Incremental Linear Discriminant Analysis Using Sufficient Spanning Set Approximations

Tae-Kyun Kim[1] Shu-Fai Wong[1] Björn Stenger[2] Josef Kittler[3] Roberto Cipolla[1]

[1]Department of Engineering
University of Cambridge
Cambridge CB2 1PZ, UK

[2]Toshiba Research Europe Ltd
Computer Vision Group
Cambridge CB3 2NH, UK

[3]CVSSP
University of Surrey
Guildford GU2 7XH, UK

Abstract

This paper presents a new incremental learning solution for Linear Discriminant Analysis (LDA). We apply the concept of the sufficient spanning set *approximation in each update step, i.e. for the between-class scatter matrix, the projected data matrix as well as the total scatter matrix. The algorithm yields a more general and efficient solution to incremental LDA than previous methods. It also significantly reduces the computational complexity while providing a solution which closely agrees with the batch LDA result. The proposed algorithm has a time complexity of $O(Nd^2)$ and requires $O(Nd)$ space, where d is the reduced subspace dimension and N the data dimension. We show two applications of incremental LDA: First, the method is applied to semi-supervised learning by integrating it into an EM framework. Secondly, we apply it to the task of merging large databases which were collected during MPEG standardization for face image retrieval.*

1. Introduction

Face descriptors have been proposed as candidates for MPEG-7 standardization for face image retrieval in video streams [5, 6, 8]. An ideal face descriptor should be extracted without any prior knowledge about the current image content (person identity), i.e. it should be constructed from images of persons other than those contained in the test data. The descriptor should also be compact, even for large data sets. A challenging problem is to retrieve face images with large variations in lighting and pose when only a single query image is given. Out of all methods, Linear Discriminant Analysis (LDA)-based description methods have shown the best performance in the MPEG-7 competition [5, 6].

LDA finds the linear projections of data which best separate two or more classes under the assumption that the classes have equal covariance Gaussian structure [2]. LDA is an effective and widely employed technique for dimension reduction and feature extraction. It is often beneficial to learn the LDA basis from large training sets, which may not be available initially. This motivates techniques for incrementally updating the discriminant components when more data becomes available.

A number of incremental versions of LDA have been suggested, which can be applied to on-line learning tasks [4, 7, 9, 14]. Ye et al. [14] proposed an incremental version of LDA, which can include only a single new data point in each time step. A further limitation is the computational complexity of the method when the number of classes C is large, as the method involves an eigendecomposition of $C \times C$-dimensional scatter matrices. Pang et al. [9] introduced a scheme for updating the between-class and within-class scatter matrices. However, no incremental method is used for the subsequent LDA steps, i.e. eigenanalysis of the scatter matrices, which remains computationally expensive. Gradient-based incremental learning of a modified LDA was proposed by Hiraoka et al. [4]. Limitations of the method are that it includes only a single new data point at each time step and that it requires setting a learning rate. To circumvent the difficulty of incrementally updating the product of scatter matrices, Yan et al. [13] used a modified criterion by computing the difference of the between-class and within-class scatter matrices. However, this may lead to regularization problems of the two scatter matrices. Lin et al. [7] dealt with the online update of discriminative models for the purpose of object tracking. As their task is binary classification, the discriminative model and the update method are limited to the two-class case.

Inspiration for incremental LDA can be drawn from work on incremental Principal Component Analysis (PCA). Numerous algorithms have been developed to update the eigenbasis as more data samples arrive. However, most methods assume zero mean in updating the eigenbasis except [3, 10] where the update of the mean is handled cor-

1-4244-1180-7/07/$25.00 ©2007 IEEE

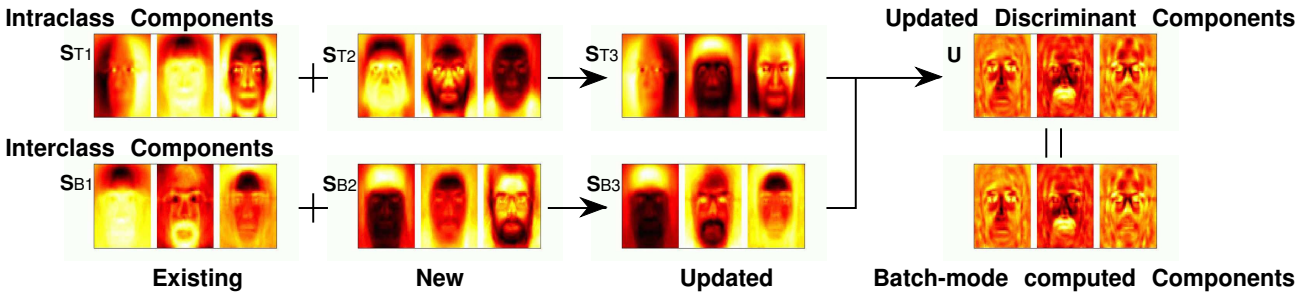


Figure 1. **On-line update of an LDA basis:** *The basis computed by the new incremental LDA algorithm (top right) closely agrees with the one computed by batch LDA (bottom right). Shown for each scatter matrix* $\mathbf{S_{T,i}}$ *and* $\mathbf{S_{B,i}}$ *are the first three principal components, which are combined by merging eigenspaces.*

rectly. The dimension of the eigenproblem can be reduced by using the *sufficient spanning set* (a reduced set of basis vectors spanning the space of most data variation). As the computation of the eigenproblem is cubic in the subspace dimension of the respective scatter matrix, this update scheme is highly efficient.

It is also worth noting the existence of efficient algorithms for kernel PCA and LDA [1, 12]. While studying the incremental learning of such non-linear models is worthwhile, when considering retrieval from large data sets, the computational cost of feature extraction of new samples is as demanding as updating the models [5, 6, 8]. Also note that the LDA method in [12] assumes a small number of classes for the update, similar to [14].

This paper proposes a new solution for incremental LDA, which is accurate as well as efficient in both time and memory. The benefit of the proposed algorithm over other LDA update algorithms [7, 14] lies in its ability to efficiently handle large data sets with many classes. This is particularly important for the face image retrieval task, where hundreds of different face classes have to be merged. An example of an LDA basis of face images is shown in Figure 1. The result obtained with the incremental algorithm closely agrees with the batch LDA solution. Note that previous studies have not shown close agreement between incremental and batch LDA solutions [12, 14].

In the proposed method an LDA criterion which is equivalent to the Fisher criterion, namely maximizing the ratio of the between-class and the total scatter matrix, is used to keep the discriminative information during the update. First the principal components of the two scatter matrices are efficiently updated and then the discriminant components are efficiently computed from these two sets of principal components. The concept of sufficient spanning sets is applied in each step, making the eigenproblem computation efficient.The algorithm is also memory efficient as it only needs to store the two sets of principal components to avoid losing discriminatory data.

The paper is structured as follows: Section 2 presents the new incremental LDA algorithm. In section 3 we show how it can be applied to semi-supervised learning within an EM-framework. Experimental results for the task of merging face databases are presented in section 4.

2. Incremental LDA

As noted by Fukunaga [2], there are equivalent variants of Fisher's criterion to find the projection matrix $\mathbf{U}$ to maximize class separability of the data set:

$$\max_{\arg \mathbf{U}} \frac{\mathbf{U}^T\mathbf{S}_B\mathbf{U}}{\mathbf{U}^T\mathbf{S}_W\mathbf{U}} = \max_{\arg \mathbf{U}} \frac{\mathbf{U}^T\mathbf{S}_T\mathbf{U}}{\mathbf{U}^T\mathbf{S}_W\mathbf{U}} = \max_{\arg \mathbf{U}} \frac{\mathbf{U}^T\mathbf{S}_B\mathbf{U}}{\mathbf{U}^T\mathbf{S}_T\mathbf{U}}, \quad (1)$$

where

$$\mathbf{S}_B = \Sigma_{i=1}^{C} n_i(\mathbf{m}_i - \boldsymbol{\mu})(\mathbf{m}_i - \boldsymbol{\mu})^T \quad (2)$$

is the between-class scatter matrix,

$$\mathbf{S}_W = \Sigma_{i=1}^{C} \Sigma_{\mathbf{x} \in C_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T \quad (3)$$

is the within-class scatter matrix,

$$\mathbf{S}_T = \Sigma_{all\ \mathbf{x}} (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T = \mathbf{S}_B + \mathbf{S}_W \quad (4)$$

the total scatter matrix, C the total number of classes, n_i the sample number of class i, $\mathbf{m}_i$ the mean of class i, and $\boldsymbol{\mu}$ the global mean. The algorithm in this paper uses the third criterion in equation 1 and separately updates the principal components as the minimal sufficient spanning sets of $\mathbf{S}_B$ and $\mathbf{S}_T$. The scatter matrix approximation with a small number of principal components (corresponding to significant eigenvalues) allows an efficient update of the discriminant components. The $\mathbf{S}_T$ matrix rather than $\mathbf{S}_W$ is used to avoid losing discriminatory data during the update. If we only kept track of the significant principal components of $\mathbf{S}_B$ and $\mathbf{S}_W$, any discriminatory information contained in the null space of $\mathbf{S}_W$ would be lost (note that any component in the null space maximizes the LDA criterion). However, as $\mathbf{S}_T = \mathbf{S}_B + \mathbf{S}_W$ and both $\mathbf{S}_B$ and $\mathbf{S}_W$ are positive

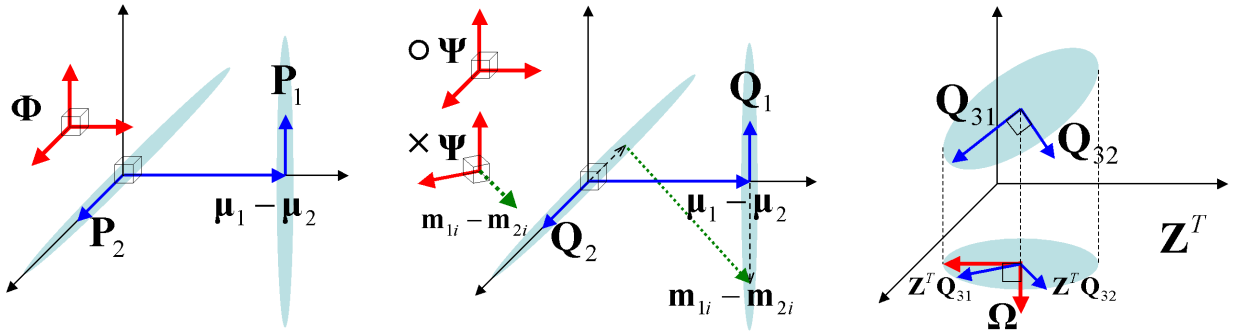


Figure 2. **Concept of sufficient spanning sets** *of the total scatter matrix* **(left)**, *the between-class scatter matrix* **(middle)** *and the projected matrix* **(right)**. *The union set of the principal components* $\mathbf{P}_1, \mathbf{P}_2$ *or* $\mathbf{Q}_1, \mathbf{Q}_2$ *of the two data sets and the mean difference vector* $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$ *can span the respective total or between-class scatter data space* **(left and middle)**. *The dimension for the component* $\mathbf{m}_{1i} - \mathbf{m}_{2i}$ *should not be removed (cross=incorrect) from the sufficient set of the between-class scatter data but retained in the set (circle=correct)* **(middle)**. *The projection and orthogonalization of the original components* $\mathbf{Q}_{31}, \mathbf{Q}_{32}$ *yields the principal components of the projected data up to rotation* **(right)**. *See the corresponding sections for detailed explanations.*

semi-definite, vectors in the null space of $\mathbf{S}_T$ are also in the null space of $\mathbf{S}_B$, and are thus being ignored in the update.

The three steps of the algorithm are: (1) Update the total scatter matrix $\mathbf{S}_T$, (2) Update the between-class scatter matrix $\mathbf{S}_B$ and (3) from these compute the discriminant components $\mathbf{U}$. These steps are explained in more detail in the following sections.

2.1. Updating the total scatter matrix

The total scatter matrix is approximated with a set of orthogonal vectors that span the subspace occupied by the data and represent it with sufficient accuracy. The eigenspace merging algorithm of Hall et al. [3] can be used with the slight modifications ([3] considered merging covariances) in order to incrementally compute the principal components of the total scatter matrix: Given two sets of data represented by eigenspace models

$$\{\boldsymbol{\mu}_i, M_i, \mathbf{P}_i, \boldsymbol{\Lambda}_i\}_{i=1,2}, \tag{5}$$

where $\boldsymbol{\mu}_i$ is the mean, M_i the number of samples, $\mathbf{P}_i$ the matrix of eigenvectors and $\boldsymbol{\Lambda}_i$ the eigenvalue matrix of the i-th data set, the combined eigenspace model $\{\boldsymbol{\mu}_3, M_3, \mathbf{P}_3, \boldsymbol{\Lambda}_3\}$ is computed. Generally only a subset of $d_{T,i}$ eigenvectors have significant eigenvalues and thus only these are stored in $\boldsymbol{\Lambda}_i$ and the corresponding eigenvectors in $\mathbf{P}_i$. We wish to compute the eigenvectors and eigenvalues of the new eigenspace model that satisfy $\mathbf{S}_{T,3} \simeq \mathbf{P}_3 \boldsymbol{\Lambda}_3 \mathbf{P}_3^T$. The eigenvector matrix $\mathbf{P}_3$ can be represented by a sufficient spanning set (see below for discussion) and a rotation matrix as

$$\mathbf{P}_3 = \boldsymbol{\Phi}\mathbf{R} = h([\mathbf{P}_1, \mathbf{P}_2, \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2])\mathbf{R}, \tag{6}$$

where $\boldsymbol{\Phi}$ is the orthonormal column matrix spanning the combined scatter matrix, $\mathbf{R}$ is a rotation matrix, and h is an orthonormalization function (e.g. QR decomposition).

Using the sufficient spanning set, the eigenproblem is converted into a smaller eigenproblem as

$$\mathbf{S}_{T,3} = \mathbf{P}_3 \boldsymbol{\Lambda}_3 \mathbf{P}_3^T \;\Rightarrow\; \boldsymbol{\Phi}^T \mathbf{S}_{T,3} \boldsymbol{\Phi} = \mathbf{R} \boldsymbol{\Lambda}_3 \mathbf{R}^T. \tag{7}$$

By computing the eigendecomposition on the r.h.s. one obtains $\boldsymbol{\Lambda}_3$ and $\mathbf{R}$ as the respective eigenvalue and eigenvector matrices. After removing nonsignificant components in $\mathbf{R}$ according to the eigenvalues in $\boldsymbol{\Lambda}_3$, the minimal sufficient spanning set is obtained as $\mathbf{P}_3 = \boldsymbol{\Phi}\mathbf{R}$. Note the matrix $\boldsymbol{\Phi}^{\mathrm{T}} \mathbf{S}_{T,3} \boldsymbol{\Phi}$ has the reduced size $d_{T,1} + d_{T,2} + 1$ and the dimension of the approximated combined total scatter matrix is $d_{T,3} \leq d_{T,1} + d_{T,2} + 1$, where $d_{T,1}, d_{T,2}$ are the number of the eigenvectors in $\mathbf{P}_1$ and $\mathbf{P}_2$ respectively. Thus the eigenanalysis here only takes $O((d_{T,1} + d_{T,2} + 1)^3)$ computations, whereas the eigenanalysis in batch mode (on the l.h.s. of (7)) requires $O(\min(N, M_3)^3)$, where N is the dimension of the input data [1]. See Section 2.4 for a more detailed discussion about the time and space complexity.

Discussion. We conclude this section by giving more insight into the sufficient spanning set concept. Generally, given a data matrix $\mathbf{A}$, the sufficient spanning set $\boldsymbol{\Phi}$ can be defined as any set of vectors s.t.

$$\mathbf{B} = \boldsymbol{\Phi}^T \mathbf{A}, \quad \mathbf{A}' = \boldsymbol{\Phi}\mathbf{B} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T \mathbf{A} \simeq \mathbf{A}. \tag{8}$$

That is, the reconstruction $\mathbf{A}'$ of the data matrix by the sufficient spanning set should approximate the original data matrix. Let $\mathbf{A} \simeq \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^T$ where $\mathbf{P}, \boldsymbol{\Lambda}$ are the eigenvector and

[1] When $N \gg M$, the batch mode complexity can effectively be $O(M^3)$ as follows: $\mathbf{S}_T = \mathbf{Y}\mathbf{Y}^T$, where $\mathbf{Y} = [..., \mathbf{x}_i - \boldsymbol{\mu}, ...]$. SVD of $\mathbf{Y}$ s.t. $\mathbf{Y} = \mathbf{U}\Sigma\mathbf{V}^T$ yields the eigenspace model of $\mathbf{S}_T$ by $\mathbf{U}$ and $\Sigma\Sigma^T$ as the eigenvector and eigenvalue matrix respectively. $\mathbf{Y}^T\mathbf{Y} = \mathbf{V}\Sigma^T\Sigma\mathbf{V}^T$ as $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. That is, by SVD of the low-dimensional matrix $\mathbf{Y}^T\mathbf{Y}$, the eigenvector matrix is efficiently obtained as $\mathbf{Y}\mathbf{V}\Sigma^{-1}$ and the eigenvalue matrix as $\Sigma^T\Sigma$. This greatly reduces the complexity when obtaining the eigenspace model of a small new data set in batch mode prior to combining.

eigenvalue matrix corresponding to most energy. Then, $\mathbf{PR}$ where $\mathbf{R}$ is an arbitrary rotation matrix can be a sufficient spanning set:

$$\mathbf{A}' = \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{A} \simeq \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T \simeq \mathbf{A} \tag{9}$$

as $\mathbf{R}\mathbf{R}^T = \mathbf{P}^T\mathbf{P} = \mathbf{I}$. This also applies to the sufficient spanning set in equation (6).

As visualized on the left of Figure 2, the union of the two principal components and the mean difference vector can span all data points of the combined set in the three-dimensional space. The principal components of the combined set are found by rotating this sufficient spanning set.

Note that this use of the sufficient spanning set is only possible in the case of merging generative models where the scatter matrix of the union set is represented as the sum of the scatter matrices of the two sets explicitly as

$$\mathbf{S}_{T,3} = \mathbf{S}_{T,1} + \mathbf{S}_{T,2} + M_1M_2/M_3 \cdot (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T, \tag{10}$$

where $\{\mathbf{S}_{T,i}\}_{i=1,2}$ are the scatter matrices of the first two sets. The method can therefore not be used to directly merge the discriminant components of LDA models.

2.2. Updating the between-class scatter matrix

In the update of the total scatter matrix, a set of new vectors are added to a set of existing vectors. The between-class scatter matrix, however, is the scatter matrix of the class mean vectors, see equation (12). Not only is a set of new class means added, but the existing class means also change when new samples belong to existing classes. Interestingly, the proposed update can be interpreted as simultaneous incremental (adding new data points) and decremental (removing existing data points) learning (see below).

The principal components of the combined between-class scatter matrix can be efficiently computed from the two sets of between-class data, represented by

$$\{\boldsymbol{\mu}_i, M_i, \mathbf{Q}_i, \mathbf{\Delta}_i, n_{ij}, \boldsymbol{\alpha}_{ij} \,|j = 1, ..., C_i\}_{i=1,2}, \tag{11}$$

where $\boldsymbol{\mu}_i$ is the mean vector of the data set i, M_i is the total number of samples in each set, $\mathbf{Q}_i$ are the eigenvector matrices, $\mathbf{\Delta}_i$ are the eigenvalue matrices of $\mathbf{S}_{B,i}$, n_{ij} the number of samples in class j of set i, and C_i the number of classes in set i. The $\boldsymbol{\alpha}_{ij}$ are the coefficient vectors of the j-th class mean vector $\mathbf{m}_{ij}$ of set i with respect to the subspace spanned by $\mathbf{Q}_i$, i.e. $\mathbf{m}_{ij} \simeq \boldsymbol{\mu}_i + \mathbf{Q}_i\boldsymbol{\alpha}_{ij}$. The task is to compute the eigenmodel $\{\boldsymbol{\mu}_3, M_3, \mathbf{Q}_3, \mathbf{\Delta}_3, n_{3j}, \boldsymbol{\alpha}_{3j} \,|j = 1, ..., C_3\}$ for the combined between-class scatter matrix. To obtain the sufficient spanning set for efficient eigencomputation, the combined between-class scatter matrix is represented by the sum of the between-class scatter matrices of the first two data sets, similar to (10). The between-class scatter matrix $\mathbf{S}_{B,i}$ can be written as

$$\mathbf{S}_{B,i} = \sum_{j=1}^{C_i} n_{ij}(\mathbf{m}_{ij} - \boldsymbol{\mu}_i)(\mathbf{m}_{ij} - \boldsymbol{\mu}_i)^T \tag{12}$$

$$= \sum_{j=1}^{C_i} n_{ij}\mathbf{m}_{ij}\mathbf{m}_{ij}^T - M_i\boldsymbol{\mu}_i\boldsymbol{\mu}_i^T. \tag{13}$$

The combined between-class scatter matrix can further be written w.r.t. the original between-class scatter matrices and an auxiliary matrix $\mathbf{A}$ as

$$\mathbf{S}_{B,3} = \mathbf{S}_{B,1} + \mathbf{S}_{B,2} + \mathbf{A} + M_1M_2/M_3 \cdot (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T, \tag{14}$$

where

$$\mathbf{A} = \sum_{k \in s} \frac{-n_{1k}n_{2k}}{n_{1k} + n_{2k}}(\mathbf{m}_{2k} - \mathbf{m}_{1k})(\mathbf{m}_{2k} - \mathbf{m}_{1k})^T. \tag{15}$$

The set $s = \{k|k = 1, ..., c\}$ contains the indices of the common classes of both data sets. The matrix $\mathbf{A}$ needs to be computed only when the two sets have common classes, otherwise it is simply set to zero. If we assume that each between-class scatter matrix is represented by the first few eigenvectors such that $\mathbf{S}_{B,1} \simeq \mathbf{Q}_1\mathbf{\Delta}_1\mathbf{Q}_1^T$, $\mathbf{S}_{B,2} \simeq \mathbf{Q}_2\mathbf{\Delta}_2\mathbf{Q}_2^T$, the sufficient spanning set for the combined between-class scatter matrix can be similarly set as

$$\mathbf{\Psi} = h([\mathbf{Q}_1, \mathbf{Q}_2, \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2]), \tag{16}$$

where the function h is the orthonormalization function used in section 2.1. Note that the matrix $\mathbf{A}$ is negative semi-definite and does not add any more dimensions to $\mathbf{\Psi}$. As illustrated in the middle of Figure 2, the sufficient spanning set can be a union set of the two eigen-components and the mean difference vector. The negative semi-definite matrix $\mathbf{A}$ can conceptually be seen as the scatter matrix of the components to be removed from the combined data. When ignoring the scale factors, the decremental elements are $\mathbf{m}_{2i} - \mathbf{m}_{1i}$. This decreases the data variance along the direction of $\mathbf{m}_{2i} - \mathbf{m}_{1i}$ but the respective dimension should not be removed from the sufficient spanning set. The resulting variance reduction along this direction is taken into account when removing eigencomponents with nonsignificant eigenvalues in the subsequent eigenanalysis.

Let $d_{B,i}$ and N be the dimension of $\mathbf{Q}_i$ and input vectors, respectively. Whereas the eigenanalysis of the combined between-class scatter in batch mode [2] requires $O(\min(N, C_3)^3)$, the proposed incremental scheme requires only $O((d_{B,1} + d_{B,2} + 1)^3)$ computation for solving

$$\mathbf{S}_{B,3} = \mathbf{\Psi}\mathbf{R}\mathbf{\Delta}_3\mathbf{R}^T\mathbf{\Psi}^T \;\Rightarrow\; \mathbf{\Psi}^T\mathbf{S}_{B,3}\mathbf{\Psi} = \mathbf{R}\mathbf{\Delta}_3\mathbf{R}^T, \tag{17}$$

[2] The batch solution of the between-class scatter matrix can be computed using the low-dimensional matrix similarly to the total scatter matrix when $N \gg C$. Note $\mathbf{S}_{B,i} = \mathbf{Y}\mathbf{Y}^T$, $\mathbf{Y} = [..., \sqrt{n_{ij}}(\mathbf{m}_{ij} - \boldsymbol{\mu}_i), ...]$.

Algorithm 1. Incremental LDA (ILDA)

Input: The total and between-class eigenmodels of an existing data set, $\{\mathbf{P}_1, ...\}$, $\{\mathbf{Q}_1, ...\}$ and a new data set
Output: Updated LDA components $\mathbf{U}$

1. Compute $\{\mathbf{P}_2, ...\}$, $\{\mathbf{Q}_2, ...\}$ from the new data set in batch mode.
2. Update the total scatter matrix for $\{\mathbf{P}_3, ...\}$:
   Compute $\mathbf{S}_{T,3}$ by (10) and $\{\mathbf{S}_{T,i}\}_{i=1,2} \simeq \mathbf{P}_i \Lambda_i \mathbf{P}_i^T$.
   Set Φ by (6) and compute the principal components $\mathbf{R}$ of $\Phi^T \mathbf{S}_{T,3} \Phi$. $\mathbf{P}_3 = \Phi \mathbf{R}$.
3. Update the between-class scatter for $\{\mathbf{Q}_3, ...\}$:
   Obtain $\mathbf{S}_{B,3}$ from (14), $\{\mathbf{S}_{B,i}\}_{i=1,2} \simeq \mathbf{Q}_i \Delta_i \mathbf{Q}_i^T$ and $\mathbf{m}_{ij} \simeq \boldsymbol{\mu}_i + \mathbf{Q}_i \boldsymbol{\alpha}_{ij}$.
   Set Ψ by (16) and eigendecompose $\Psi^T \mathbf{S}_{B,3} \Psi$ for the eigenvector matrix $\mathbf{R}$. $\mathbf{Q}_3 = \Psi \mathbf{R}$.
4. Update the discriminant components:
   Compute $\mathbf{Z} = \mathbf{P}_3 \Lambda_3^{-1/2}$ and $\Omega = h([\mathbf{Z}^T \mathbf{Q}_3])$.
   Eigendecompose $\Omega^T \mathbf{Z}^T \mathbf{Q}_3 \Delta_3 \mathbf{Q}_3^T \mathbf{Z} \Omega$ for the eigenvector matrix $\mathbf{R}$. $\mathbf{U} = \mathbf{Z} \Omega \mathbf{R}$.

Table 1. **Pseudocode of Incremental LDA.**

where $\mathbf{R}$ is a rotation matrix. Finally, the eigenvectors of the combined between-class scatter matrix, which are memorized for the next update, are obtained by $\mathbf{Q}_3 = \boldsymbol{\Psi} \mathbf{R}$ after the components having zero eigenvalues in $\mathbf{R}$ are removed, i.e. $d_{B,3} \leq d_{B,1} + d_{B,2} + 1$. All remaining parameters of the updated model are obtained as follows: $\boldsymbol{\mu}_3$ is the global mean updated in Section 2.1, $M_3 = M_1 + M_2$, $n_{3j} = n_{1j} + n_{2j}, \boldsymbol{\alpha}_{3j} = \mathbf{Q}_3^T (\mathbf{m}_{3j} - \boldsymbol{\mu}_3)$, where $\mathbf{m}_{3j} = (n_{1j} \mathbf{m}_{1j} + n_{2j} \mathbf{m}_{2j}) / n_{3j}$.

2.3. Updating discriminant components

After updating the principal components of the total scatter matrix and the between-class scatter matrix, the discriminative components are found using the updated total data $\{\boldsymbol{\mu}_3, M_3, \mathbf{P}_3, \boldsymbol{\Lambda}_3\}$ and the updated between-class data $\{\boldsymbol{\mu}_3, M_3, \mathbf{Q}_3, \boldsymbol{\Delta}_3, n_{3j}, \boldsymbol{\alpha}_{3j} \; | j = 1, ..., C_3\}$ using the new sufficient spanning set. Let $\mathbf{Z} = \mathbf{P}_3 \boldsymbol{\Lambda}_3^{-1/2}$, then $\mathbf{Z}^T \mathbf{S}_{T,3} \mathbf{Z} = \mathbf{I}$. As the denominator of the LDA criterion is the identity matrix in the projected space, the optimization problem is to find the components that maximize $\mathbf{Z}^T \mathbf{S}_{B,3} \mathbf{Z}$ s.t. $\mathbf{W}^T \mathbf{Z}^T \mathbf{S}_{B,3} \mathbf{Z} \mathbf{W} = \boldsymbol{\Lambda}$ and the final LDA components are obtained by $\mathbf{U} = \mathbf{Z} \mathbf{W}$. This eigenproblem of the projected data can be solved using the sufficient spanning set defined by

$$\boldsymbol{\Omega} = h([\mathbf{Z}^{\mathrm{T}} \mathbf{Q}_3]). \tag{18}$$

See the right of Figure 2. The original components are projected and orthogonalized to construct the sufficient spanning set. The principal components of the projected data can be found by rotating the sufficient spanning set. By this sufficient spanning set, the eigenvalue problem changes into a smaller dimensional eigenvalue problem by

$$\mathbf{Z}^{\mathrm{T}} \mathbf{S}_{B,3} \mathbf{Z} = \boldsymbol{\Omega} \mathbf{R} \boldsymbol{\Lambda} \mathbf{R}^T \boldsymbol{\Omega}^T \;\Rightarrow\; \boldsymbol{\Omega}^T \mathbf{Z}^T \mathbf{S}_{B,3} \mathbf{Z} \boldsymbol{\Omega} = \mathbf{R} \boldsymbol{\Lambda} \mathbf{R}^T. \tag{19}$$

The final discriminant component is given as

$$\mathbf{Z} \mathbf{W} = \mathbf{Z} \boldsymbol{\Omega} \mathbf{R}. \tag{20}$$

This eigenproblem takes $O(d^3)$ time, where d is the dimension of $\boldsymbol{\Omega}$, which is equivalent to $d_{B,3}$, the dimension of $\mathbf{Q}_3$. Note that in LDA, $d_{T,3}$, the dimension of $\mathbf{P}_3$ is usually larger than $d_{B,3}$ and therefore the use of the sufficient spanning set further reduces the time complexity of the eigenanalysis: $O(d_{T,3}^3) \rightarrow O(d_{B,3}^3)$. The pseudocode of the complete incremental LDA algorithm is given in Table 1.

	Batch LDA	Inc LDA
time	$O(NM_3^2$	$O(d_{T,1}^3 + d_{B,1}^3$
	$+min(N, M_3)^3)$	$+Nd_{T,3}d_{B,3})$
space	$O(NM_3 + NC_3)$	$O(Nd_{T,3} + Nd_{B,3})$

Table 2. **Comparison of time and space complexity:** *The savings of incremental LDA are significant as usually $M_3 \gg d_{T,3} \geq d_{B,3}$. N is the data dimension and M_3, C_3 are total number of data points and classes, respectively, $d_{T,i}, d_{B,i}$ are the dimensions of the total and between-class scatter subspaces.*

2.4. Time and space complexity

So far we have mainly considered the computational complexity of solving the eigenproblem for merging two data sets. Table 1 provides a comparison of the batch and the proposed incremental LDA in total time complexity (considering the necessary matrix products e.g. those in (7)) and space complexity, when the additional set is relatively small compared to the existing set, i.e. $M_2 \ll M_1$.

The computational saving of the incremental solution compared to the batch version is large as normally $M_3 \gg d_{T,3} \geq d_{B,3}$. Both time and space complexity of the proposed incremental LDA are independent of the size of the total sample set and the total number of classes. The important observation from the face data base merging experiments (see Table 3) is that the intermediate dimensions $d_{T,3}$ and $d_{B,3}$ do not increase significantly when new data is successively added.

3. Semi-supervised incremental learning

This section deals with the LDA update when the class labels of new samples are not given. Unlike incremental learning of generative models [3, 10], discriminative models such as LDA, require the class labels of additional samples for the model update. The proposed incremental LDA can be incorporated into a semi-supervised learning algorithm so that the LDA update can be computed

efficiently without the class labels of the additional data set being known. For an overview of semi-supervised learning, including an explanation of the role of unlabeled data, see [15]. Although graph-based methods have been widely adopted for semi-supervised learning [15], the classic mixture model has long been recognized as a natural approach to modeling unlabeled data. A mixture model makes predictions for arbitrary new test points and typically has a relatively small number of parameters. Additionally mixture models are compatible with the proposed incremental LDA model assuming multiple Gaussian-distributed classes [2]. Here, classic EM-type learning is employed to generate the probabilistic labels of the new samples. Running EM in the updated LDA subspaces allows for more accurate estimation of the class labels. We iterate the E-step and M-step with all data vectors projected into the LDA subspaces (similar to [11]), which are incrementally updated in an intermediate step. The class posterior probabilities of the new samples are set to the probabilistic labels.

Incremental LDA with EM. The proposed EM algorithm employs a generative model with the most recent LDA transformation $\mathbf{U}$ by

$$P(\mathbf{U}^T\mathbf{x}|\Theta) = \sum_{k=1}^{C} P(\mathbf{U}^T\mathbf{x}|C_k;\Theta_k)P(C_k|\Theta_k), \quad (21)$$

where class $C_k, k = 1, ..., C$ is parameterized by $\Theta_k, k = 1, ..., C$, and $\mathbf{x}$ is a sample of the initial labeled set $\mathcal{L}$ and the new unlabeled set $\mathcal{U}$. The E-step and M-step are iterated to estimate the MAP model over the projected samples $\mathbf{U}^T\mathbf{x}$ of the labeled and unlabeled sets. The proposed incremental LDA is performed every few iterations on the data sets $\{\mathbf{x}_j, y_j|\mathbf{x}_j \in \mathcal{L}\}$ and $\{\mathbf{x}_j, y'_{jk}|\mathbf{x}_j \in \mathcal{U}, k = 1, ..., C\}$, where y_j is the class label and y'_{jk} is the probabilistic class label given as the class posterior probability

$$y'_{jk} = P(C_k|\mathbf{U}^T\mathbf{x}_j). \quad (22)$$

We set

$$\mathbf{m}_{2i} = \frac{\sum_j \mathbf{x}_j y'_{ji}}{\sum_j y'_{ji}}, \quad n_{2i} = \sum_{j=1}^{M_2} y'_{ji}. \quad (23)$$

for the update of the between-class scatter matrix. All other steps for incremental LDA are identical to the description in Section 2 as they are independent of class label information.

Discussion. Using a common covariance matrix Θ_k for all class models rather than C class covariance matrices is more consistent with the assumption of LDA [2] and can additionally save space and computation time during the M-step. The common covariance matrix can be conveniently updated by $\mathbf{U}^T(\mathbf{S}_{T,3} - \mathbf{S}_{B,3})\mathbf{U}/M_3$, where $\mathbf{S}_{T,3}, \mathbf{S}_{B,3}$ are the combined total and between-class scatter matrices, which are kept track of in the incremental LDA as the associated first few eigenvector and eigenvalue matrices. The other parameters of Θ are also obtained from the output of the incremental LDA algorithm.

So far it is assumed that the new data points are in one of the existing classes, but this is not necessarily the case. Samples with new class labels can be screened out so that the LDA update is not biased to those samples by

$$y'_{jk} = P(C_k|\mathbf{U}^T\mathbf{x}_j) \cdot P(\mathbf{C}|\mathbf{U}^T\mathbf{x}_j), \quad (24)$$

where $P(\mathbf{C}|\mathbf{U}^T\mathbf{x}_j)$ denotes a probability of a hyper class. We can set this probability as being close to zero for samples with new class labels.

4. Experimental results

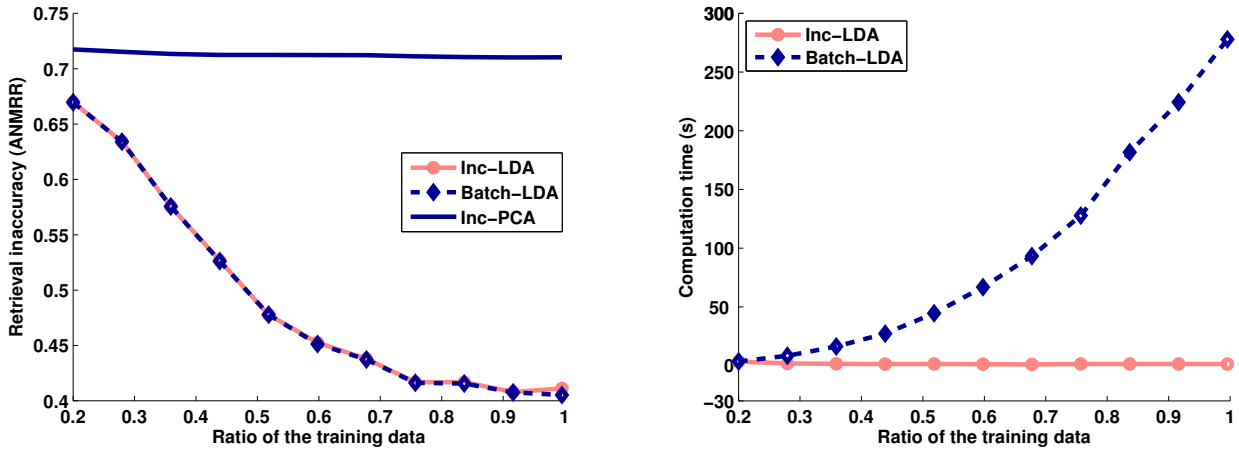
The algorithm was applied the task of face image retrieval from a large database. All experiments were performed on a 3 GHz Pentium 4 PC with 1GB RAM.

4.1. Database and protocol

In the experiments we followed the protocols of evaluating face descriptors for MPEG-7 standardization [6]. Many MPEG-7 proposals, including the winning method, have adopted LDA features as their descriptors [5, 6]. A descriptor vector is extracted without knowledge of the test subject's identity, i.e. its statistical basis should be generated from images of subjects other than those in the test set. As it is necessary to learn the LDA basis from a very large training set, which may not be available initially, the proposed algorithm can be used to successively update the LDA basis as more data becomes available. An experimental face database was obtained consisting of the version 1 MPEG data set (635 persons, 5 images per person), the *Altkom* database (80 persons, 15 images per person), the *XM2VTS* database (295 persons, 5 images per person), and the *BANCA* database (52 persons, 10 images per person). The version 1 MPEG data set itself consists of several public face sets (e.g. *AR, ORL*). All 6370 images in the database were normalized to 46×56 pixels using manually labeled eye positions. The images for the experiments were strictly divided into training and test sets. All basis vectors were extracted from the training set. All test images were used as query images to retrieve other images of the corresponding persons (called ground truth images) in the test data set. As a measure of retrieval performance, we used the average normalized modified retrieval rate (ANMRR) [8]. The ANMRR is 0 when images of the same person (ground truth labeled) are ranked on top, and it is 1 when all images are ranked outside the first m images ($m = 2N_G$, where N_G is the number of ground truth images in the test data set).

LDA update	M_3 [# images]	C_3 [# classes]	$d_{T,3}$ [dim($S_{t,3}$)]	$d_{B,3}$ [dim($S_{b,3}$)]
1[first] – 10[final]	465–2315	93–463	158–147	85–85

Table 3. **Efficient LDA update:** *Despite the large increase in the number of images and classes, the number of required principal components, $d_{T,3}$ and $d_{B,3}$, remains small during the update process implying that computation time remains low.*



(a) (b)

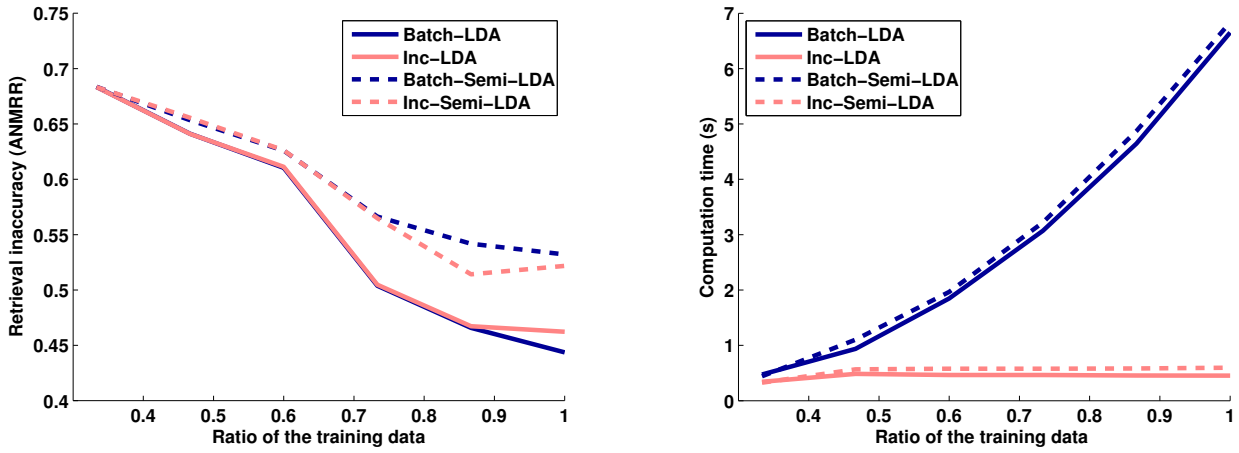
Figure 3. **Database merging experiments for the *MPEG+XM2VTS* data set:** *The solution of incremental LDA closely agrees to the batch solution while requiring much lower computation time. (a) Retrieval inaccuracy, ANMRR is 0 when all ground truth images are ranked on top, and 1 when none of the ground truth images are ranked among the first m images. (b) Computational cost.*

4.2. Results

The training set was further partitioned into an initial training set and several new sets which are added successively for re-training. We used the combined set of MPEG and *XM2VTS* database (the total number of classes is 930) for the experiment where the new sets contain the images of new classes. We also performed the experiments for the *Altkom* and *BANCA* database separately where the additional sets contain new images of the existing classes of the initial training set. The proposed incremental LDA yielded nearly the same solution as batch LDA for both scenarios. The basis images of LDA of the incremental and batch versions are compared in Figure 1. The accuracy of the incremental solution can be seen in Figure 3 (a). Incremental LDA yields essentially the same accuracy as batch LDA, provided enough components are stored of the total and between-class scatter matrices. This is an accuracy vs. speed trade-off: using less components is clearly beneficial in terms of computational cost. The subspace dimensions for incremental learning were chosen from the eigenvalue plots by setting a fixed threshold on the variance of each component (similar results were obtained by choosing the first components that contain a specified fraction of the total variance)[3]. Table 3 shows the number of components selected during the experiment using the *MPEG+XM2VTS* data set. Even if the total number of images or classes increases, the number of components does not increase significantly, thus saving time and space (See section 2.4). The computational costs of the batch and the incremental version are compared in Figure 3 (b). Whereas the computational cost of the batch version increases significantly as data is successively added, the cost of the incremental solution remains low. Note that the cost of incremental LDA is comparable to that of incremental PCA while giving a much higher retrieval accuracy as shown in Figure 3 (a). Incremental PCA did not significantly increase the retrieval accuracy.

Figure 4 shows the result of comparing the proposed semi-supervised incremental LDA solution with the LDA solution when the correct class labels are provided. For this experiment the projection of all data points into the LDA subspace was performed once with the most recent LDA components before the EM iteration, and the incremental LDA with the probabilistic labels was carried out after EM converged, typically after ten iterations. The solution boosted the retrieval accuracy even without the class labels and its incremental solution yielding the same solution as the batch version. The cost of semi-supervised LDA is slightly higher than that of incremental LDA, but still far

[3]Note that accuracy of LDA is dependent on dimensionality of intermediate components (total scatter matrix) and final components (discriminant components). These dimensions of ILDA were set to be the same as those of batch LDA.



(a) (b)

Figure 4. **Performance of semi-supervised incremental LDA:** *Semi-supervised incremental LDA decreases the error rate without the class labels of new training data being available, while being as time-efficient as incremental LDA with given labels. (a) Retrieval inaccuracy (ANMRR), (b) computational costs for the* Altkom *database. Similar results were obtained for the* BANCA *database.*

lower than any batch-mode computation.

5. Conclusions

The proposed incremental LDA solution allows highly efficient learning to adapt to new data sets. A solution closely agreeing with the batch LDA result can be obtained with far lower complexity in both time and space. The incremental LDA algorithm can also be incorporated into a classic semi-supervised learning framework and applied to many other problems in which LDA-like discriminant components are required. Directions for future research are the extension to the non-linear case, adaptive learning with a time-decaying function and using temporal information for more efficient semi-supervised learning.

References

[1] T.-J. Chin and D. Suter. Incremental Kernel PCA for Efficient Non-linear Feature Extraction. In *BMVC*, 2006.

[2] K. Fukunaga. *Introduction to statistical pattern recognition*. Academic Press, Boston, 1990.

[3] P. Hall, D. Marshall, and R. Martin. Merging and splitting eigenspace models. *IEEE Trans. on PAMI*, 22(9):1042–1049, 2000.

[4] K. Hiraoka, K. Hidai, M. Hamahira, H. Mizoguchi, T. Mishima, and S. Yoshizawa. Successive learning of linear discriminant analysis: Sanger-type algorithm. In *ICPR*, 2000.

[5] T. Kamei, A. Yamada, T. Kim, H. Kim, W. Hwang, S. Kee. Advanced face descriptor using Fourier and inten-sity LDA features. ISO/IEC JTC1/SC29/WG11 M8998, Oct 2002.

[6] T.-K. Kim, H. Kim, W. Hwang, and J. Kittler. Component-based LDA face description for image retrieval and MPEG-7 standardisation. *Image and Vision Computing*, 23:631–642, 2005.

[7] R.-S. Lin, D. Ross, J. Lim, and M.-H. Yang. Adaptive discriminative generative model and its applications. In *NIPS*, 2005.

[8] B. S. Manjunath, P. P. Salembier, and T. Sikora. *Introduction to MPEG-7: Multimedia Content Description Interface*. Wiley, New York, 2002.

[9] S. Pang, S. Ozawa, and N. Kasabov. Incremental linear discriminant analysis for classification of data streams. *IEEE Trans. on System, Man and Cybernetics*, pages 905–914, 2005.

[10] D. Skocaj and A. Leonardis. Weighted and robust incremental method for subspace learning. In *ICCV*, 2003.

[11] Y. Wu and S. Huang. View-independent recognition of hand postures. In *CVPR*, pages 2088–2094, 2000.

[12] X. Tao, J. Ye, Q. Li, R. Janardan, and V. Cherkassky. Efficient Kernel Discriminant Analysis via QR Decomposition. In *NIPS*, 2004.

[13] J. Yan, B. Zhang, S. Yan, Q. Yang, and H. Li. IMMC: Incremental maximum margin criterion. In *Int'l Conf. Knowledge Discovery and Data Mining*, 2004.

[14] J. Ye, Q. Li, H. Xiong, H. Park, V. Janardan, and V. Kumar. IDR/QR: An incremental dimension reduction algorithm via QR decomposition. *IEEE Trans. on Knowledge and Data Engineering*, 17(9):1208–1222, 2005.

[15] X. Zhu. *Semi-Supervised learning literature survey*. Computer Sciences TR 1530, University of Wisconsin-Madison, 2006.