - \hat{U}_r \hat{U}_r^T)\phi_j|| = 0$. Finally, if $0 < s < m-r$, then it follows that $0 < ||(I - \hat{U}_r \hat{U}_r^T)\phi_j|| < ||(I - U_r U_r^T)\phi_j||$.

Similarly to [2], we can also prove the convergence of the approximation vectors of E-LANBI. Assume that b is a starting vector, $s_i = \hat{U}_i \hat{U}_i^T Ab$ is the approximation vector to the vector Ab along the direction ϕ_j, then applying (18) yields

$$\begin{aligned} |\langle Ab - s_i, \phi_j \rangle| &= |\langle (I - \hat{U}_i \hat{U}_i^T) Ab, \phi_j \rangle| \\ &= |\langle (I - \hat{U}_i \hat{U}_i^T)\phi_j, Ab \rangle| \\ &\leq ||(I - \hat{U}_i \hat{U}_i^T)\phi_j|| \cdot ||Ab|| \\ &< c_j ||Ab||\ T_{i-j}(\gamma_j)^{-1}\ . \end{aligned}$$

This completes our proof, and the above inequality gives the convergence rate of the approximation vector s_i to the vector Ab along the direction ϕ_j. A similar convergence proof is easily derived for the right singular vector, whose details we skip here.

V. Experiments

In this section, we evaluate the effectiveness of Algorithm E-LANBI. All computations are done using Matlab version 7 on an Intel CPU @1.86 GHz, 1.5 Gb memory computer.

A. Latent Semantic Indexing for Information Retrieval

To determine which documents to retrieve, we compare the cosine of the angles between the query vector $q \in R^m$ and the document vectors to a fixed threshold Θ. Those cosines $\cos\theta_i$ are computed as in (3).

Performance in information retrieval depends on Θ and is assessed according to the following two measures:

$$\text{Precision} = \frac{D_r}{D_t} \quad \text{and} \quad \text{Recall} = \frac{D_r}{N_r}$$

where D_r is the number of relevant documents retrieved, D_t is the number of documents retrieved, and N_r is the number of relevant documents in database. Large (resp. small) values of threshold Θ are expected to lead to high (resp. low) precision and low (resp. high) recall.

We compare three methods : the first method is the truncated singular value decomposition (tsvd); the second method is Algorithm LANBI of section 3 (skipping Steps 2 and 3 since they are not needed for LSI); and the third method is Algorithm E-LANBI, proposed in section 4.

Three data sets were used in our experiments [14]: MED, containing 1033 documents and 30 queries; CRAN, containing 1398 documents and 255 queries; and CISI, containing 1460 documents and 35 queries. All three data sets are typical in the sense that the number of distinct terms is larger than the number of documents, i.e., $m > n$.

Example 1

In this example, we consider the MED data set, which contains 5735 terms, 1033 documents and 30 queries. We choose to consider Q2 as a query vector, for which there are 16 relevant documents.

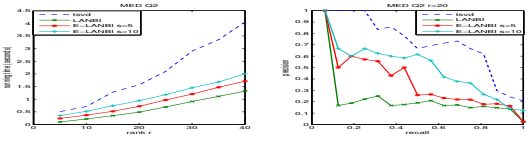

Fig 1. Left: running time vs. rank; Right: precision vs. recall (rank=20).

D_r	1	2	3	4	5	6	7	8
truncated svd	1	2	3	4	6	7	9	12
LANBI	1	12	16	18	20	36	40	42
E-LANBI s=5	1	4	5	7	9	14	28	31
E-LANBI s=10	1	3	5	6	8	10	12	13
D_r	9	10	11	12	13	14	15	16
truncated svd	13	14	15	18	21	47	62	77
LANBI	43	60	64	81	82	95	110	606
E-LANBI s=5	34	43	50	55	73	77	92	655
E-LANBI s=10	16	24	29	33	49	64	108	122

Table 1. First and sixth row: number of relevant document retrieved (D_r); From second row to fifth row and from seventh row to tenth row: total number of document retrieved (D_t) by different methods (tsvd, LANBI, E-LANBI)

The left graph in Fig. 1 shows that E-LANBI requires only a little extra work over LANBI (since it computes only a few more Lanczos steps) and that the truncated singular value decomposition needs a lot more time than the other two methods. The right graph in Fig. 1 shows that while the truncated singular value decomposition performs best in terms of precision vs. recall, E-LANBI behaves significantly better than LANBI. Numbers of documents retrieved reported in Table 1 confirm the relative ranking between the three methods.

Example 2

The second experiment shows the results on the CISI data set, containing 5544 terms, 1460 documents and 35 queries. Here Q1 is selected as a query vector, and there are 46 relevant documents for Q1.

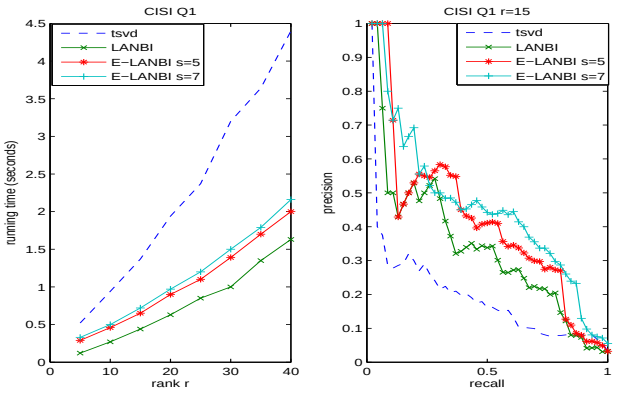

Fig. 2. Left: running time vs. rank; Right: precision vs. recall (rank=15).

The results of this experiment are different from those of Example 1. In this example, although the truncated singular value decomposition is the most expensive, it produces the worst results. Compared with LANBI, E-LANBI obtains better results, while requiring a little more time.

Example 3

For the third experiment, we show results obtained on the CRAN data set, containing 4563 terms, 1398 documents and 225 queries. We also select Q1 as a query vector, for which there are 25 relevant documents.

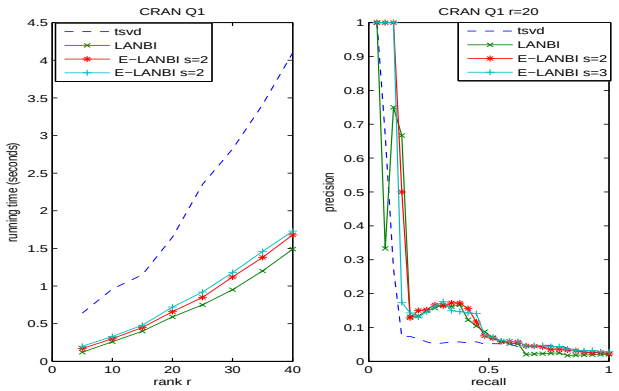

Fig. 3. Left: running time vs. rank; Right: precision vs. recall (rank=20).

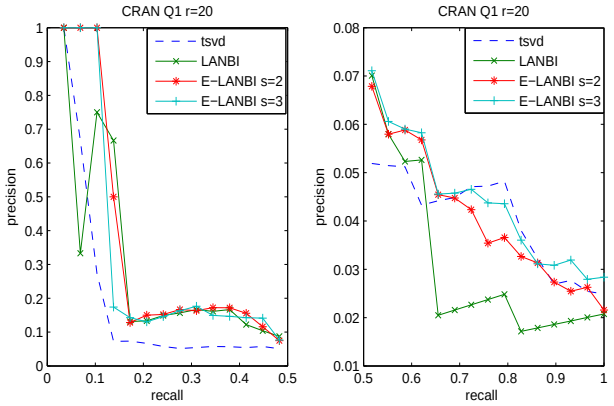

Fig. 4. precision vs. recall (rank=20).

These experiments again behave differently from the previous two. The left graph on Fig. 4 indicates that when low recall is needed, the truncated singular value decomposition gets the worst results. The right graph on Fig. 4 indicates that when high recall is needed, LANBI gets the worst results. In both cases, E-LANBI compares favorably with truncated singular value decomposition and with LANBI.

These three experiments show the advantages of E-LANBI in Latent Semantic Indexing. The combination of speed and accuracy make E-LANBI a competitive alternative to both truncated singular value decomposition and Lanczos bidiagonalization.

B. PCA for information retrieval

Two popular face databases, the Indian Face Database [15] and ORL Face [16], are used to demonstrate the effectiveness of the Extended Lanczos Bidiagonalization method proposed in section 4. In our experiments, all images in the database were manually cropped and resized to 92×112 pixels, with 256 gray levels per pixel. After the image cropping, most

of the complex background has been excluded. To apply the eigenfaces technique, each image is vectorized to a column. As suggested in section 2, LANBI and E-LANBI are applied directly to the centered data matrix $A = [a_1 - \bar{a}, \cdots, a_n - \bar{a}]$. To remove randomness, both the Lanczos bidiagonalization and the Extended Lanczos bidiagonalization were initialized with the same vector $b = (1, 1, \cdots, 1)^T$. After feature selection, the following simple classification scheme is used. For a given test vector a, the distance between a and training class C_j is defined by

$$d(a, C_j) = \frac{1}{|C_j|} \sum_{i \in C_j} ||U_{PCA}^T (a - a_i)||_2^2$$

where $|C_j|$ is the number of vectors in class C_j. This measure is simply the average squared Euclidean distance between the test vector and elements of a given class, measured in the approximate subspace. Test vector a will then be classified the class j with the smallest value of $d(a, C_j)$.

Example 4

In this example, we consider the Indian Face Database, which contains faces of 22 female and 37 male subjects. For each of these 59 subjects, the database contains 11 face images (but we only select the first 10 of these images). In our experiments, the training set consists of the first five images, and the testing set consists of the remaining 5 images of each subject.

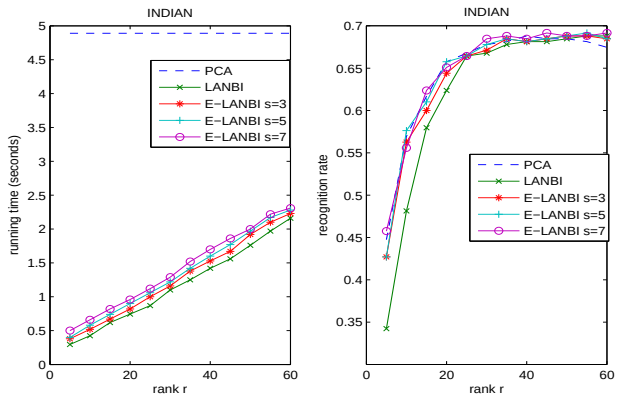

Fig. 5. Comparisons of running time and recognition rate with different rank (INDIAN): (left) running time vs. rank and (right) recognition rate vs. rank.

Example 5

In this example, we consider the ORL Face Database, composed of 40 persons and 10 face images per person. In this example, we choose the first five images per person for training, the other five for testing, i.e. estimation of the generalization performance.

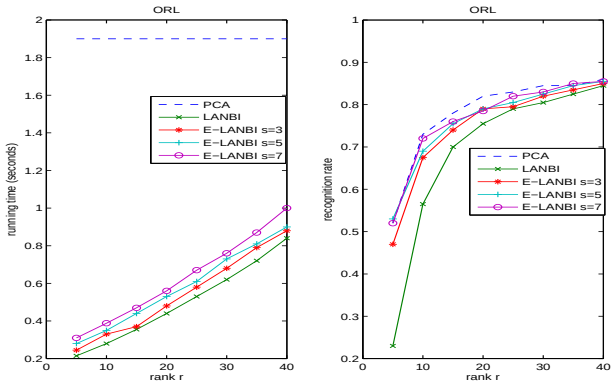

Fig. 6. Comparisons of running time and recognition rate with different rank (ORL): (left) running time vs. rank and (right) recognition rate vs. rank.

Fig. 5 and Fig. 6 show that E-LANBI obtains recognition rates comparable to PCA, while being much cheaper. Compared with LANBI, E-LANBI only requires a little extra cost, and produces higher recognition rates, in particular when the rank r is small.

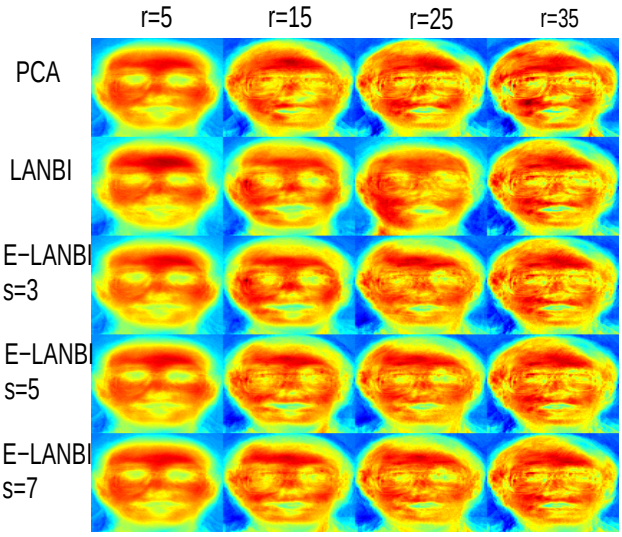

Fig. 7. Comparison of the reconstruction capability using different methods (training images).

Now, using the ORL database, we compare visually the reconstruction capability of PCA, LANBI and E-LANBI for training images. Fig. 7 shows that PCA obtains the best reconstructed images, and LANBI produces the worst reconstruction. Effectiveness of E-LANBI increases with s and reaches results comparable to PCA when $s = 7$.

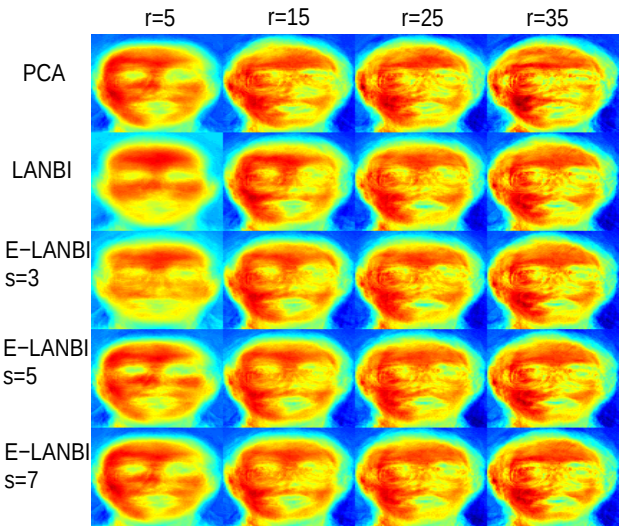

Fig. 8. Comparison of the reconstruction capability using different methods (testing images).

In Fig. 8, we also use the ORL database for testing images. Fig. 8 shows that PCA produces again the best reconstructed images. Compared with LANBI, E-LANBI obtains better reconstructed images, because it constructs a better approximating subspace.

Finally, we observe from our experiments that a small value for parameter s is often already enough to obtain good results.

VI. Conclusion

In this paper, we propose a new Lanczos-type algorithm for dimension reduction. Because it produces the best r-dimensional approximation within a larger $(r+s)$-dimensional subspace, it yields a better matrix approximation than the classical LANBI method. Compared with the truncated singular value decomposition, our method frequently obtains a low-dimensional approximation of similar quality, despite the lack of optimality. Our experiments demonstrate that E-LANBI provides a good and reasonably cheap alternative to truncated singular value decomposition.

Acknowledgments

This work was performed while the first author was visiting Université catholique de Louvain on a fellowship by China Scholarship Council. The research is also supported by National Natural Science Foundation of China (Grant No.10961010 and No. 11001232).

This work was also partly supported by the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office and supported by CNR under the Short Term Mobility Program. The scientific responsibility rests with its authors.

References

[1] B.N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, 1998.
[2] J. Chen and Y. Saad, *Lanczos Vectors versus Singular Vectors for Effective Dimension Reduction*, IEEE Transactions on Knowledge and Data Engineering, vol. 21, no 8, pp. 1091-1103, August 2009.
[3] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins Univ. Press, 1996.
[4] H.D. Simon, *Analysis of the Symmetric Lanczos Algorithm with Reorthogonalization Methods*, Linear Algebra Applications, vol. 61, pp. 101-131, 1984.
[5] R.M. Larsen, *Efficient Algorithms for Helioseismic Inversion*, PhD dissertation, Dept. of Computer Science, Univ. of Aarhus, Denmark, Oct. 1998.
[6] H. Park, M. Jeon, and J. Ben Rosen, *Lower dimensional representation of text data in vector space based information retrieval*, In Computational Information Retrieval, M. W. Berry, ed., SIAM, Philadelphia, 2001, pp. 3-23.
[7] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, New York, 1990.
[8] K.V. Ravi Kanth, D. Agrawal, A. El Abbadi, and A. Singh, *Dimensionality Reduction for Similarity Searching in Dynamic Databases*, Computer Vision and Image Understanding: CVIU, vol.75, nos. 1-2, pp. 59-72, 1999.
[9] M. Turk, A. Pentland, *Eigenfaces for recognition*, Journal of Cognitive Neuroscience 3 (1991)71-86.
[10] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman, *Indexing by Latent Semantic Analysis*, J. Soc. Information science, Vol. 41, pp. 391-407, 1990.
[11] M. Turk and A. Pentland, *Face Recognition Using Eigenfaces*, Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 586-591, 1991.
[12] M.W. Berry, S.T. Dumais, and G.W. O'Brien, *Using linear algebra for intelligent information retrieval*, SIAM Rev., 37:573-595, 1995.
[13] Y. Saad, *On the Rates of Convergence of the Lanczos and the Block-Lanczos Methods*, SIAM J. Numerical Analysis,vol. 17, no.5, pp 687-706, Oct. 1980.
[14] *ftp://ftp.cs.cornell.edu/pub/smart/.*
[15] *http://vis-www.cs.umass.edu/ vidit/IndianFaceDatabase/*, 2002.
[16] *http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html*