

# Incremental LDA Learning by Combining Reconstructive and Discriminative Approaches \*

Martina Uray<sup>1</sup>, Danijel Skočaj<sup>2</sup>, Peter M. Roth<sup>1</sup>,  
Horst Bischof<sup>1</sup> and Aleš Leonardis<sup>2</sup>

<sup>1</sup> Institute for Computer Graphics and Vision, Graz University of Technology  
{uray, pmroth, bischof}@icg.tugraz.at

<sup>2</sup> Faculty of Computer and Information Science, University of Ljubljana  
{danijel.skocaj, ales.leonardis}@fri.uni-lj.si

## Abstract

Incremental subspace methods have proven to enable efficient training if large amounts of training data have to be processed or if not all data is available in advance. In this paper we focus on incremental LDA learning which provides good classification results while it assures a compact data representation. In contrast to existing incremental LDA methods we additionally consider reconstructive information when incrementally building the LDA subspace. Hence, we get a more flexible representation that is capable to adapt to new data. Moreover, this allows to add new instances to existing classes as well as to add new classes. The experimental results show that the proposed approach outperforms other incremental LDA methods even approaching classification results obtained by batch learning.

## 1 Introduction

When discussing object classifiers one faces two main philosophies, generative and discriminative classifiers. Whereas generative models try to find an efficient representation of the data, discriminant classifiers are particularly designed for classification tasks. Compared to generative methods this allows to train more specific classifiers resulting in higher recognition rates. In fact, several studies (e.g., [1, 13]) have shown that discriminative classifiers outperform generative models (if enough training data is available).

But discriminative classifiers have several drawbacks, e.g., a huge amount of training data is necessary, they are not robust (neither during learning nor during evaluation), or they can not take advantage of unlabeled data. Hence, there was a considerable interest to bring both philosophies together (e.g., [2, 7, 9–12]). Most of these approaches are based on two-stages (e.g., [7, 9, 10]). In the first stage a generative model (probabilistic constellation model [7], clustering algorithm [9], probabilistic PCA [10]) is estimated

---

\*This research has been supported by: Research program Computer Vision P2-0214 (Slovenian Ministry of Higher Education, Science and Technology), EU FP6-004250-IP project CoSy, EU FP6-511051 project MOB-VIS, EU FP6-507752 NoE MUSCLE IST, Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04 and FFG project AUTOVISTA 813395 under the FIT-IT program.

and in the second stage a discriminant classifier (SVM on Fisher Scores [7], multi-layer perceptron [9], negative samples [10]) is built from the generative model.

In the field of subspace methods this combination is mainly addressed by classifier fusion of a generative model (mostly PCA) and a discriminative model (e.g., LDA) which improves the recognition results (e.g., [11, 12]). Similarly, the combination of PCA and LDA was addressed by Fidler et al. [2]. They constructed a new basis that contains all discriminative information but in addition also contains sufficient reconstructive information to enable a robust classification. Due to its reconstructive properties we will further refer PCA to as a reconstructive model.

But none of these approaches consider the benefits of combining generative and discriminative methods for incremental learning which is desirable for two main reasons. First, if not all data is given in advance (e.g., when processing a video stream) a batch method can not be applied. Second, incremental learning requires less memory and has lower computational costs. In the following we will focus on incremental LDA learning. Existing incremental LDA methods, e.g., [6, 15, 18] are based on directly updating the between class scatter matrix and the within class scatter matrix. That is, they consider mostly discriminative information focusing on specific features that best discriminate the data seen so far. To illustrate that using both, discriminative and reconstructive information, is beneficial consider the following example. Given an LDA classifier trained from a small number of samples to separate three classes. As can be seen in Figure 1 this *initial classifier* separates the classes well in the projected LDA space. This subspace representation is then updated with new instances in two different ways, using discriminative information only and using discriminative as well as reconstructive information. From Figure 1 it can be seen that for the first case (*LDA*) the representation gets more and more dispersed since the classifier is biased by the starting model and can not adapt to the new information. Conversely, if discriminative and reconstructive information is used (*PCA & LDA*) the representation remains compact and still separates the three clusters well.

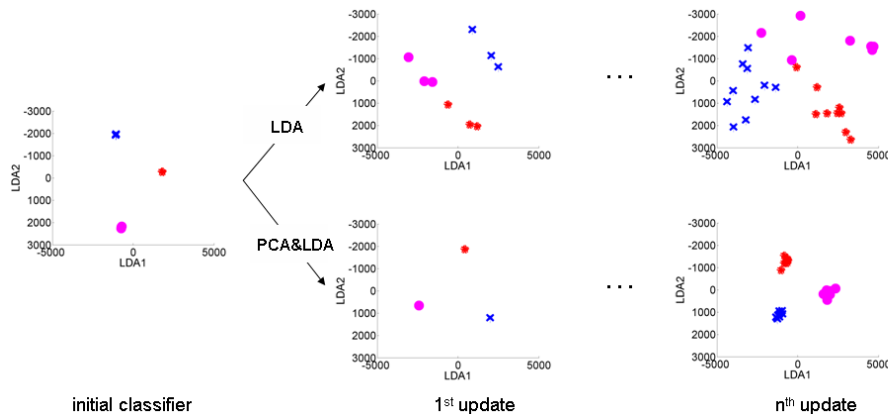


Figure 1: Incremental discriminative subspace learning: using only discriminative information the incrementally estimated model is biased by the first samples and the representation get dispersed (*LDA*); using reconstructive and discriminative information a more generic model is estimated which yields well separable clusters (*PCA & LDA*).

The main contribution of this paper is an incremental discriminative subspace learning algorithm that combines reconstructive and discriminative information during training. In particular, we use Linear Discriminant Analysis (LDA) [3] as discriminative model and Principal Component Analysis (PCA) [8] as reconstructive model. For LDA learning it is common to apply PCA as a preprocessing step for dimensionality reduction or to handle singularity problems (e.g., [1, 17]). Conversely, our approach implicitly combines both models. Since every update increases the size of the representation the updated subspace has to be truncated. Usually the least significant (reconstructive) directions are skipped but this may result in a loss of discriminative information. To avoid this we propose a more sophisticated truncation step. By applying an appropriate constructed orthogonal projection the updated subspace is projected onto another subspace of desired size. Thus, preserving the reconstructive as well as the discriminative information. Moreover, this update procedure allows to incrementally update the current representation by adding new instances to existing classes as well as by adding new classes.

The outline of the paper is as follows: In Section 2 we deduce the combined reconstructive/discriminative subspace projection and show how it can be applied for incremental learning. Next, in Section 3 we demonstrate the advantages of our approach compared to other incremental methods on small as well as large data sets and for different kinds of updates. Finally, we give a conclusion and an outlook in Section 4.

## 2 Incremental Updating by Subspace Combination

### 2.1 Subspace Methods

The goal of subspace methods is to find a projection that transforms the training data (images) such that new unknown images can be efficiently classified. For batch methods it is assumed that all training data is given in advance. Formally, given  $n$  images  $\mathbf{x}_j \in \mathbf{R}^m$  represented as vectors which are organized in a matrix  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbf{R}^{m \times n}$ . Let  $\boldsymbol{\mu} \in \mathbf{R}^m$  be the mean image and let  $c$  be the total number of classes. In the following we take a closer look at the subspace methods that are relevant for our approach.

*Principal Component Analysis* [8] is an unsupervised method that estimates a low dimensional representation of the data that minimizes the squared reconstruction error. Therefore, the data  $\mathbf{X}$  is projected onto a lower-dimensional subspace by

$$\mathbf{A} = \mathbf{U}^T (\mathbf{X} - \boldsymbol{\mu} \mathbf{1}_{1 \times n}), \quad (1)$$

where the projection matrix  $\mathbf{U}$  is build from the eigenvectors of the covariance matrix of  $\mathbf{X}$  and  $\mathbf{1}_{m \times n}$  denotes an  $m \times n$  matrix of ones. Usually only  $k$ ,  $k \ll n \ll m$ , columns are needed to reconstruct  $\mathbf{X}$  to a desired accuracy:

$$\mathbf{X} \approx \mathbf{U}_k \mathbf{A}_k + \boldsymbol{\mu} \mathbf{1}_{1 \times n}, \quad (2)$$

where  $\mathbf{U}_k = [\mathbf{u}_1, \dots, \mathbf{u}_k] \in \mathbf{R}^{m \times k}$  is the truncated reconstructive basis and the rows of  $\mathbf{A}_k = [\mathbf{a}_1^T, \dots, \mathbf{a}_n^T]^T \in \mathbf{R}^{k \times n}$  are the image representations in the truncated subspace.

*Linear Discriminant Analysis* [3], in contrast, is a supervised method that also integrates the class label information of the training samples. It seeks for  $(c - 1)$  hyperplanes that are capable to separate the given data. This is realized by a projection

$\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_{(c-1)}] \in \mathbb{R}^{m \times (c-1)}$  that minimizes the intra-class scatter while it maximizes the inter-class scatter. Hence, a test sample  $\mathbf{x}$  is projected onto the subspace by

$$g(\mathbf{x}) = \mathbf{W}^T (\mathbf{x} - \boldsymbol{\mu}). \quad (3)$$

The class label  $l$  is then assigned according to the result of a nearest neighbor classification. Therefore, the Euclidean distances  $d$  of the projected sample  $g(\mathbf{x})$  and the class centers  $\mathbf{v}_i$  are compared:

$$l = \min_{i=1, \dots, c} d(g(\mathbf{x}), \mathbf{v}_i). \quad (4)$$

## 2.2 Incremental LDA Learning

In this section we deduce an incremental update scheme for an LDA classifier (*ILDAAPCA*). Incremental subspace methods allow to directly add new images to an already built lower-dimensional representation. Thus, the original images can be discarded directly after they have been processed.

To avoid losing any discriminative information we apply the idea of augmented subspaces that was proposed by Fidler et al. [2] for robust classification. Therefore, we first create an augmented PCA subspace by augmenting the  $k$  dimensional reconstructive subspace with additional  $c - 1$  vectors containing discriminative information. Those supplementary vectors are constructed from vectors that would be discarded when truncating the subspace to  $k$ -dimensions. In this way, the full discriminative information is maintained. Second, we build the actual LDA representation from the thus obtained augmented subspace.

Assuming that a subspace representation was already build from  $n$  images the current augmented PCA subspace is given by the *aPCA* vectors  $\mathbf{U}^{(n)} \in \mathbb{R}^{m \times (k+c-1)}$ , the *aPCA* coefficients  $\mathbf{A}^{(n)} \in \mathbb{R}^{(k+c-1) \times n}$ , and the mean vector  $\boldsymbol{\mu}^{(n)} \in \mathbb{R}^{m \times 1}$ ; the current augmented LDA representation is given by the vectors  $\mathbf{V}^{(n)} \in \mathbb{R}^{(k+c-1) \times (c-1)}$  and the class centers  $\mathbf{v}_i^{(n)} \in \mathbb{R}^{(c-1) \times 1}$ . To update the subspace representation with a new image  $\mathbf{x}^{(n+1)}$  first the augmented subspace is updated using an incremental PCA method (*IPCA*) [16] which is outlined in Algorithm 1.

Second, the actual LDA step is carried out on the extended representation (Algorithm 2, step 1) giving the updated augmented LDA space  $\mathbf{V}$  and the corresponding class centers  $\mathbf{v}_i^{(n+1)}$ . Here,  $\mathbf{U}$  and  $\mathbf{V}$  represent those subspaces that correctly describe the updated information. That is,  $g(\mathbf{x})$  projects onto a subspace describing the old data as well as the new input image. But as a consequence the dimension of the subspace representation is increased:  $\mathbf{U} \in \mathbb{R}^{m \times (k+c)}$ ,  $\mathbf{A} \in \mathbb{R}^{(k+c) \times (n+1)}$ ,  $\mathbf{V} \in \mathbb{R}^{(k+c) \times (c-1)}$ .

Finally, in order to retain a pre-specified size we have to truncate the subspaces. To avoid that discriminative information is lost we project these matrices onto another subspace having the desired size while preserving the full discriminative information. Therefore, we want to keep the first  $k$  vectors of the subspaces unchanged. This is achieved by splitting the LDA vectors into sub-matrices  $\mathbf{V}_k \in \mathbb{R}^{k \times (c-1)}$  and  $\mathbf{V}_c \in \mathbb{R}^{c \times (c-1)}$  (Algorithm 2, step 2). The orthogonalized matrix  $\tilde{\mathbf{V}}_c$  (Algorithm 2, step 3) has exactly the desired properties (preserving the discriminative information).

---

**Algorithm 1** : IPCA on Augmented Basis

---

**Input:** Current augmented principal subspace: aPCA vectors  $\mathbf{U}^{(n)}$ , aPCA coefficients  $\mathbf{A}^{(n)}$  and mean vector  $\boldsymbol{\mu}^{(n)}$ ; new input image  $\mathbf{x}^{(n+1)}$

**Output:** New augmented principal subspace: aPCA vectors  $\mathbf{U}$ , aPCA coefficients  $\mathbf{A}$  and mean vector  $\boldsymbol{\mu}^{(n+1)}$

- 1: Project  $\mathbf{x}^{(n+1)}$  onto current subspace:  $\mathbf{a} = \mathbf{U}^{(n)\top}(\mathbf{x}^{(n+1)} - \boldsymbol{\mu}^{(n)})$
  - 2: Reconstruct:  $\mathbf{y} = \mathbf{U}^{(n)}\mathbf{a} + \boldsymbol{\mu}^{(n)}$
  - 3: Compute residual vector:  $\mathbf{r} = \mathbf{x}^{(n+1)} - \mathbf{y}$
  - 4: Append new basis vector  $\mathbf{r}$ :  $\mathbf{U}' = \begin{bmatrix} \mathbf{U}^{(n)} & \frac{\mathbf{r}}{\|\mathbf{r}\|} \end{bmatrix}$
  - 5: Determine coefficients in new basis:  $\mathbf{A}' = \begin{bmatrix} \mathbf{A}^{(n)} & \mathbf{a} \\ \mathbf{0} & \|\mathbf{r}\| \end{bmatrix}$
  - 6: Perform PCA on  $\mathbf{A}'$  obtaining mean value  $\boldsymbol{\mu}''$  and eigenvectors  $\mathbf{U}''$
  - 7: Project coefficient vectors to new basis:  $\mathbf{A} = \mathbf{U}''^\top(\mathbf{A}' - \boldsymbol{\mu}''\mathbf{1}_{1 \times (n+1)})$
  - 8: Rotate subspace  $\mathbf{U}'$  for  $\mathbf{U}''$ :  $\mathbf{U} = \mathbf{U}'\mathbf{U}''$
  - 9: Update mean:  $\boldsymbol{\mu}^{(n+1)} = \boldsymbol{\mu}^{(n)} + \mathbf{U}'\boldsymbol{\mu}''$
- 

Hence, we can construct a projection matrix  $\mathbf{M} \in \mathbb{R}^{(k+c) \times (k+c-1)}$  containing an identity matrix  $\mathbf{I} \in \mathbb{R}^{k \times k}$ , the matrix  $\tilde{\mathbf{V}}_c \in \mathbb{R}^{c \times (c-1)}$ , and the zero matrices  $\mathbf{0}_c \in \mathbb{R}^{c \times k}$  and  $\mathbf{0}_k \in \mathbb{R}^{k \times (c-1)}$  (see Algorithm 2, step 4). Finally, the current subspaces are projected onto  $\mathbf{M}$  and we obtain  $\mathbf{U}^{(n+1)} \in \mathbb{R}^{m \times (k+c-1)}$ ,  $\mathbf{A}^{(n+1)} \in \mathbb{R}^{(k+c-1) \times (n+1)}$ , and  $\mathbf{V}^{(n+1)} \in \mathbb{R}^{(k+c-1) \times (c-1)}$  (Algorithm 2, step 5).

---

**Algorithm 2** : LDA on Updated Augmented Basis

---

**Input:** Updated augmented principal subspace: aPCA vectors  $\mathbf{U}$ , aPCA coefficients  $\mathbf{A}$

**Output:** New subspaces of pre-specified size: aPCA vectors  $\mathbf{U}^{(n+1)}$ , aPCA coefficients  $\mathbf{A}^{(n+1)}$ ; LDA vectors  $\mathbf{V}^{(n+1)}$  and class centers  $\mathbf{v}_i^{(n+1)}$

- 1: Perform LDA on  $\mathbf{A}$  obtaining LDA vectors  $\mathbf{V}$  and class centers  $\mathbf{v}_i^{(n+1)}$
  - 2: Split  $\mathbf{V} = \begin{bmatrix} \mathbf{V}_k \\ \mathbf{V}_c \end{bmatrix}$
  - 3: Orthogonalize  $\mathbf{V}_c$ :  $\tilde{\mathbf{V}}_c = \mathbf{V}_c(\mathbf{V}_c^\top \mathbf{V}_c)^{-1/2}$
  - 4: Build projection matrix  $\mathbf{M}$ :  $\mathbf{M} = \begin{bmatrix} \mathbf{I}_k & \mathbf{0}_k \\ \mathbf{0}_c & \tilde{\mathbf{V}}_c \end{bmatrix}$
  - 5: Project  $\mathbf{U}$ ,  $\mathbf{A}$  and  $\mathbf{V}$ :  $\mathbf{U}^{(n+1)} = \mathbf{U}\mathbf{M}$ ,  $\mathbf{A}^{(n+1)} = \mathbf{M}^\top \mathbf{A}$ ,  $\mathbf{V}^{(n+1)} = \mathbf{M}^\top \mathbf{V}$
- 

The main point is that by this truncation step the projection  $g(\mathbf{x})$  does not change:

$$\begin{aligned} g^{(n+1)}(\mathbf{x}) &= \left( \mathbf{U}^{(n+1)} \mathbf{V}^{(n+1)} \right)^\top (\mathbf{x} - \boldsymbol{\mu}^{(n+1)}) = \\ &= \left( \mathbf{U} \mathbf{M} \mathbf{M}^\top \mathbf{V} \right)^\top (\mathbf{x} - \boldsymbol{\mu}^{(n+1)}) = \\ &= \left( \mathbf{U} \mathbf{V} \right)^\top (\mathbf{x} - \boldsymbol{\mu}^{(n+1)}) = g(\mathbf{x}) . \end{aligned} \tag{5}$$

Thus, while still retaining the dimensions of the subspaces no discriminative information is lost.

### 3 Experimental Results

To show the benefits of the proposed incremental discriminative subspace method we performed two experiments that are also relevant for practical applications: adding new instances to existing classes (e.g., adding a new view of an already known object) and adding new classes (e.g., adding a new person to an authentication system). We compared the recognition results of the proposed *ILDAaPCA* method to the batch method (*batchLDA*) and two different incremental LDA approaches (*ILDAonK* and *ILDAonL*). *batchLDA* builds a new model from scratch in each update step using the same number of images as the incremental algorithms. *ILDAonK* updates a PCA basis truncated to the size  $\hat{k} = k + c - 1$  (to enable a comparison of subspaces of the same size<sup>1</sup>) and then discards the spare eigenvector in each update step. Thus, only reconstructive information is used. Conversely, *ILDAonL* updates the  $(c - 1)$ -dimensional LDA basis directly; only discriminative information is used.

All experiments were performed on three datasets different in size, design, and topic. Color images were converted to gray scale and every second image of each class was used for training and the remaining ones for testing:

1. The *pre-cropped Sheffield Face Database* [5] (denoted as SFD) consists of 20 persons with at least 19 images of each individual covering poses from profile to frontal views (e.g., see Figure 2(a)). To have identical class sizes we used exactly 19 images per individual.
2. The *Columbia Image Database Library* [14] (COIL-100) consists of 100 objects with 72 colored images of views from 0 to 360 degrees in 5 degree steps (e.g., see Figure 2(b)).
3. The *Amsterdam Library Of Images ALOI* [4] is a color image collection of 1000 small objects, where we used the *Object Viewpoint Collection* (in their quarter resolution) which is similar to COIL-100. In contrast to COIL-100, where the objects are cropped to fill the full image in ALOI the images contain the objects in their original size as well as some background (e.g., see Figure 2(c)). For calculation purposes (to be able to compare our results to *batchLDA*) we used only the first 100 objects (denoted as ALOI-100).

**Adding new instances:** For the first experiment we assumed that all classes are present in the starting model. We took two images of each class, having 40 images in total for SFD and 200 images in total for ALOI-100 and COIL-100. Subsequently, in each iteration one additional training image is attached to all classes. For *batchLDA* the model was built from scratch with growing instance number for each class. Figure 3 shows the recognition results for SFD (Figure 3(a)) and ALOI-100 (Figure 3(b)) after each update step. This demonstrates that new images bring additional knowledge into the model improving the current representations resulting in a better performance of the classifier. Even though there are only minor differences between the compared methods for small datasets, our method can handle large data sets considerably better than the other incremental methods.

---

<sup>1</sup>The parameter  $k$  was chosen such that it encompasses 80% of the energy (defined as a fraction of the total variance) of the starting model unchanged during the entire experiment.

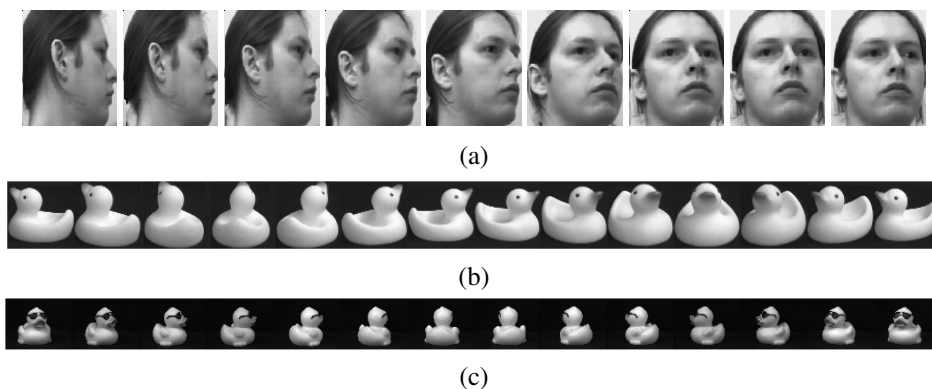


Figure 2: Sample images of one class from (a) SFD, (b) COIL-100, and (c) ALOI.

Moreover, the recognition rates of the proposed method are approaching those obtained by the batch method.

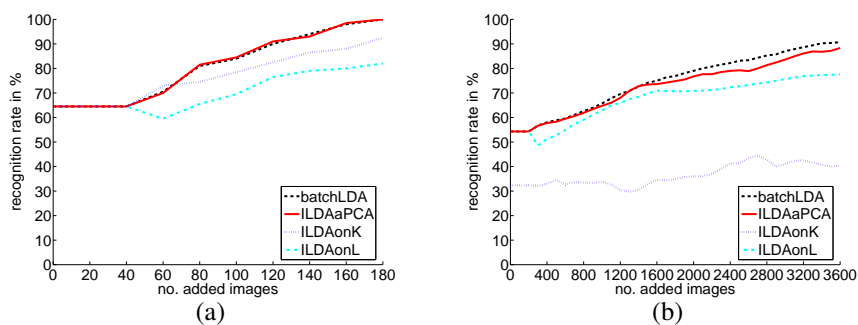


Figure 3: Recognition rates when adding new class instances: (a) SFD and (b) ALOI-100.

In addition, we demonstrate the importance of the proposed subspace truncation step (Algorithm 2, steps 2–5). Therefore, we compared the recognition rate on the Coil-100 database for our approach (see Figure 4(a)) and for an incremental method (denoted as *ILDAaPCAonK*) that starts with an augmented basis but simply truncates the last eigenvector in each update step. In Figure 4(b) it can be seen that the results are similar to *ILDAonK* showing the loss of important discriminative information.

**Adding new classes:** For the second experiment we assumed that only two classes (of full training size) are known in advance. Thus, initially we had a training base of size 18 for SFD and of size 72 for COIL-100 and ALOI-100. Subsequently, we added full classes. *batchLDA* was calculated in each step with full training size of actually known classes. Since the LDA subspace size depends on the number of classes it has to grow for each added class. *ILDAaPCA* handles this automatically; for the other methods we skipped the truncation step for the first image of each new class. For evaluation only test images from already known classes were used. The obtained recognition rates for SFD and ALOI-100

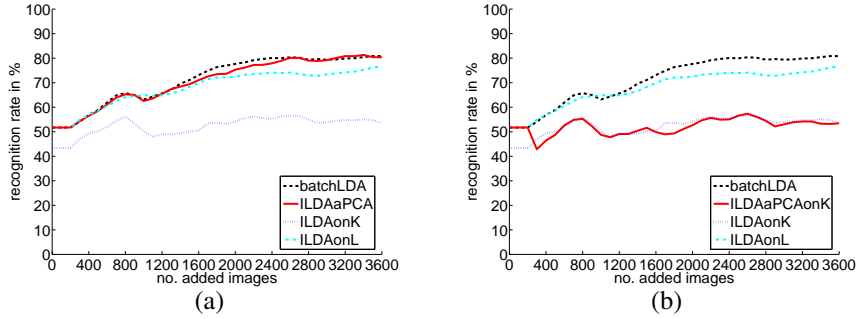


Figure 4: Update of the Coil-100 aPCA subspace: (a)  $\mathbf{U}^{(n+1)} = \mathbf{U}\mathbf{M}$ ; (b)  $\mathbf{U}^{(n+1)} = \mathbf{U}_{k+c-1}$ .

are shown in Figure 5(a) and Figure 5(b), respectively. It can be seen that due to the increasing complexity of the task (larger number of classes) the recognition rate drops for all methods even for *batchLDA*. But *ILDAaPCA* outperforms the other incremental methods.

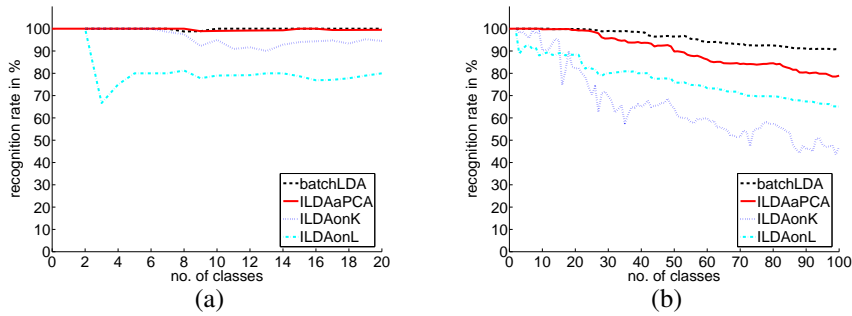


Figure 5: Recognition rates when adding new classes: (a) SFD and (b) ALOI-100.

Since for batch learning all training images are given and the basis vectors are not truncated the full information is available. Thus, it finds the most suitable hyperplanes and *batchLDA* always yields the best recognition rates. Updating the low-dimensional discriminative representations directly (*ILDAonL*) is not capable to adopt to a new task since it focuses only on initial given discriminative features. Conversely, by updating only the reconstructive representation (*ILDAonK*) too much discriminative information is lost. But the combination of both which is realized by *ILDAaPCA* is well suited for the task of incremental learning. Hence, the classification results are close to those of the batch method while *ILDAonL* and *ILDAonK* yield inferior classification results.

**Time and Memory:** As shown previously the recognition rate of the proposed incremental update scheme is comparable to batch learning. Thus, we can estimate an appropriate subspace even if not all images are given in advance. But incremental methods are more efficient which is shown in the following.



First, we compare the computational costs of *batchLDA* and *ILDAAaPCA*. Figure 6(a) shows the computation time<sup>2</sup> needed for the update from  $n$  to  $n + 1$  images (evaluated for COIL-100). Since *ILDAAaPCA* encompasses a model of fixed size only the representation matrix is growing. Thus, the computational costs for the updates are small. On the other hand, for *batchLDA* the representation has to be build from scratch for each additional image. Thus, the computation time is obviously increasing with each additional image.

Regarding memory requirements *ILDAAaPCA* has to store only  $\mathbf{U}$ ,  $\mathbf{A}$ ,  $\mathbf{V}$ , and  $\mu$  while *batchLDA* needs to store the full data matrix  $\mathbf{X}$ . Figure 6(b) illustrates the allocated memory in MB. It can be seen, that the memory requirements for *ILDAAaPCA* are only slightly increasing since only the size of the lower-dimensional representations is growing. For *batchLDA* considerable more memory is allocated if new images are added. As a result for larger data sets batch computation is not possible while the incremental update always is applicable.

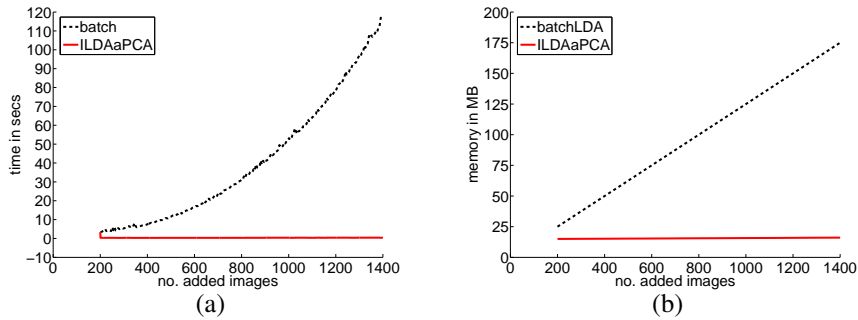


Figure 6: Requirements for adding images: (a) time and (b) memory.

## 4 Conclusion

We proposed a method capable to incrementally update an existing LDA subspace. This is achieved by additionally incorporating reconstructive information to the discriminative model. The reconstructive property ensures an adequate representation of the training images and allows the discriminative information to focus on new features whenever it is necessary. Hence, our method is capable to add new instances of already known classes as well as adding unknown classes. The experiments have shown that our method outperforms other tested incremental subspace methods that keep only discriminative or reconstructive information. Thus, these methods are not able to adapt to new data, especially when processing large datasets. Furthermore, our method yields similar results as the batch approach but it is more efficient considering the computational costs and memory requirements. Further work will include the evaluation of the recognition rate if the influence of the reconstructive basis is growing with the number of represented classes. In addition, since the reconstructive representation also allows detection of outliers the proposed method can easily be extended in a robust way.

<sup>2</sup>These results were obtained on a PC with a 2xDual-Core Intel Xeon 5160, 3.0 GHz CPU and 16GB RAM.

## References

- [1] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *PAMI*, 19(7):711–720, 1997.
- [2] S. Fidler, D. Skočaj, and A. Leonardis. Combining Reconstructive and Discriminative Subspace Methods for Robust Classification and Regression by Subsampling. *PAMI*, 28(3):337–350, 2006.
- [3] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.
- [4] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders. The Amsterdam Library of Object Images. *Computer Vision*, 61(1):103–112, 2005.
- [5] D. B. Graham and N. M. Allinson. *Face recognition: From theory to applications*, volume 163 of *NATO ASI Series F, Computer and Systems Sciences*, chapter Characterizing virtual eigensignatures for general purpose face recognition, pages 446–456. Springer, 1998.
- [6] K. Hiraoka, S. Yoshizawa, K. Hidai, M. Hamahira, H. Mizoguchi, and T. Mishima. Convergence analysis of online linear discriminant analysis. In *Proc. Int. Joint Conf. on Neural Networks*, volume 3, pages 387–391, 2000.
- [7] Alex D. Holub, Max Welling, and Pietro Perona. Combining generative models and fisher kernels for object recognition. In *Proc. IEEE Intern. Conf. on Computer Vision*, volume I, pages 136–143, 2005.
- [8] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Educational Psychology*, 24:417–441, 1933.
- [9] Yi Li, Linda G. Shapiro, and Jeff A. Bilmes. A generative/discriminative learning algorithm for image classifications. In *Proc. IEEE Intern. Conf. on Computer Vision*, volume II, pages 1605–1612, 2005.
- [10] Ruei-Sung Lin, David Ross, Jongwoo Lim, and Ming-Hsuan Yang. Adaptive discriminative generative model and its applications. In *Proc. Conf. on Neural Information Processing Systems*, pages 801–808, 2005.
- [11] X. Lu, Y. Wang, and A.K. Jain. Combining classifiers for face recognition. In *Proc. Int. Conf. on Multimedia and Expo*, volume 3, pages 13–16, 2003.
- [12] G. L. Marcialis and F. Roli. Fusion of PCA and LDA for face verification. In *Proc. Workshop on Biometric Authentication*, pages 30–37, 2002.
- [13] A. M. Martínez and A. C. Kak. PCA versus LDA. *PAMI*, 23(2):228–233, 2001.
- [14] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (COIL-100). Technical Report CUCS-006-96, Department of Computer Science, Columbia University, 1996.
- [15] S. Pang, S. Ozawa, and N. Kasabov. Incremental linear discriminant analysis for classification of data streams. *Systems, Man and Cybernetics, Part B*, 35(5):905–914, 2005.
- [16] D. Skočaj and A. Leonardis. Weighted and robust incremental method for subspace learning. In *Proc. ICCV*, volume 2, pages 1494–1501, 2003.
- [17] J. Yang and J.-Y. Yang. Why can LDA be performed in PCA transformed space? *Pattern Recognition*, 36:563–566, 2003.
- [18] J. Ye, H. Xiong Q. Li, H. Park, R. Janardan, and V. Kumar. IDR/QR: An Incremental Dimension Reduction Algorithm via QR Decomposition. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 364–373, 2004.