Noname manuscript No.
(will be inserted by the editor)

Incremental Linear Discriminant Analysis Using Sufficient Spanning Sets and Its Applications

Tae-Kyun Kim · Björn Stenger · Josef Kittler · Roberto Cipolla

Received: date / Accepted: date

Abstract This paper presents an incremental learning solution for Linear Discriminant Analysis (LDA) and its applications to object recognition problems. We apply the *sufficient spanning set* approximation in three steps i.e. update for the total scatter matrix, between-class scatter matrix and the projected data matrix, which leads an online solution which closely agrees with the batch solution in accuracy while significantly reducing the computational complexity. The algorithm yields an efficient solution to incremental LDA even when the number of classes as well as the set size is large. The incremental LDA method has been also shown useful for semi-supervised online learning. Label propagation is done by integrating the incremental LDA into an EM framework. The method has been demonstrated in the task of merging large datasets which were collected during MPEG standardization for face image retrieval, face authentication using the BANCA dataset, and object categorisation using the Caltech101 dataset.

Keywords Linear Discriminant Analysis · LDA · Incremental Learning · Online Learning · Label Propagation · Semi-supervised Learning · Face Image Retrieval · Object Recognition · Object Categorisation · Face Authentication

T-K. Kim
Sidney Sussex College, University of Cambridge, Cambridge, CB2 3HU, UK.
Tel.: +44-1223-765149
Fax: +44-1223-332662
E-mail: tkk22@cam.ac.uk

B. Stenger
Toshiba Research Europe Ltd, 208 Cambridge Science Park, Cambridge CB4 0GZ, UK.
Tel.: +44-1223-436900
Fax: +44-1223-436909
E-mail: bjorn.stenger@crl.toshiba.co.uk

J. Kittler
Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, GU2 7XH, UK.
Tel.: +44-1483-686030
Fax: +44-1483-686031
E-mail: J.Kittler@surrey.ac.uk

R. Cipolla
Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK.
Tel.: +44-1223-332849
Fax: +44-1223-332662
E-mail: cipolla@eng.cam.ac.uk

1 Introduction

Linear Discriminant Analysis (LDA) finds the linear projections of data that best separate two or more classes under the assumption that the classes have equal covariance Gaussian structures [6]. LDA is an effective and widely employed technique for dimension reduction and feature extraction. LDA has been successfully applied to face recognition problems by combining it with: raw intensity or filtered images, Gabor wavelet representations, and Local Binary Patterns, which is a popular histogram representation in many areas including pedestrian detection [1], originally texture analysis as well as face recognition [2]. Usefulness of dimension reduction methods such as Principal Component Analysis (PCA) and LDA has been also proven in object categorisation and action recognition problems e.g. [24–27]. Various representations of objet images, appearance or shape, e.g. Bag of words (BoW) histograms and Scale Invariant Feature Transform (SIFT) descriptors, have been followed by a dimension reduction method.

The obtained low-dimensional vectors are then combined with classifiers or generative models. PCA as a unsupervised learning method has been a more often choice but supervised learning methods like LDA could be more useful when class information is available as e.g. in [26,28].

Incremental (also called online) learning has become an important topic in cognitive computer vision. Environments are continually changing and, practically, the assumptions are that a complete set of learning samples is not given in advance. An efficient update method is greatly needed to accumulate the new information so that the system's future accuracy is enhanced. The system needs to learn without explicitly accessing old data and the data model should be maintained compact when learning by new learning instances. 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. Compared to online classifier (Support Vector Machine) learning, LDA is a technique more about representation, further being able to be combined with classifiers or any models as a meta-algorithm.

A number of incremental versions of LDA have been suggested, which can be applied to on-line learning tasks. Ye et al. [19] proposed an incremental version of LDA, which includes a single new data point in each time step. A major limitation is the computational complexity of the method when the number of classes C is large, as the method involves an eigendecomposition of $C \times C$-sized scatter matrices. The incremental LDA solution of Uray et al. [20] first performs incremental PCA then updates LDA bases. The method similarly takes a single new data point as input and suffers when C is large. Pang et al. [14] 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. [8]. Limitations of the method are that it requires setting a learning rate. The learning complexity over a new data set is not analytically provided. To circumvent the difficulty of incrementally updating the product of scatter matrices in the LDA criterion, Yan et al. [18] used a modified criterion by computing the difference of the between-class and within-class scatter matrices and proposed an alternating solution with convergence proof. However, this leads to regularization problems of the two scatter matrices. Lin et al. [11] dealt with online update of discriminative models for the purpose of object tracking. Their task is binary classification, the discriminative model and the update method are limited to the two-class case. The prior-arts aforementioned can be partitioned into two categories: methods directly updating discriminant components as in [8,18] and methods computing discriminant components based on updated PCA components in [20,11,19]. A closed-form solution to directly update the discriminative components is hard to be obtained. The methods in [8,18] used a modified differentiable LDA criterion which is not equivalent to that of the original LDA and resorted to an iterative optimisation technique i.e. gradient-descent. In the PCA-based methods [20,11,19], no alternation is required but a single data point is taken as input thus requiring too frequent updates. The methods assume a small number of classes ignoring an efficient update of the scatter matrix in the numerator of the LDA criterion, i.e. the between-class scatter matrix.

Inspiration for incremental LDA can be drawn from work on incremental PCA. Numerous algorithms have been developed to update eigenbases as more data samples arrive. However, most methods assume zero mean in updating the eigenbases except [7,15] where the update of the mean is handled correctly. In the methods [7, 15], the size of the matrix to be eigendecomposed is 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 size of the respective scatter matrix, this update scheme is highly efficient. See Section 2.

It is also worth noting the existence of efficient algorithms for kernel PCA and LDA [4,17]. While studying the incremental learning of such non-linear models is worthwhile, when considering recognition from large data sets, the computational cost of feature extraction of new samples is as demanding as updating the models [9,10,12]. Also note that the LDA method in [17] assumes a small number of classes for the update.

This paper proposes a three-step solution for incremental LDA, which is accurate as well as efficient in both time and memory. Based on an earlier version [23], this work includes a more thorough analysis of time and space complexity, discussions and new experiments. Matlab code and data sets used in the experiments have been made publicly available [42]. 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 better 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 computed from these two sets of principal components. The concept of suffi-

cient spanning sets is applied in each step, making the eigenproblems computationally efficient. The algorithm is also memory efficient as it only needs to store the two sets of principal components. The proposed algorithm does not require the iterations in [8,18]. The benefit of the proposed algorithm over the methods [11,19,20] lies in its ability to efficiently handle large data sets with *many classes*. This is particularly important when the number of classes increases in an online setting and thus a large number of object classes have to be merged. It also handles a set of new data points (as well as a single data point), thus not requiring frequent updates. The result obtained with the proposed incremental algorithm closely agrees with the batch LDA solution. Note that previous studies have shown a gap in performance between incremental and batch LDA solutions [17,19]. We also propose an incremental LDA method with label propagation. The proposed method incorporated into an EM-framework enables online learning without the class labels of new train data being known. The usefulness of the proposed solution is shown for object categorisation as well as face recognition tasks by various image representations.

The paper is structured as follows: Section 2 briefly reviews the incremental PCA method of Hall et al. [7], which is a base element of our method. Section 3 presents the new incremental LDA algorithm. In Section 4 we show how it can be applied to semi-supervised incremental learning by the EM-based label propagation. We show the experimental results for the task of merging face databases for face image retrieval, face authentication and general object categorisation in Section 5.

2 Incremental PCA

For a set of M data vectors, $\mathbf{x} \in \mathbb{R}^N$, the covariance matrix is

$$\mathbf{C} = 1/M \Sigma_{all\ \mathbf{x}} (\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T \tag{1}$$

where $\boldsymbol{\mu}$ is the data mean. PCA decomposes the covariance matrix s.t. $\mathbf{C} \simeq \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^T$ where $\mathbf{P}, \boldsymbol{\Lambda}$ are the matrices containing the first eigenvectors and eigenvalues. Given two sets of data represented by eigenspace models $\{\boldsymbol{\mu}_i, M_i, \mathbf{P}_i, \boldsymbol{\Lambda}_i\}_{i=1,2}$, the algorithm of Hall et al. [7] efficiently computes the eigenspace model of the combined data $\{\boldsymbol{\mu}_3, M_3, \mathbf{P}_3.\boldsymbol{\Lambda}_3\}$. The combined mean is obtained as $\boldsymbol{\mu}_3 = (M_1\boldsymbol{\mu}_1 + M_2\boldsymbol{\mu}_2)/M_3$ and the combined covariance matrix is

$$\mathbf{C}_3 = \frac{M_1}{M_3}\mathbf{C}_1 + \frac{M_2}{M_3}\mathbf{C}_2 + \frac{M_1 M_2}{M_3^2} \cdot (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T, \tag{2}$$

where $\{\mathbf{C}_i\}_{i=1,2}$ are the covariance matrices of the first two sets and $M_3 = M_1 + M_2$. The eigenvector matrix $\mathbf{P}_3$ can be represented as

$$\mathbf{P}_3 = \boldsymbol{\Phi}\mathbf{R} = h([\mathbf{P}_1, \mathbf{P}_2, \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2])\mathbf{R}, \tag{3}$$

where $\boldsymbol{\Phi}$ is the orthonormal column matrix spanning the combined covariance matrix i.e. *the sufficient spanning set*, $\mathbf{R}$ is a rotation matrix, and h is an orthonormalization function (e.g. QR decomposition) followed by removal of zero vectors. Using this representation, the eigenproblem is converted into a smaller eigenproblem as

$$\mathbf{C}_3 \simeq \mathbf{P}_3\boldsymbol{\Lambda}_3\mathbf{P}_3^T \ \Rightarrow\ \boldsymbol{\Phi}^T\mathbf{C}_3\boldsymbol{\Phi} \simeq \mathbf{R}\boldsymbol{\Lambda}_3\mathbf{R}^T. \tag{4}$$

By computing the eigendecomposition on the r.h.s. $\boldsymbol{\Lambda}_3$ and $\mathbf{R}$ are obtained as the respective eigenvalue and eigenvector matrices. The eigenvector matrix to seek is given as $\mathbf{P}_3 = \boldsymbol{\Phi}\mathbf{R}$. Note the eigenanalysis on the r.h.s only takes $O((d_1+d_2+1)^3)$ computations (d_1, d_2 are the number of the eigenvectors stored in $\mathbf{P}_1$ and $\mathbf{P}_2$), whereas the eigenanalysis in a batch mode on the l.h.s. of (4)) requires $O(\min(N, M_3)^3)$.

3 Incremental LDA

As noted by Fukunaga [6], there are equivalent variants of Fisher's criterion to find the projection matrix $\mathbf{U}$ to maximize class separability of the data set:

$$\begin{aligned}\arg\max_{\mathbf{U}} \frac{|\mathbf{U}^T\mathbf{S}_B\mathbf{U}|}{|\mathbf{U}^T\mathbf{S}_W\mathbf{U}|} &= \arg\max_{\mathbf{U}} \frac{|\mathbf{U}^T\mathbf{S}_T\mathbf{U}|}{|\mathbf{U}^T\mathbf{S}_W\mathbf{U}|} \\ &= \arg\max_{\mathbf{U}} \frac{|\mathbf{U}^T\mathbf{S}_B\mathbf{U}|}{|\mathbf{U}^T\mathbf{S}_T\mathbf{U}|},\end{aligned} \tag{5}$$

where

$$\mathbf{S}_B = \Sigma_{i=1}^{C} n_i(\mathbf{m}_i - \boldsymbol{\mu})(\mathbf{m}_i - \boldsymbol{\mu})^T \tag{6}$$

is the between-class scatter matrix,

$$\mathbf{S}_W = \Sigma_{i=1}^{C} \Sigma_{\mathbf{x}\in C_i}(\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T \tag{7}$$

is the within-class scatter matrix,

$$\mathbf{S}_T = \Sigma_{all\ \mathbf{x}}(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T = \mathbf{S}_B + \mathbf{S}_W \tag{8}$$

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 LDA projection matrix $\mathbf{U}$ can be obtained as the eigenvector matrix of $\mathbf{S}_W^{-1}\mathbf{S}_B$ and one might think of directly merging the two projection matrices $\mathbf{U}_1, \mathbf{U}_2$ similarly to $\mathbf{P}_1, \mathbf{P}_2$ in the previous section. This, however, is not right since the matrix $\mathbf{S}_W^{-1}\mathbf{S}_B$ of the combined data is not given as sum of the same of the first two sets (see below for

more discussions). The algorithm in this paper uses the third criterion in (5) 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 better keep discriminatory data during the update. E.g. 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 semi-definite, it follows that

$$\mathbf{u}^{\mathrm{T}}\mathbf{S}_T\mathbf{u} = 0 \;\Rightarrow\; \mathbf{u}^{\mathrm{T}}\mathbf{S}_W\mathbf{u} = 0 \;\wedge\; \mathbf{u}^{\mathrm{T}}\mathbf{S}_B\mathbf{u} = 0\,, \tag{9}$$

which means vectors in the null space of $\mathbf{S}_T$ are also in the null space of $\mathbf{S}_B$, and the eigenvectors of $\mathbf{S}_B$ that have zero eigenvalues do not contribute to classification. Theoretically, such components at the present time can still reappear to be contributive by have nonzero eigenvalues during updates, but from the experiments showing the very close accuracy of our method to that of the batch LDA, it seems that the components of the least significant eigenvalues of $\mathbf{S}_T$ have an ignorable chance to be important in the LDA update.

The three main steps of the proposed incremental LDA are:

1. Given two sets of data, each represented by an eigenspace model, the principal components of the total scatter matrix $\mathbf{S}_T$ of the union set is computed by merging the eigenspace models.
2. Similarly the principal components of the combined between-class scatter matrix $\mathbf{S}_B$ is updated by merging the respective two eigenspace models.
3. The final step is to compute the discriminant components $\mathbf{U}$ using the updated principal components of the previous steps.

The steps of the algorithm are explained in details in Section 3.1, 3.2, 3.3.

Discussion. We conclude this section by giving more insight into the sufficient spanning set concept. Generally, given a data matrix $\mathbf{A}$ of $\mathbb{R}^{N\times M}$ where N, M are the dimension and number of input data vectors, the sufficient spanning set $\mathbf{\Phi}$ can be defined as any set of vectors s.t.

$$\mathbf{B} = \mathbf{\Phi}^T\mathbf{A}, \quad \mathbf{A}' = \mathbf{\Phi}\mathbf{B} = \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{A} \simeq \mathbf{A}. \tag{10}$$

That is, the reconstruction $\mathbf{A}'$ of the data matrix by the sufficient spanning set should approximate the original data matrix. Let $\mathbf{A} \simeq \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T$ where $\mathbf{P}, \mathbf{\Lambda}$ are the eigenvector and eigenvalue matrix corresponding to most energy. Then, $\mathbf{PR}$ where $\mathbf{R}$ is an arbitrary rotation matrix can be a sufficient spanning set:

$$\mathbf{A}' = \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{A} \simeq \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T \simeq \mathbf{A} \tag{11}$$

as $\mathbf{R}\mathbf{R}^T = \mathbf{P}^T\mathbf{P} = \mathbf{I}$.

When combining two sets of data as in Section 2, the union of the two matrices of principal components and the mean difference vector in (3) can span all data points of the combined set. The case in the three dimensional space is visualized on the left of Figure 1. The principal components of the combined set are then found by rotating this sufficient spanning set according to data variance. Note that the efficient sufficient spanning set can only be obtained in the case of merging covariance matrices or scatter matrices (not products of scatter matrices) as the matrix of the union set to eigendecompose is represented as the sum of the matrices of the two sets explicitly as (2). The matrix $(\mathbf{S}_W^{-1}\mathbf{S}_B)_3$ can not be similarly decomposed into $\{(\mathbf{S}_W^{-1}\mathbf{S}_B)_i\}_{i=1,2}$ and thus a small-sized sufficient spanning set can not be obtained.

3.1 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. [7], which merged covariance matrices, is slightly modified in order to incrementally compute the principal components of the total scatter matrix: Given two sets of data represented by eigenspace models

$$\{\boldsymbol{\mu}_i, M_i, \mathbf{P}_i, \mathbf{\Lambda}_i\}_{i=1,2}, \tag{12}$$

where $\boldsymbol{\mu}_i$ is the mean, M_i the number of samples, $\mathbf{P}_i$ the matrix of eigenvectors and $\mathbf{\Lambda}_i$ the eigenvalue matrix of the i-th data set, the combined eigenspace model $\{\boldsymbol{\mu}_3, M_3, \mathbf{P}_3, \mathbf{\Lambda}_3\}$ is computed. Generally only a subset of $d_{T,i}$ eigenvectors have significant eigenvalues and thus only these are stored in $\mathbf{\Lambda}_i$ and the corresponding eigenvectors in $\mathbf{P}_i$.

We wish to compute the eigenvectors and eigenvalues of the new eigenspace model that satisfy $\mathbf{S}_{T,3} \simeq \mathbf{P}_3\mathbf{\Lambda}_3\mathbf{P}_3^T$. Since

$$\mathbf{S}_{T,3} = \mathbf{S}_{T,1} + \mathbf{S}_{T,2} + M_1M_2/M_3 \cdot (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T, \tag{13}$$

where $\{\mathbf{S}_{T,i}\}_{i=1,2}$ are the total scatter matrices of the first two sets, the eigenvector matrix $\mathbf{P}_3$ can be represented by a sufficient spanning set $\mathbf{\Phi}$ and a rotation

matrix $\mathbf{R}$ as

$$\mathbf{P}_3 = \mathbf{\Phi R} = h([\mathbf{P}_1, \mathbf{P}_2, \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2])\mathbf{R}, \tag{14}$$

where h is an orthonormalization function followed by removal of zero vectors. See Figure 1. Using the sufficient spanning set, a smaller eigenproblem is obtained as

$$\mathbf{S}_{T,3} = \mathbf{P}_3 \mathbf{\Lambda}_3 \mathbf{P}_3^T \;\Rightarrow\; \mathbf{\Phi}^T \mathbf{S}_{T,3} \mathbf{\Phi} = \mathbf{R} \mathbf{\Lambda}_3 \mathbf{R}^T. \tag{15}$$

By computing the eigendecomposition on the r.h.s. one obtains $\mathbf{\Lambda}_3$ and $\mathbf{R}$ as the respective eigenvalue and eigenvector matrices. After removing nonsignificant components in $\mathbf{R}$ according to the eigenvalues in $\mathbf{\Lambda}_3$, the minimal sufficient spanning set is obtained as $\mathbf{P}_3 = \mathbf{\Phi R}$. Note the matrix $\mathbf{\Phi}^{\mathrm{T}} \mathbf{S}_{T,3} \mathbf{\Phi}$ has the size $d_{T,1} + d_{T,2} + 1$ and the size 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 (15)) requires $O(\min(N, M_3)^3)$, where N is the dimension of the input data. When a small new set is merged into an existing data set, for which we have already computed the eigenspace model, solving the eigenproblem for merging takes the major computational load of the entire update process [1]. See Section 3.4 for the total time and space complexity.

3.2 Updating the between-class scatter matrix

The between-class scatter matrix is incrementally updated as the other ingredient for computing the discriminant components. 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 (17). 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.

[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.

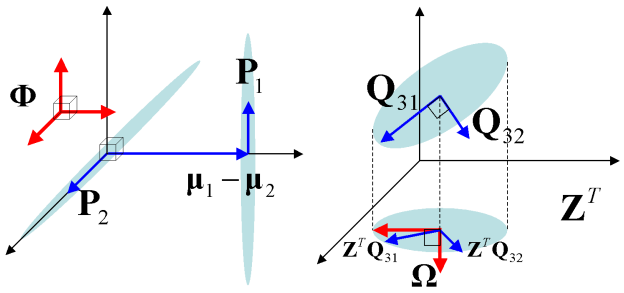


Fig. 1 Concept of sufficient spanning sets *of the total scatter matrix (similarly the between-class scatter matrix)* **(left)** *and the projected matrix* **(right)**. *The union set of the principal components* $\mathbf{P}_1, \mathbf{P}_2$ *or* $\mathbf{Q}_1, \mathbf{Q}_2$ *of the two data sets and the mean difference vector* $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$ *can span the respective total or between-class scatter data space* **(left)**. *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.*

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

$$\{\boldsymbol{\mu}_i, M_i, \mathbf{Q}_i, \mathbf{\Delta}_i, n_{ij}, \boldsymbol{\alpha}_{ij} \;|j = 1, ..., C_i\}_{i=1,2}, \tag{16}$$

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 eigen-computation, 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 (13). The between-class scatter matrix $\mathbf{S}_{B,i}$ can be written as

$$\mathbf{S}_{B,i} = \sum_{j=1}^{C_i} n_{ij}(\mathbf{m}_{ij} - \boldsymbol{\mu}_i)(\mathbf{m}_{ij} - \boldsymbol{\mu}_i)^T \tag{17}$$

$$= \sum_{j=1}^{C_i} n_{ij}\mathbf{m}_{ij}\mathbf{m}_{ij}^T - M_i\boldsymbol{\mu}_i\boldsymbol{\mu}_i^T. \tag{18}$$

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

$$\mathbf{S}_{B,3} = \mathbf{S}_{B,1} + \mathbf{S}_{B,2} + \mathbf{A} + M_1 M_2 / M_3 \cdot (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T, \tag{19}$$

where

$$\mathbf{A} = \sum_{k \in s} \frac{-n_{1k} n_{2k}}{n_{1k} + n_{2k}} (\mathbf{m}_{2k} - \mathbf{m}_{1k})(\mathbf{m}_{2k} - \mathbf{m}_{1k})^T. \tag{20}$$

The set $s = \{k|k = 1, ..., c\}$ contains the indices of the common classes in the two 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

$$\mathbf{\Psi} = h([\mathbf{Q}_1, \mathbf{Q}_2, \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2]), \tag{21}$$

where the function h is the orthonormalization function used in section 3.1. Note that the matrix $\mathbf{A}$ is negative semi-definite and does not add dimensions to $\mathbf{\Psi}$. Thus, 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 size of $\mathbf{Q}_i$ and the dimension of 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

$$\mathbf{S}_{B,3} = \mathbf{\Psi}\mathbf{R}\mathbf{\Delta}_3\mathbf{R}^T\mathbf{\Psi}^T \;\Rightarrow\; \mathbf{\Psi}^T\mathbf{S}_{B,3}\mathbf{\Psi} = \mathbf{R}\mathbf{\Delta}_3\mathbf{R}^T, \tag{22}$$

where $\mathbf{R}$ is a rotation matrix. Note that $d_{B,1} + d_{B,2} + 1$ is the size of $\mathbf{\Psi}^T\mathbf{S}_{B,3}\mathbf{\Psi}$. Finally, the eigenvectors of the combined between-class scatter matrix, which are memorized for the next update, are obtained by $\mathbf{Q}_3 = \mathbf{\Psi}\mathbf{R}$ 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 3.1, $M_3 = M_1 + M_2$, $n_{3j} = n_{1j} + n_{2j}$, $\boldsymbol{\alpha}_{3j} = \mathbf{Q}_3^T(\mathbf{m}_{3j} - \boldsymbol{\mu}_3)$, where $\mathbf{m}_{3j} = (n_{1j}\mathbf{m}_{1j} + n_{2j}\mathbf{m}_{2j})/n_{3j}$.

3.3 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 $\{\boldsymbol{\mu}_3, M_3, \mathbf{P}_3, \mathbf{\Lambda}_3\}$ and the updated between-class data $\{\boldsymbol{\mu}_3, M_3, \mathbf{Q}_3, \mathbf{\Delta}_3, n_{3j}, \boldsymbol{\alpha}_{3j} \;|j = 1, ..., C_3\}$ using the new sufficient spanning set. Let $\mathbf{Z} = \mathbf{P}_3\mathbf{\Lambda}_3^{-1/2}$, 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. $\mathbf{W}^T\mathbf{Z}^T\mathbf{S}_{B,3}\mathbf{Z}\mathbf{W} = \mathbf{\Lambda}$ 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

$$\mathbf{\Omega} = h([\mathbf{Z}^{\mathrm{T}}\mathbf{Q}_3]). \tag{23}$$

See the right of Figure 1. The original components are projected and orthogonalised 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

$$\mathbf{Z}^{\mathrm{T}}\mathbf{S}_{B,3}\mathbf{Z} = \mathbf{\Omega}\mathbf{R}\mathbf{\Lambda}\mathbf{R}^T\mathbf{\Omega}^T \;\Rightarrow\; \mathbf{\Omega}^T\mathbf{Z}^T\mathbf{S}_{B,3}\mathbf{Z}\mathbf{\Omega} = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^T. \tag{24}$$

The final discriminant component is given as

$$\mathbf{Z}\mathbf{W} = \mathbf{Z}\mathbf{\Omega}\mathbf{R}. \tag{25}$$

This eigenproblem takes $O(d^3)$ time, where d is the number of components of $\mathbf{\Omega}$, which is equivalent to $d_{B,3}$, the size of $\mathbf{Q}_3$. Note that in LDA, $d_{T,3}$, the size of $\mathbf{P}_3$ is usually larger than $d_{B,3}$ and therefore the use of the

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 set of new data vectors
Output: Updated LDA components $\mathbf{U}$

1. Compute $\{\mathbf{P}_2, ...\}, \{\mathbf{Q}_2, ...\}$ from the new data set in batch mode (see footnotes 1,2).
2. Update the total scatter matrix for $\{\mathbf{P}_3, ...\}$:
   Compute $\mathbf{S}_{T,3}$ by (13) and $\{\mathbf{S}_{T,i}\}_{i=1,2} \simeq \mathbf{P}_i\mathit{\Lambda}_i\mathbf{P}_i^T$.
   Set $\mathit{\Phi}$ by (14) and compute the principal components $\mathbf{R}$ of $\mathit{\Phi}^T\mathbf{S}_{T,3}\mathit{\Phi}$. $\mathbf{P}_3 = \mathit{\Phi}\mathbf{R}$.
3. Update the between-class scatter for $\{\mathbf{Q}_3, ...\}$:
   Obtain $\mathbf{S}_{B,3}$ from (19), $\{\mathbf{S}_{B,i}\}_{i=1,2} \simeq \mathbf{Q}_i\mathit{\Delta}_i\mathbf{Q}_i^T$ and $\mathbf{m}_{ij} \simeq \boldsymbol{\mu}_i + \mathbf{Q}_i\boldsymbol{\alpha}_{ij}$.
   Set $\mathit{\Psi}$ by (21) and eigendecompose $\mathit{\Psi}^T\mathbf{S}_{B,3}\mathit{\Psi}$ for the eigenvector matrix $\mathbf{R}$. $\mathbf{Q}_3 = \mathit{\Psi}\mathbf{R}$.
4. Update the discriminant components:
   Compute $\mathbf{Z} = \mathbf{P}_3\mathit{\Lambda}_3^{-1/2}$ and $\mathit{\Omega} = h([\mathbf{Z}^T\mathbf{Q}_3])$.
   Eigendecompose $\mathit{\Omega}^T\mathbf{Z}^T\mathbf{Q}_3\mathit{\Delta}_3\mathbf{Q}_3^T\mathbf{Z}\mathit{\Omega}$ for the eigenvector matrix $\mathbf{R}$. $\mathbf{U} = \mathbf{Z}\mathit{\Omega}\mathbf{R}$.

Table 1 Pseudocode of Incremental LDA.

[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 $\mathbf{S}_{B,i} = \mathbf{Y}\mathbf{Y}^T$, $\mathbf{Y} = [..., \sqrt{n_{ij}}(\mathbf{m}_{ij} - \boldsymbol{\mu}_i), ...]$.

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.

3.4 Time and space complexity

So far we have mainly considered the computational complexity of solving the eigenproblem for merging two data sets represented as the eigenspace models. This section provides a more detailed analysis of the total update complexity. Batch LDA has a space complexity of $O(NM_3 + NC_3)$ and a time complexity of $O(NM_3^2 + \min(N, M_3)^3)$.

In the proposed incremental LDA, for the update of the principal components of the total scatter matrix, we only need to keep track of the data associated with $\{\boldsymbol{\mu}_3, M_3, \mathbf{P}_3, \boldsymbol{\Lambda}_3\}$ taking $O(Nd_{T,3})$ space. The total process can be partitioned into the merging and solving the eigenproblem of the new data set. Note that the computation cost of the orthonormalization in (14) and the necessary matrix products in (15) can be efficiently reduced by exploiting the orthogonality of the eigenvectors [7]. This cost is bounded by $O(Nd_{T,1}d_{T,2})$ and the eigendecomposition takes $O(d_{T,3}^3)$. The eigenanalysis of the new data set is computed in $O(NM_2^2 + \min(N, M_2)^3)$.

Similarly only $\{\boldsymbol{\mu}_3, M_3, \mathbf{Q}_3, \boldsymbol{\Delta}_3, n_{3j}, \boldsymbol{\alpha}_{3j} \,|j = 1, ..., C_3\}$ is required to be stored for the update of the between-class scatter matrix, taking $O(Nd_{B,3})$. The computational complexity of this update is $O(Nd_{B,1}d_{B,2}+d_{B,3}^3)$, and $O(NC_2^2 + \min(N, C_2)^3)$ for the merging and the eigenanalysis of the new set respectively.

The final LDA components are computed only from the two sets of data above in time $O(Nd_{T,3}d_{B,3})$.

Table 2 provides a comparison of the batch and the proposed incremental LDA in total time complexity 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 experiments (see Table 3) is that the dimensions $d_{T,3}$ and $d_{B,3}$ do not increase significantly when new data is successively added.

	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.*

4 Semi-supervised incremental learning by label propagation

Unlike incremental learning of generative models [7, 15], 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 [21]. Although graph-based methods have been widely adopted for semi-supervised learning [21], a classic mixture model has long been recognized as a natural approach to modeling unlabeled data. The 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 method under the assumption that classes are Gaussian-distributed [6]. Here, standard EM-type learning is employed to generate the *probabilistic labels* of the new samples. Running EM in the updated LDA subspaces allows for 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 [16]), 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 projection $\mathbf{U}$ by

$$P(\mathbf{U}^T\mathbf{x}|\Theta) = \sum_{k=1}^{C} P(\mathbf{U}^T\mathbf{x}|C_k;\Theta_k)P(C_k|\Theta_k), \qquad (26)$$

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

$$y'_{jk} = P(C_k | \mathbf{U}^T \mathbf{x}_j). \tag{27}$$

We set

$$\mathbf{m}_{2i} = \frac{\sum_j \mathbf{x}_j y'_{ji}}{\sum_j y'_{ji}}, \quad n_{2i} = \sum_{j=1}^{M_2} y'_{ji}. \tag{28}$$

for the update of the between-class scatter matrix. All other steps for incremental LDA are identical to the description in Section 3 as they are independent of class label information.

Discussion. Using a common covariance matrix for all class models $\Theta_k, k = 1, ..., C$ rather than C covariance matrices is more consistent with the assumption of LDA [6] 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_k 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

$$y'_{jk} = P(C_k | \mathbf{U}^T \mathbf{x}_j) \cdot P(\{C_k\}_{k=1,...,C} | \mathbf{U}^T \mathbf{x}_j), \tag{29}$$

where $P(\{C_k\}_{k=1,...,C} | \mathbf{U}^T \mathbf{x}_j)$ denotes a probability of a hyper class. We can set this probability as being close to zero for samples with new class labels.

The projection to the LDA subspace helps the data vectors be class-wise Gaussian distributed, but it is yet limited to the linear transformation. Any non-linear models or spectral analysis [37] may be further considered in future.

5 Experimental results

All experiments were performed on a 3 GHz Pentium 4 PC with 1GB RAM. The Matlab code for the proposed incremental LDA method and the data set used are publicly available [42].

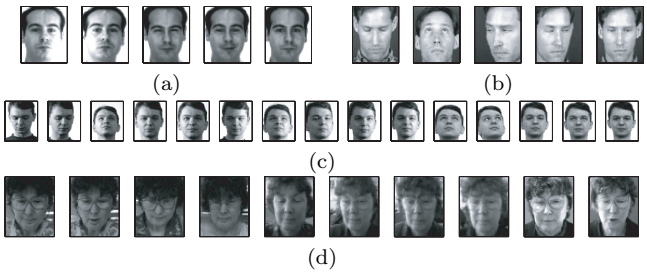


Fig. 2 Face image data set: *(a) The version 1 MPEG dataset. (b) XM2VTS dataset. (c) Altkom dataset. (d) BANCA dataset.*

5.1 Face image retrieval

The algorithm is applied to the task of face image retrieval from a large database.

5.1.1 Database and protocol

In the experiments we followed the protocols of evaluating face descriptors for MPEG-7 standardization [10]. Many MPEG-7 proposals, including the winning method, have adopted LDA features as their descriptors [9,10]. 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. Each image in the test database is used as a query image to retrieve other images of the same subject. 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. See Figure 2. 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) [12]. 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.*

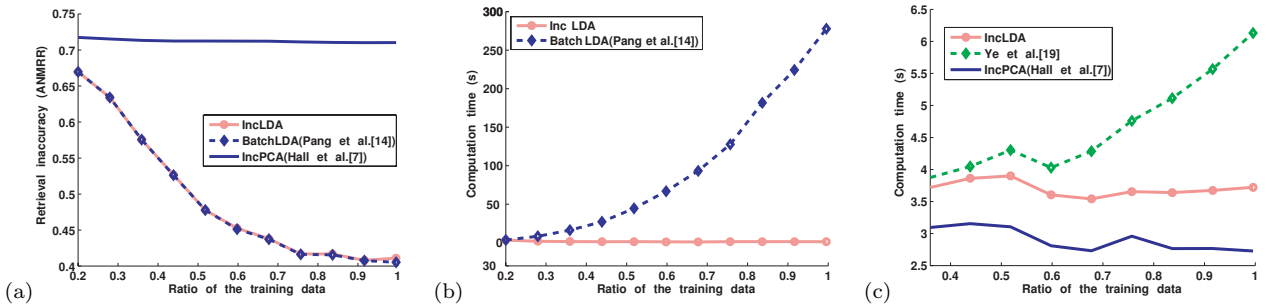


Fig. 3 Database merging experiments for the *MPEG+XM2VTS* data set: *The solution of incremental LDA (with the true class labels of new data) closely agrees to the batch solution while requiring much lower computation time. (a) Retrieval inaccuracy (ANMRR). (b) Computational cost. (c) The update time for the methods in [19, 20] significantly increases when the number of classes is large.*

The training set was further partitioned into an initial training set and several new sets which are added successively for re-training. We performed three experiments using the combined set of MPEG and *XM2VTS* database, the *Altkom* and *BANCA* database. For the MPEG and *XM2VTS* database, the total number of classes (persons) is 930 and each class has 5 images. The data set was divided into 465 persons for training and 465 persons for testing. The training set initially consists of 93 persons (5 images per person) and is augmented 10 times by 37 persons (5 images per person) each time. The new train sets, thus, contain the images of *new classes*. We also performed the experiments for the *Altkom* and *BANCA* database separately where additional sets contain new images of *existing classes* in the initial training set. For the *Altkom* database, the total data set was divided into 40 persons for training and 40 persons for testing. The *BANCA* database was similarly equally divided into 26 persons for training and 26 persons for testing. See Section 5.1.3 for the detailed settings on the Altkom and BANCA datasets.

We report the retrieval performance (ANMRR) and the computation time during updates. In the incremental LDA method, initially, the eigenspace models of the total and between-class scatter matrices of the first train set are built in batch mode and the LDA projection is computed using the eigenspace models. Whenever a new train set is added, the eigenspace models of the new train set are obtained in batch mode, merged with those of the previous, then the LDA projection is computed using the merged eigenspace models. Therefore, the initial computation time is dependent on the size of the first train set and the computation time of subsequent updates is determined by the additional set size, which is fixed, and the subspace dimensions, which are varying accordingly to data variance, during updates. The subspace dimensions are automatically chosen accordingly to the variance of the merged data in each update.

5.1.2 Results on MPEG+XM2VTS by adding new classes

The accuracy of the incremental solution can be seen in Figure 3 (a). Incremental LDA yielded nearly the same solution as batch LDA. 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 (almost constant).

The incremental solution 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 beneficial in terms of computational cost. See Figure 4 for the performance of the proposed method with different number of components. Using more components gave better accuracy but increased the computational time. The computation time of the method except the blue line remains low and approximately constant during the update after the first two steps (the additional set size is fixed and the merged data variance dose not largely change). In the incremen-

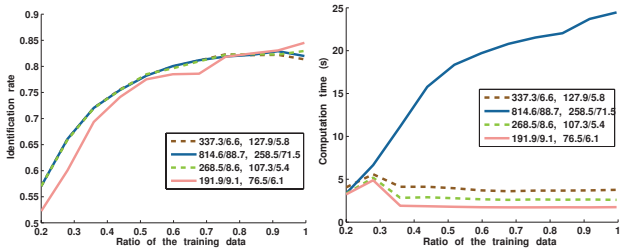


Fig. 4 Performance of incremental LDA for the different subspace dimensions: *Identification rate (left) and computation time (right) on the MPEG+XM2VTS experiment. Each line is indexed by mean/stddev. of $d_{T,3}$ and mean/stddev. of $d_{B,3}$.*

tal learning, we chose the subspace dimensions, $d_{T,3}, d_{B,3}$, to represent most data energy from the eigenvalue plots[3].

Table 3 shows the number of components selected during the experiment using the *MPEG+XM2VTS* data set shown in Figure 3. Even if the total number of images or classes increases, the number of components does not increase significantly (actually it remains almost constant). This means that finding new directions of the components was sufficient to reflect the variation of the increasing data, not adding new dimensions.

Pang et al. [14] have addressed only the efficient update of scatter matrices for LDA leaving the crucial step, subspace analysis, be the same as batch computation. For all our experiments, the scatter matrices are efficiently updated by (13) and (19) in both batch and incremental solutions. Therefore, the batch LDA in the experiment is very close to Pang et al.'s method, which costs much more time than the proposed incremental LDA method. We have also implemented Ye et al.'s incremental LDA method [19]. Note that the original algorithm of Ye et al.'s can only take a single new data point. The incremental PCA method of Hall et al. [7] is integrated into the algorithm to take a set of new data points: the update of the within-class scatter matrix is done by the incremental PCA and the rest of steps remains the same except that they are processed for a chunk of new data, not for a individual data point. Running the original algorithm 37×5 times (we add 37×5 images) in each time update for the experiments is highly time-demanding, as the algorithm involves the process of $O(C^3)$ computations (C is the number of classes), which is similar to [20]. As shown in Figure 3 (c), the computation time of Ye et al.'s method yet significantly grows compared to the proposed method when the number of classes becomes large. The cost of our incremental LDA method is comparable to that of Hall et al.'s incremental PCA method while giving a much higher retrieval accuracy as shown in Figure 3 (a) and (c). The computation time of the incremental PCA and LDA methods in Figure 3 (c) is dependent on the dimension of the eigenspace models used. They were automatically chosen according to the accumulated data variance, which varied by different images to add in each step. Overall, from a certain point, they remain approximately constant not increasing.

[3] Note that accuracy of LDA is dependent on the subspace dimension of the total scatter matrix and the number of discriminant components. They were set to be the same for batch LDA and incremental LDA.

5.1.3 Results on Altkom, BANCA by updating existing classes and semi-supervised incremental LDA

Figure 5 (a-c) shows the label propagation accuracy, i.e. the ratio of the number of correctly estimated samples and the total number of unlabeled samples, for the Altkom, BANCA and ETH80 dataset respectively. For the Altkom dataset (Figure 2 (c)), we use 40 persons, 15 images per person. The leftmost 3 to 13 images per person are labeled and the rest of images are unlabeled. For the BANCA dataset (Figure 2 (d)), we use 260 images of 26 persons and use the leftmost 3,5,7,9 labeled images per person and the rest of it unlabeled. For evaluating the proposed method over other label propagation methods, we use the ETH80 dataset [24]. It contains 8 object categories as shown in Figure 6 and in each category there are 10 different objects, and for each object there are 41 different poses. We randomly draw 9,18,27,...,81 labeled samples of apples, pears and tomatoes (10-fold cross-validation was performed) as in [37] (we directly compare the accuracies reported in [37]). 20×20 pixel gray-value images were used. LDA was computed with the labeled train data and class label estimation of the unlabeled samples was done by the maximum posterior probabilities (27). The EM algorithm in the LDA subspace converged after ten iterations in all three experiments. The label propagation accuracy reasonably improves when more labeled images are used as shown in Figure 5 (a-c). The proposed method delivers the comparable accuracy to Linear Neighborhood Propagation (LNP) [39] method, outperforming Gaussian Kernel Similarity (GKS) [38] and K-Nearest Neighbor (KNN) method. It lags behind Sparsity Induced Similarity (SIS) [37] method in accuracy, but note that the proposed method is an efficient incremental method whereas the SIS is a purely batch method that is hard to cope with a large scale dataset in both memory and time. Despite a standard EM incorporated into our method, the LDA learns a class-discriminative subspace, greatly facilitating the label propagation. The label propagation accuracy of the proposed method may be further improved by combining it

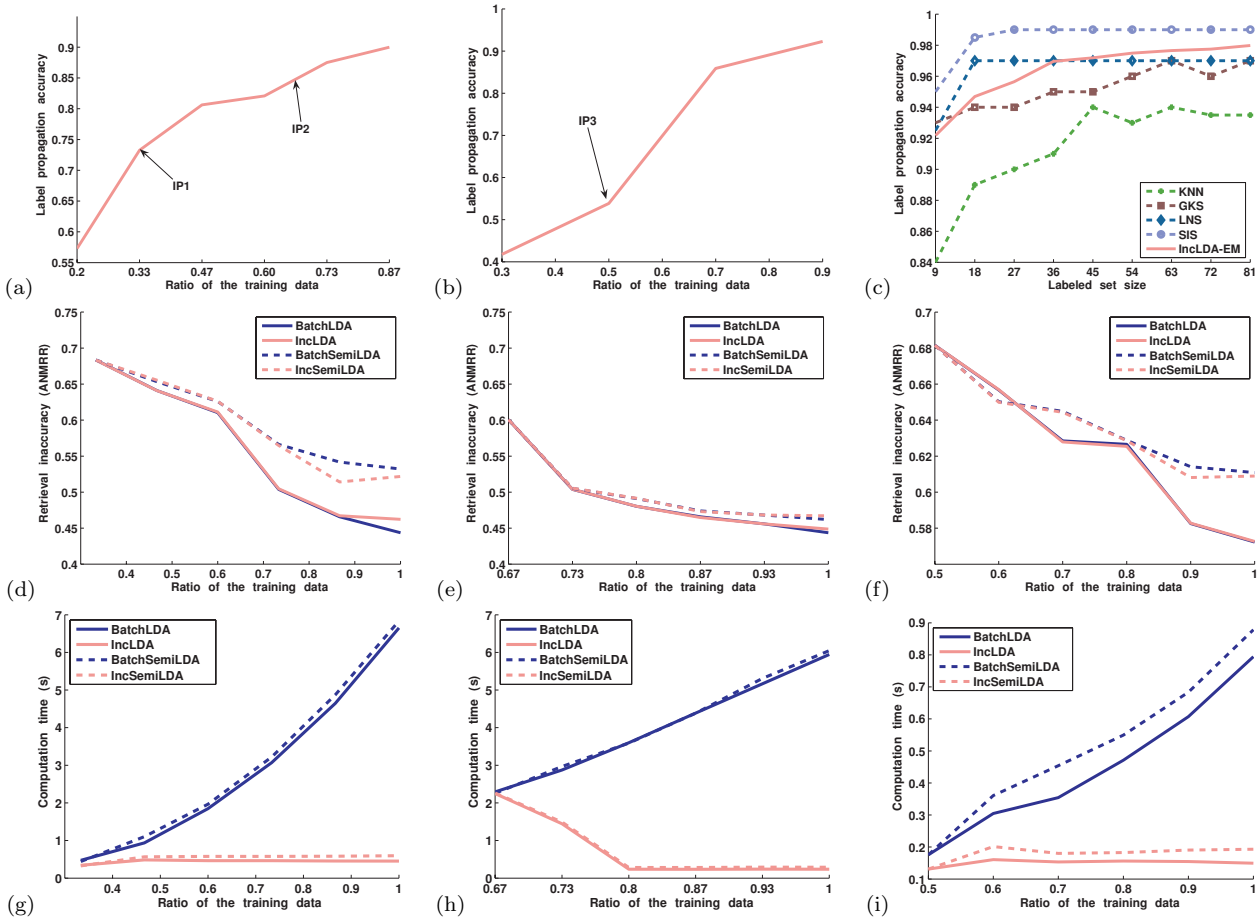


Fig. 5 Performance of semi-supervised incremental LDA: *Label propagation accuracy based on the semi-supervised learning for (a) Altkom (b) BANCA (c) ETH80 dataset. The proposed method exhibits comparable accuracy to Sparsity Induced Similarity (SIS) [37], Gaussian Kernel Similarity (GKS) [38], Linear Neighborhood Propagation (LNP) [39] and K-Nearest Neighbor (KNN) method on the ETH80 dataset. Retrieval inaccuracy (ANMRR) and computation costs for the* Altkom *database when the amount of initial labeled data is (d,g) 33 percent (e,h) 67 percent, and (f,i) for the* BANCA *database when the half of the train set is labeled. Semi-supervised incremental LDA method decreases the retrieval inaccuracy without the class labels of new training data being available, while being as time-efficient as incremental LDA with given labels. The accuracy difference between the two methods is smaller when using more labeled data.*

Fig. 6 ETH80 data set contains 8 different object categories.

with spectral analysis [38], sparsity measures [37], etc, which remains as our future work.

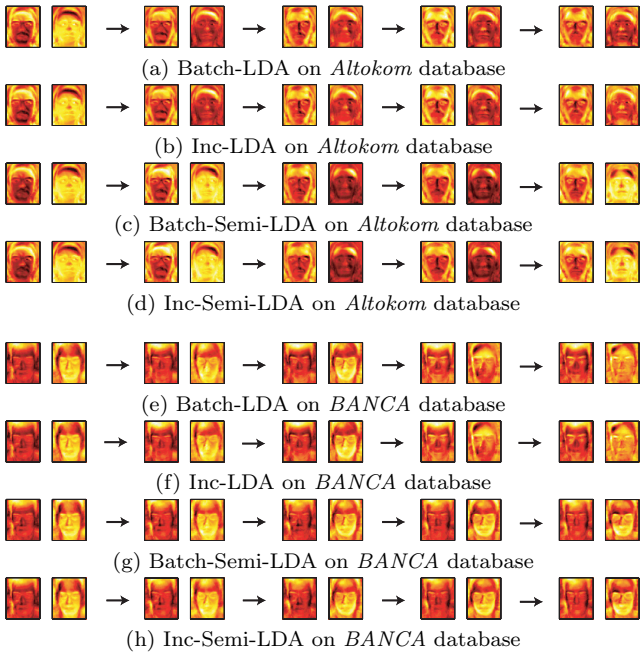
Figure 5 (d-i) shows the results of face image retrieval by incremental learning with new images of existing classes. It compares the proposed semi-supervised incremental LDA, the semi-supervised batch LDA, the incremental LDA and the batch LDA. As in the previous section, the whole data set is partitioned into the two halfs, one for training the LDA bases, and the other for evaluating the retrieval performance. The true class labels of the initial train data are given for the semi-supervised methods, while all train data are labeled for the incremental and batch LDA methods. In the semi-supervised methods, the train data points are projected into the LDA subspace with the most recent LDA components computed either by the incremental or batch method before the EM iteration. The LDA is carried out using the probabilistic labels (27) by EM. The EM algorithm converged typically after ten iterations. We took the two points in Figure 5 (a) denoted as *IP1* and *IP2*: for IP1 the leftmost 5 images per person (Figure 2 (c)) are used as the initial labeled train set and the next 2 images per person are added without labels at each update having 5 updates in total (40 persons), and for IP2 the leftmost 10 images are initially labeled and the single next image is added without labels each time having 5 updates in total (26 persons). Similarly, for IP3 in Figure 5 (b), the leftmost 5 images per person (Figure 2 (d)) serve as the initial labeled train set and

	init	1st	2nd	3rd	4th	5th
IP1	0.00	10.00	13.89	17.50	19.42	22.00
IP2	0.00	1.36	2.92	4.62	6.25	8.33
IP3	0.00	4.49	13.19	18.27	22.22	26.15

Table 4 Label error accumulation during the update: *The number of mislabeled samples/the total train size, during updates, is reported based on the proposed semi-supervised learning for the three different initial points (IP1 and IP2 for Altkom, IP3 for BANCA dataset). Error accumulation of IP2 is smaller than that of IP1 owing to more labeled initial data.*

the next image per person as unlabeled new train data each time, thus having 5 updates in total. The retrieval accuracies are shown in Figure 5 (d-f) and the computation time in Figure 5 (g-i) for IP1, IP2 and IP3 respectively. The incremental LDA gives the close accuracy to that of the batch LDA at much lower computation time. The semi-supervised solution effectively decreases the retrieval inaccuracy even without the class labels of new train data and its incremental solution yields the same solution as the batch version. Table 4 shows the label propagation error accumulated during the update. As shown in Figure 5 (d-f), the accuracy gap between the semi-supervised methods and supervised methods grew as more label errors were accumulated. However, the error accumulation is reasonably slow and the proposed method continually improves the retrial accuracy owing to the use of probabilistic soft labels which mitigate the effect of wrong labels. The accuracy loss by the semi-supervised methods is smaller as more labeled initial train data are used (see Figure 5 (d,e)). The cost of semi-supervised LDA methods is slightly higher than that of supervised methods, as the EM iterations are performed in the low-dimensional (equivalent to the number of classes-1) LDA subspace. Note that the semi-supervised incremental LDA requires far lower computation time than the batch LDA. The computation time in Figure 5 (g,h,i) is measured as in the MPEG-XM2VTS experiment. Therefore, the initial time is dependent on the size of the first train set and the time for subsequent updates by the additional set size, which is fixed, and the subspace dimensions, which are varying for the variance of the merged data each time.

See Figure 7 for the updated bases. The bases incrementally updated look almost identical to those of batch computation for both supervised and semi supervised learning. We have also measured cross-correlations (i.e. similarity in direction not scale) of the LDA vectors computed by the batch method and the proposed incremental method. The *Altkom* database of 80 classes (2 images per class) was divided into two disjoint sets and the two sets were merged by the methods. The size of the first set was increasing (from 1 to 79 for the number of classes) along the x-axis of Figure 8 (left) with the second set accordingly decreasing. Figure 8 (left) shows the mean values of cross-correlations of all 79 (the number of classes-1) vectors. It tends to have a lower peak when the two sets are of the same size. Regardless of the set size (even if a set is very small), the log of cross-correlations were very close to zero (when perfect match), which has been similarly observed in [7]. See also the example pairs of the highly-correlated and less-correlated with their values in the figure. Figure 8 (right) shows the mean value of cross-correlations over all merging for different components.



(a) Batch-LDA on *Altokom* database
(b) Inc-LDA on *Altokom* database
(c) Batch-Semi-LDA on *Altokom* database
(d) Inc-Semi-LDA on *Altokom* database
(e) Batch-LDA on *BANCA* database
(f) Inc-LDA on *BANCA* database
(g) Batch-Semi-LDA on *BANCA* database
(h) Inc-Semi-LDA on *BANCA* database

Fig. 7 Basis update: *The first two LDA components are shown at each update. Whereas the first components are rather steady, the second components are gradually changed, i.e. updated. The bases incrementally updated look almost identical to those of batch computation for both supervised and semi-supervised learning. Those learnt by the proposed semi-supervised method also look similar to those of the method using labels of new samples.*

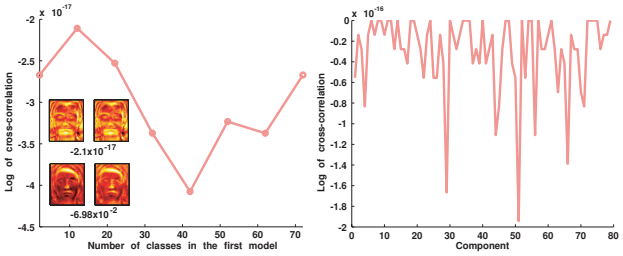


Fig. 8 Cross-correlations of the LDA components computed by the batch method and the incremental method.

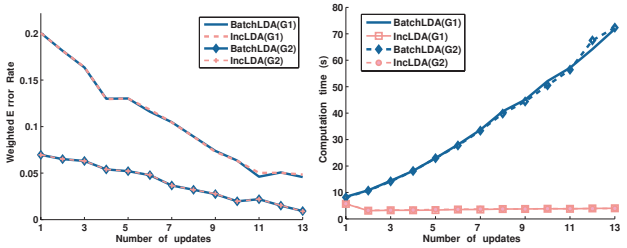


Fig. 9 Incremental LDA for BANCA face authentication: *(left) Decrease of the weighted error rate of the batch and incremental LDA methods for the number of updates. (right) Computational time (sec) of the two methods. For clarity, we only show the case of (R=1) for the group 1 and (R=10) for the group 2.*

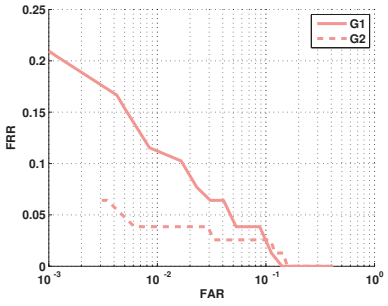


Fig. 10 DET curves for the BANCA dataset: *Using the group 1 and 2 in the Mc protocol.*

	R=0.1		R=1		R=10		
	G1	G2	G1	G2	G1	G2	Av
Proposal	3.98	1.43	4.58	2.23	1.79	0.86	2.48
HMM	7.52	4.90	5.45	0.64	2.56	0.12	3.53
LDA-auxdata	6.53	1.17	7.05	2.88	1.28	2.10	3.50
LDA-color	7.12	0.89	5.58	1.98	1.47	0.92	2.99
DLFA	4.12	3.90	3.04	3.10	1.97	2.12	3.04
*LBP-gmm	0.75	6.26	1.63	7.37	1.22	2.77	3.33
*Gabor-gmm	1.05	0.42	0.77	2.31	0.45	4.20	1.53
*Gabor-kda	0.86	2.18	2.34	4.81	2.32	2.02	2.42

Table 5 Weighted Error Rates: *Using the groups G1 and G2 in the Mc protocol at the three different operating points. The proposed method outperforms the still-based methods and yields the comparable accuracy to the video-based methods in the BANCA competitions [41,40]. *: video-based methods. See text for more explanations.*

5.2 Face authentication using the BANCA database

The BANCA database is one of the most popular benchmark datasets for face verification. The dataset has face images of 260 persons in 5 different languages, but only the English subset, a total of 52 persons, is used in the competitions [41,40]. The 52 persons are divided into two sets of users, which are called G1 and G2, respectively, each set having 13 males and 13 females. When G1 is used as a development set (to build the users template/model i.e. LDA in our method), G2 is used as an evaluation set. Their roles are then switched for two-fold cross-validation. For each face, there are 12 images collected. We used the match controlled (Mc) protocol, which is adopted in both still-based [41] and video-based competition [40]. In the Mc protocol, session 1 data is used for enrolment whereas the data from sessions 2,3,4 are reserved for testing. Example images of sessions 1-4 are the leftmost four images in Figure 2 (d). Note that a sequence of images is used in the video-based competition [40] while a single image in the still-based competition [41]. We used the pre-registered face images provided [41]. The accuracy measurement is the Weighted Error Rate (WER) for the test data of groups G1 and G2 at the three different values of R. The WER is defined as $WER(R) = (FRR + R \cdot FAR)/(1 + R)$, where FRR and FAR are the false rejection rate and false acceptance rate respectively.

In the proposed method, face images are represented as Multi-Scale Local Binary Patterns [2] and the incremental LDA is applied to the histogram vectors. An image is first divided into $m \times n$ non-overlapping blocks. For each pixel in every block the change in the relative intensity values of the neighboring pixels (P) that are at a distance R from it is calculated. For a given block b, P and R, a histogram $H^b_{(P,R)}$ of these changes is obtained by bagging them into $h \in [0, (P-1)P+2]$ bins. Individual bins in the histogram represent either the orientation of edge, a maxima/minima location or otherwise. The histograms of various values of P and R in a given block are concatenated into a column vector, $[H^b_{(P,R)}], \forall P, R$. Chan et al. have suggested the values for $R \in [1, 10], P = 8$ and m, n are taken to be equal to 4 giving a feature vector of length of 590 per block, 16 blocks in total. LDA is trained using the images of the development set and 10 randomly perturbed enrollment images of the evaluation set. LDA is applied to each block, having 16 LDA projection matrices learnt in total. The similarity score of two face images is given as the sum of cross-correlations of the projected vectors over 16 blocks.

Figure 9 (left) and (right) shows the weighted error rate (WER) and computational time of the batch LDA and the incremental LDA method, when the images of two persons were initially given and the images of two more persons were added each time having 13 updates in total. Computation time of all 16 LDA projection matrices on the histogram vectors was measured. The WER decreases as more train images are used. The incremental LDA method delivers the close accuracy to the batch LDA at much lower computational time for both G1 and G2. Figure 10 shows the DET curves, whose x-axis is FRR and y-axis is FAR, of the proposed

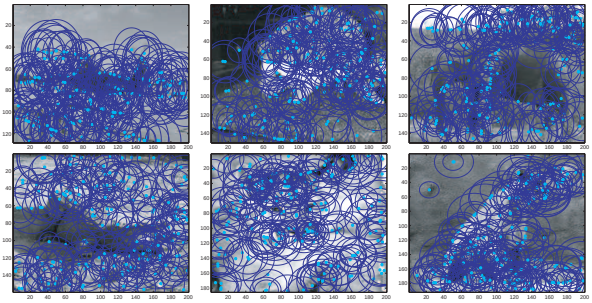


Fig. 11 Example images of Caltech101 data set: *Different class images are shown with the interest points detected.*

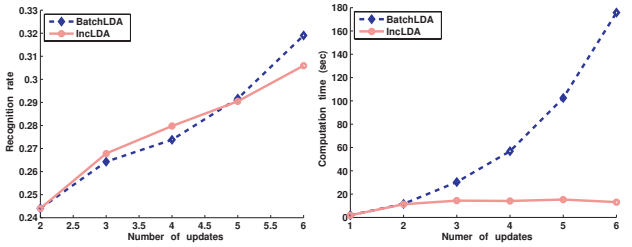


Fig. 12 Object categorisation accuracy on Caltech101 dataset: *(left) Accuracy improvement of the incremental and batch LDA method for the number of updates. (right) Computational time (sec) of the two methods.*

Method		Method	
IncLDA-NN	26.90 (34.57)	IncLDA-SVM	33.57 (42.39)
IncPCA-NN	21.19	IncPCA-SVM	18.10

Table 6 Classification accuracy for Caltech101 dataset: *The methods are evaluated on 84 categories using 10 images per category. The numbers in bracket are obtained by the setting in [34], i.e. using 15 images for training and the rest of images for testing of all 101 categories (5-fold cross validation was performed). The accuracies of state-of-the-arts from [34, 35] are: Baseline [32]: 14.5, Fei Fei [31]: 15.5, Mutch Lowe (base): 33, Serre et al. : 35, Holub et al. [35] : 37, Berg et al. [32] : 45, Grauman Darrell [33]: 49.5, Mutch Lowe (final): 51 %.*

method at the final update (i.e. using all 26 persons of the development set defined by the Mc protocol). Table 5 compares the performance of our method with the top-runners in the competitions: Pseudo-2D Hidden Markov Models (HMM), LDA trained on the symmetrised face images using a large auxiliary dataset, Dynamic Local Feature Analysis (DLFA), and LDA applied to colour channels, all of which are still-based methods from [41], and Local Binary Patterns with Gaussian Mixture Model (GMM), Gabor features with GMM, Kernel Discriminant Analysis (KDA) on Gabor features, all of these are video-based methods from [40]. Our method outperforms all still-based methods and one video-based method on average, and all video-based methods for $R = 10$. Note also that the methods in [41, 40] use different features, classifiers and even large auxiliary data sets, but often adopt LDA as a component. The proposed incremental LDA method as a general meta-algorithm could be conveniently applied to various other methods.

5.3 Object cateogorisation by Caltech101 dataset

We have tested our incremental LDA method on the object categorisation problem using the Caltech101 dataset. The data set consists of 101 object categories with varying number of images up to 800 per category [31]. Mostly objects are presented in real cluttered backgrounds (cf. the ETH80 dataset in the previous section were captured in the uniform background). For the online learning experiment, we used 84 categories removing the background category and the categories that have less than 40 images. 40 images were exploited per category. The 40 images per category were partitioned into 30 for training and 10 for testing. The training data was further partitioned into 6 sets, each of which has 5 images per category. The train data was incrementally grown by adding one set each time. In each image, interest points were detected by Harris-corners and represented by Scale-Invariant-Feature Transform (SIFT) 128 dimensional vectors. Some example images with the interest points detected are shown in Figure 11. The k-means clustering (k was set 1000) was performed on the set of SIFT vectors collected from entire training images to form a codebook and all train and test images were represented as the histograms of codewords, i.e. Bags of Words (BoW). The LDA projection matrix was learnt using the histograms of the train data and Nearest Neighbour classification of the test data was performed in the LDA subspace. As shown in Figure 12, the incremental LDA algorithm effectively boosts the categorisation accuracy (from 24.4 to 31.9 percents) when more training images are available. The proposed incremental LDA method delivers the close accuracy to that of the batch LDA method at much lower computation time (see Figure 12 right).

Table 6 shows the accuracies of the four methods using 15 image per category for training and 10 image per category for testing (i.e. at the 3rd update in Figure 12). Nearest Neighbour classification is performed in the IncLDA-NN and IncPCA-NN methods, while Support Vector Machine is applied to the LDA or PCA features in the IncLDA-SVM or IncPCA-SVM methods. $C(C-1)/2$ one vs one linear SVMs are used (C is the number of classes) and multi-class classification is done by majority voting. The LDA methods significantly outperforms the PCA methods at the same dimension (set as 60 in the experiments). The

LDA-SVM method largely improves the accuracy of the LDA-NN method, whereas the PCA-SVM is not better than the PCA-NN method. The proposed incremental LDA method efficiently captures discriminative information in a low-dimensional space (the input dimension was reduced from 1000 into 60) facilitating large-scale data storage and time-efficient SVM learning/evaluation. The incremental LDA method as a dimension reduction method should be of a value to many other methods in the area. For the comparison with state-of-the-arts, we followed the protocol of [34,35] using 15 images per category for training and all the rest of images per category for testing, using 101 categories (5-fold cross validation was performed). The accuracies of the proposed methods by this setting are shown in the bracket in Table 6. The proposed method delivers comparable accuracy to other methods. Note that standard techniques were exploited for representation in our method: Harris-corners, SIFT, k-means clustering methods as in the baseline method [32]. The LDA combined with the standard representation largely improves the accuracy of the baseline method ($14.5 \rightarrow 34.57\%$). The accuracy of the proposed method could be further improved by incorporating better image features and representations e.g. the multi-layer features of [34] and Random Forest codebook techniques [36].

6 Discussion on updating LDA-like discriminant models

The proposed three-step algorithm is general and can be applied to other incremental learning problems that seek to find discriminative components by maximizing the ratio involving two different covariance or correlation matrices [3,5,13]. The method of using the sufficient spanning set for the three steps, the component analysis of the two matrices in the numerator and the denominator, respectively, and for the discriminant component computations, allows for efficient incremental learning. Note that the number of input vectors for the numerator matrix in many methods such as the Oriented Component Analysis (OCA) [5] and Orthogonal Subspace Method (OSM) [13,22] criteria, is often large in practice. In these cases the previous incremental LDA algorithms suffer due to the assumption of a small number of input vectors for the scatter matrix in the numerator (e.g. the number of classes in the LDA). The proposed method can also be applied to an LDA mixture model [30] as in [7], or other LDA variants including direct LDAs [29] if they are piecewise linear models and are based on the Rayleigh quotient. See [22] for the application of the three-step update algorithm to the OSM for set-based object recognition.

7 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 has been also incorporated into a semi-supervised learning framework by label propagation. The experiments have shown the usefulness of the incremental LDA method as a general meta-algorithm, being combined with various image representations, for face image retrieval, face authentication, and object categorisation problems.

Directions for future research include the extension to the non-linear case, adaptive learning with time-series data. Active learning for the incremental LDA method would be also interesting for identifying unlabeled examples whose labels are most helpful to improve the classification performance.

Acknowledgment

This study has been funded in part by the Toshiba-Cambridge Scholarship. T-K. Kim is presently supported by the research fellowship of the Sidney Sussex College of the University of Cambridge. J. Kittler was partially supported by EU Projects VidiVideo and Mobio.

References

1. X. Wang, T.X. Han and S. Yan. An HOG-LBP Human Detector with Partial Occlusion Handling. In Proc. of *ICCV*, Kyoto, 2009.
2. C-H. Chan, J. Kittler, K. Messer. Multi-scale Local Binary Pattern Histograms for Face Recognition. In Proc. of *ICB* pages 809-818, 2007.
3. A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a Mahalanobis metric from equivalence constraints. *JMLR*, 6:937–965, 2005.
4. T.-J. Chin and D. Suter. Incremental Kernel PCA for Efficient Non-linear Feature Extraction. In Proc. of *BMVC*, 2006.
5. F. De la Torre Frade, R. Gross, S. Baker, and V. Kumar. Representational oriented component analysis (ROCA) for face recognition with one sample image per training class. In Proc. of *CVPR*, 2005.
6. K. Fukunaga. *Introduction to statistical pattern recognition.* Academic Press, Boston, 1990.
7. P. Hall, D. Marshall, and R. Martin. Merging and splitting eigenspace models. *IEEE Trans. on PAMI*, 22(9):1042–1049, 2000.
8. K. Hiraoka, K. Hidai, M. Hamahira, H. Mizoguchi, T. Mishima, and S. Yoshizawa. Successive learning of linear discriminant analysis: Sanger-type algorithm. In Proc. of *ICPR*, 2000.
9. T. Kamei, A. Yamada, T. Kim, H. Kim, W. Hwang, S. Kee. Advanced face descriptor using Fourier and intensity LDA features. ISO/IEC JTC1/SC29/WG11 M8998, Oct 2002.

10. 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.
11. R.-S. Lin, D. Ross, J. Lim, and M.-H. Yang. Adaptive discriminative generative model and its applications. In Proc. of *NIPS*, 2005.
12. B. S. Manjunath, P. P. Salembier, and T. Sikora. *Introduction to MPEG-7: Multimedia Content Description Interface*. Wiley, New York, 2002.
13. E. Oja. *Subspace Methods of Pattern Recognition*. Research Studies Press, 1983.
14. 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.
15. D. Skocaj and A. Leonardis. Weighted and robust incremental method for subspace learning. In Proc. of *ICCV*, 2003.
16. Y. Wu and S. Huang. View-independent recognition of hand postures. In Proc. of *CVPR*, pages 2088–2094, 2000.
17. X. Tao, J. Ye, Q. Li, R. Janardan, and V. Cherkassky. Efficient Kernel Discriminant Analysis via QR Decomposition. In Proc. of *NIPS*, 2004.
18. J. Yan, B. Zhang, S. Yan, Q. Yang, and H. Li. IMMC: Incremental maximum margin criterion. In Proc. of *Int'l Conf. Knowledge Discovery and Data Mining*, 2004.
19. 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.
20. M. Uray, D. Skocaj, P. Roth, H. Bischof, A. Leonardis, Incremental LDA learning by combining reconstructive and discriminative approaches. In Proc. of *BMVC*, 2007.
21. X. Zhu. *Semi-Supervised learning literature survey*. Computer Sciences TR 1530, University of Wisconsin-Madison, 2006.
22. B. Stenger, T. Woodley, T-K. Kim, C. Hernandez, R. Cipolla. AIDIA - Adaptive Interface for Display InterAction. In Proc. of *BMVC*, Leeds, UK, 2008.
23. T-K. Kim, S-F. Wong, B. Stenger, J. Kittler and R. Cipolla, Incremental Linear Discriminant Analysis Using Sufficient Spanning Set Approximations, In Proc. of *CVPR*, Minneapolis, MN, 2007.
24. B. Leibe and B. Schiele, Analyzing appearance and contour based methods for object categorization. In Proc. of *CVPR*, pp. 409–415, 2003.
25. T-K. Kim, J. Kittler and R. Cipolla, Discriminative Learning and Recognition of Image Set Classes Using Canonical Correlations. *IEEE Trans. on PAMI*, Vol.29, No.6, June 2007.
26. J. Winn, A. Criminisi and T. Minka, Object Categorisation by Learned Univeral Visual Distionary In Proc. of *ICCV*, 2005.
27. J.C. Niebles, H. Wang and L. Fei-Fei. Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. *IJCV*, 2008.
28. C. Bouveyron, S. Girard and C. Schmid. Dimension Reduction and Classification Methods for Object Recognition in Vision. In Proc. of *5th French-Danish Workshop on Spatial Statistics and Image Analysis in Biology*, Saint-Pierre de Chartreuse, France, May 2004.
29. H. Yu and H. Yang. A direct LDA algorithm for high-dimensional data - with application to face recogniion. *Pattern Recognition*, 34(10):2067-2070, 2001.
30. H-C. Kim, D. Kim and S.Y. Bang. Face recognition using LDA mixture model. *Pattern Recognition Letters*, 24(15):2815–2821, 2003.
31. L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples. In Proc. of *CVPR Workshop on GMBV*, 2004.
32. A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In Proc. of *CVPR*, 2005.
33. K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In Proc. of *ICCV*, 2005.
34. J. Mutch and D. Lowe. Multiclass Object Recognition with Sparse, Localized Features. In Proc. of *CVPR*, 2006
35. AD. Holub, M. Welling, P. Perona. Exploiting Unlabelled Data for Hybrid Object Classification. In Proc. of *NIPS Workshop on Inter-Class Transfer*, 2005.
36. F. Moosmann, B. Triggs and F. Jurie. Fast Discriminative Visual Codebooks using Randomized Clustering Forests. In Proc. of *NIPS*, 2007.
37. H. Cheng, Z. Liu, J. Yang. Sparsity Induced Similarity Measure for Label Propagation. In Proc. of *ICCV*, 2009.
38. M. Belkin and P. Niyogi. Learning with local and global consistency. In Proc. of *NIPS*, 2004.
39. F. Wang and C. Zhang. Label propagation through linear neighborhoods. In Proc. of *ICML*, 2007.
40. N. Poh et al. Face Video Competition. In Proc. of *ICPR*, 2009.
41. K. Messer et al. Face Authentication Competition on the BANCA Database. In Proc. of *ICPR*, 2004.
42. http://mi.eng.cam.ac.uk/∼tkk22.