Pattern Anal Applic (2008) 11:259–268
DOI 10.1007/s10044-008-0131-0

THEORETICAL ADVANCES

Online nonparametric discriminant analysis for incremental subspace learning and recognition

B. Raducanu · J. Vitrià

Received: 15 December 2006 / Accepted: 20 January 2008 / Published online: 24 July 2008

Abstract This paper presents a novel approach for online subspace learning based on an incremental version of the nonparametric discriminant analysis (NDA). For many real-world applications (like the study of visual processes, for instance) it is impossible to know beforehand the number of total classes or the exact number of instances per class. This motivated us to propose a new algorithm, in which new samples can be added asynchronously, at different time stamps, as soon as they become available. The proposed technique for NDA-eigenspace representation has been used in pattern recognition applications, where classification of data has been performed based on the nearest neighbor rule. Extensive experiments have been carried out both in terms of classification accuracy and execution time. On the one hand, the results show that the Incremental NDA converges towards the classical NDA at the end of the learning process and furthermore. On the other hand, Incremental NDA is suitable to update a large knowledge representation eigenspace in real-time. Finally, the use of our method on a real-world application is presented.

Keywords Nonparametric discriminant analysis · Subspace learning · Nearest neighbor classifier · Pattern recognition

B. Raducanu (✉) · J. Vitrià
Computer Vision Center, Edifici "O",
Campus UAB, 08193 Bellaterra,
Barcelona, Spain
e-mail: bogdan@cvc.uab.es

J. Vitrià
Department of Applied Mathematics and Analysis,
University of Barcelona, 08003 Barcelona, Spain

1 Introduction

Feature extraction and selection is a common pre-processing step in any pattern classification problem. The outcome of this process consists of obtaining either an efficient data representation (through dimensionality reduction, when class labels are ignored) or an effective data discrimination (when besides the dimensionality reduction, we are focused also on class labels) [13]. For the latter, parametric and nonparametric forms have been proposed [5]. One of the most popular techniques is the linear discriminant analysis (LDA) (also known as Fischer discriminant analysis [8, 9]). It has been successfully applied for classification problems such as face recognition [6, 20, 25], face authentication [11] or mobile robotics [23]. The shortcomings of parametric discriminant analysis (PDA) are twofold. On the one hand, it assumes that the samples present a specific distribution (such as normal distribution for the case of LDA). On the other hand, and also due to the former restriction, it fails to capture the boundary class information. Because of these limitations, methods based on parametric discriminant analysis show a serious performance degeneration in real-world applications when data present multi-modal densities and classes are not linearly separable.

Opposed to this case, nonparametric discriminant analysis [9] is more effective when dealing with general data distributions and it captures properly the structural information between class boundaries. Despite its undeniable advantages, the nonparametric case has received little attention within pattern recognition community. In [3], the authors introduced a nonparametric form of the within-class scatter matrix. This way, the matrix is normalized: instead of assuming a gaussian distribution on the points of the same class, it normalizes the distances between each point and their nearest neighbors, which has been shown to benefit the nearest neighbor rule. In [15], simple (1D) projections are combined such that together can provide more separability on the whole data set. The nonparametric nature of their approach is guaranteed by the fact that it does not require any global statistic measure of the input data. In [12], two kinds of nonparametric subspace analysis, which complement each other, are proposed. First of them is the principal nonparametric subspace analysis and is used to extract nonparametric discriminating features within the principal subspace of within-class scatter matrix. The second one is called null-space nonparametric subspace analysis and is based on the null space of the within-class scatter matrix.

Typical implementations for the above-mentioned techniques assume that all the data is provided in advance and learning is carried out in one step (for this reason, we will refer as these as batch techniques). However, in real-world applications, this is not the case, since it is unlikely that all the data is available from the very beginning. For this situation, a new learning strategy is required. One-pass incremental algorithms, which performs sequentially the update of the eigenspace representation, are the solution we are seeking. So far, several approaches have been proposed. In [1, 4, 10] the incremental principal component analysis (IPCA) is introduced. The update of the covariance matrix is achieved through a residual procedure. They keep only the learned coefficients of the eigenspace and discard the original data. In order to avoid the excessive growth of the eigenspace dimensionality a new dimension is added only when a relevant improvement in reconstruction is detected. A completely different approach was taken in [24]. They demonstrate that is possible to incrementally build an eigenspace representation without the need to compute the covariance matrix at all. On the other hand, some incremental versions of ILDA have also been proposed. In [7, 19], they embed LDA learning and classification into the incremental PCA framework. The combined subspace consists of a truncated PCA subspace and a few additional basis vectors that encompass the discriminative information. As such it contains both sufficient reconstructive information to enable incremental learning, and the previously extracted discriminative information to enable efficient classification as well. In [18], an ILDA deriving discriminant eigenspace in a streaming environment without updating the eigen-decomposition is proposed. In change, they build the discriminant eigenspace in terms of the incremental updating of the between-class and within-class scatter matrices.

In this paper, we propose an incremental version for NDA technique (referred for the rest of the paper as IncNDA). More concrete, we introduce a sequential update of the NDA-eigenspace representation. We start the procedure with at least two classes, and the rest of the data (representing both new classes or new instances of the existing classes) is added incrementally. To test the efficiency, extensive experiments were carried out on some datasets of the UCI database [2] as well as on some public face databases: ORL-ATT [16], UMIST [21] and AR [14]. In terms of classification accuracy, we prove that our approach converges (at the end of the learning process) towards the classical NDA (referred from now on as BatchNDA). In terms of computational complexity, we show that IncNDA is suitable for real-time applications.

Our choice for NDA was motivated by the fact that being a nonparametric method, its application is not limited to Gaussian distributions of data. Another advantage provided by this method is that it extracts those features which work well with the nearest–neighbor classifier.

The paper is structured as follows: Sect. 2 contains a brief review of the BatchNDA technique. Section 3 presents the newly introduced incremental version of it. Section 4 contains some comparative experimental results between BatchNDA and IncNDA. In Sect. 5, we present the use of our approach on a real-world application. Finally, in Sect. 6, we will draw our conclusions and present future work directions.

2 Classical nonparametric discriminant analysis (BatchNDA)

As introduced in [9], the within-class scatter matrix S_w and between-class scatter matrix S_b are used as a measure of inter-class separability. One of the most used criteria is the one that maximize the following expression:

$$\zeta = \mathrm{tr}(S_\mathrm{w}^{-1} S_\mathrm{b}) \tag{1}$$

It has been shown that the $M \times D$ linear transform that satisfies the Eq. 2 optimizes also the separability measure ζ:

$$\hat{W} = \arg \max_{W^T S_\mathrm{w} W = I} \mathrm{tr}(W^T S_\mathrm{b} W) \tag{2}$$

This problem has an analytical solution and is mathematically equivalent to the eigenvectors of the matrix $S_\mathrm{w}^{-1} S_\mathrm{b}$.

Let us assume that the data samples we have belong to N classes C_i, $i = 1,2, \ldots ,N$. Each class C_i is formed by n_i samples $C_i = \{x_1^i, x_2^i, \ldots ,x_{n\_C\_i}^i\}$. By $\bar{x}^{C_i}$ we will refer to the mean vector of class C_i. According to [9], the S_w and S_b scatter matrices are defined as follows:

$$S_\mathrm{w} = \sum_{i=1}^{C_N} \sum_{j \in C_i} \left(x_j - \bar{x}^{C_i}\right)\left(x_j - \bar{x}^{C_i}\right)^T \tag{3}$$

$$S_b = \sum_{i=1}^{C_N} \sum_{j=1, j \neq i}^{C_N} \sum_{t=1}^{n_{C_i}} W(C_i, C_j, t)(x_t^i - \mu_{C_j}(x_t^i))(x_t^i - \mu_{C_j}(x_t^i))^T \tag{4}$$

where $\mu_{C\_j}(x_t^i)$ is the local $K\!-\!NN$ mean, defined by:

$$\mu_{C_j}(x_t^i) = \frac{1}{k} \sum_{p=1}^{k} NN_p(x_t^i, C_j) \tag{5}$$

where $NN_p(x_t^i, C_j)$ is the pth nearest neighbor from vector (x_t^i) to the class C_j. The term $W(C_i, C_j, t)$ which appears in Eq. 4 is a weighting function whose role is to emphasize the boundary class information. It is defined by the following relation:

$$W(C_i, C_j, t) = \frac{\min\{d^\alpha(x_t^i, NN_k(x_t^i, C_i)), (x_t^i, NN_k(x_t^i, C_j))\}}{d^\alpha(x_t^i, NN_k(x_t^i, C_i)) + d^\alpha(x_t^i, NN_k(x_t^i, C_j))} \tag{6}$$

Here α is a control parameter that can be selected between zero and infinity. The sample weights take values close to 0.5 on class boundaries and drop to zero as we move away. The parameter α adjusts how fast this happens.

3 Incremental nonparametric discriminant analysis (IncNDA)

The shortcoming of the BatchNDA described in the previous section, is that assumes that all the data are available at the classification. This is not the case for real applications, when the data is coming over time, at random time intervals, and the representation of the data must be updated. Computing from the beginning the scatter matrices, each time a new sample arrives, is not computationally feasible, especially when the number of classes is very high and the number of samples per class increases significantly. For this reason, we propose the IncNDA technique, that can process sequentially later-on added samples, without the need for recalculating entirely the scatter matrices. In order to describe the proposed algorithm, we assume that we have computed the S_w and S_b scatter matrix from at least two classes. Let us now consider that a new training pattern y is presented to the algorithm. We distinguish between two situations.

3.1 The new training pattern belongs to an existing class

This situation is depicted in Fig. 1. Let us assume, for instance, that the new pattern y (represented by the solid black star) belongs to one of the existing classes C_L (i.e. y^{C_L}, where $1 < L < N$). The links from the figure represent the old and new situations, with the classes before and after

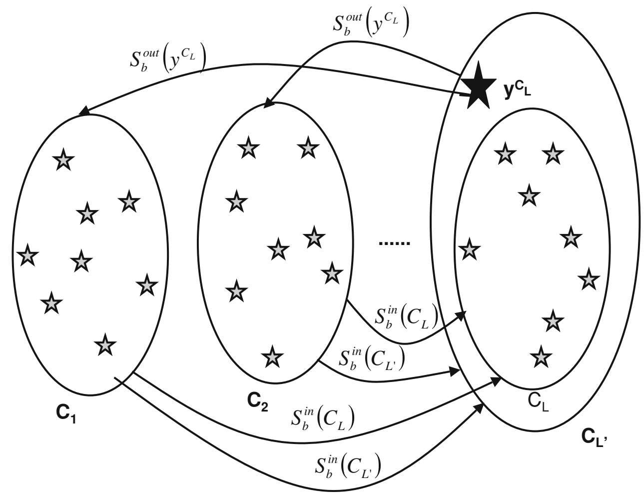

Fig. 1 The new added pattern belongs to an existing class

the introduction of the new element. At the same, they also indicate only those classes from the global S_b matrix that will be affected by the update equations.

After the introduction of the new pattern, the formula used to recursively calculate the S_b matrix is given by:

$$S_b' = S_b - S_b^{in}(C_L) + S_b^{in}(C_{L'}) + S_b^{out}(y^{C_L}) \tag{7}$$

where $C_{L'} = C_L \bigcup \{y^{C_L}\}$, $S_b^{in}(C_L)$ represents the covariance matrix between the existing classes and the class that is about to be changed, $S_b^{in}(C_{L'})$ represents the covariance matrix between existing classes and the updated class $C_{L'}$ and by $S_b^{out}(y^{C_L})$ we denote the covariance matrix between the vector y^{C_L} and the other classes. The equations to compute these matrices are given by the following formulas:

$$S_b^{in}(C_L) = \sum_{j=1, j \neq L}^{C_N} \sum_{i=1}^{n_{C_j}} W(C_j, C_L, i)(x_i^j - \mu_{C_L}(x_i^j)) (x_i^j - \mu_{C_L}(x_i^j))^T \tag{8}$$

$$S_b^{out}(y^{C_L}) = \sum_{j=1, j \neq L}^{C_N} (y^{C_L} - \mu_{C_j}(y^{C_L}))(y^{C_L} - \mu_{C_j}(y^{C_L}))^T \tag{9}$$

In the case of S_w' the update equation is the following:

$$S_w' = \sum_{j=1, j \neq L}^{C_N} S_w(C_j) + S_w(C_{L'}) \tag{10}$$

where

$$S_w(C_{L'}) = S_w(C_L) + \frac{n_{C_L}}{n_{C_L} + 1}(y - \bar{x}^{C_L})(y - \bar{x}^{C_L})^T \tag{11}$$

3.2 The new training pattern belongs to a new class

This situation is depicted in Fig. 2. Let us assume that new pattern y (again, represented by the solid black star)

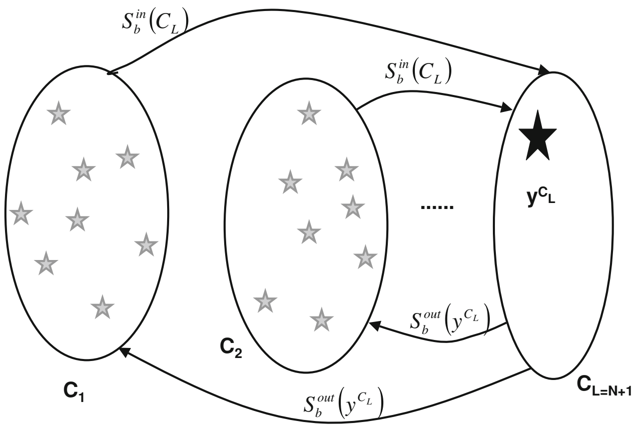

Fig. 2 The new added pattern belongs to a new class

belongs to a new class C_{N+1} (i.e. $y^{C_{N+1}}$). Exactly as in the previous case, the links indicate only those classes from the global S_b matrix that will be affected by the update equations.

After the introduction of the new pattern, the formula used to recursively calculate the S_b matrix is given by:

$$S_b' = S_b + S_b^{out}(C_{N+1}) + S_b^{in}(C_{N+1}) \tag{12}$$

where $S_b^{out}(C_{N+1})$ and $S_b^{in}(C_{N+1})$ are defined as follows:

$$S_b^{out}(C_{N+1}) = \sum_{j=1}^{C_N} (y^{C_{N+1}} - \mu_{C_j}(y^{C_{N+1}}))(y^{C_{N+1}} - \mu_{C_j}(y^{C_{N+1}}))^T \tag{13}$$

$$S_b^{in}(C_{N+1}) = \sum_{j=1}^{C_N} \sum_{i=1}^{n_{C_j}} W(C_j, C_{N+1}, i)(x_i^j - \mu_{C_{N+1}}(x_i^j))(x_i^j - \mu_{C_{N+1}}(x_i^j))^T$$

Regarding, the new S_w' matrix, this one remains unchanged, i.e:

$$S_w' = S_w \tag{14}$$

4 Experimental results

To test the efficiency of IncNDA, we applied the learned eigenspace representation for the pattern classification problem. The training samples presented up to a given moment are encoded through their projection on the NDA-eigenspace. When a test sample arrives, its identity is claimed based on the nearest–neighbor classifier: we seek the closest neighbor (in terms of Euclidean distance) of the projected test sample on the IncNDA eigenspace. As a measure of classification accuracy, we compare the class label of the found neighbor with the class label of the test sample. Extensive experiments in terms of classification accuracy have been carried out on some datasets from the UCI database and on some public face databases, such as the ORL-ATT, UMIST and AR. On the face databases, given the high-dimensionality of the data and the large number of classes, we also tested the computation complexity (in terms of execution time).

As a general framework for our experiments, we adopted the so-called 'tenfold cross-validation' procedure, a very popular technique in the pattern recognition literature. We used 90% of the data for training and the rest for test (as the name suggests, this is repeated ten times with data selected randomly). On its turn, the training dataset was also divided in two parts: usually 15–20% of the data (representing two or more classes) are used to initialize the IncNDA[1]. For each experiment run, we calculated also the 95% confidence interval (that is why the data is represented as mean ± variance) according to the Eq. 15.

$$\text{var} = 1.96 \times \frac{\text{std(recog\_rate)}}{\sqrt{\text{max\_iter}}} \tag{15}$$

where 'std' stands for the *standard deviation* of the recognition rate vector and max_iter = 10.

4.1 UCI database

The IncNDA was tested on some subsets (see Table 1) from the UCI Machine Learning Repository. The idea of the experiments is to show that the performance of IncNDA is similar to the one achieved using BatchNDA both in case of datasets with a small number classes and a few samples per class as well as with a great number of classes and many samples per class. A description of the used datasets is given in Table 1. The column 'NDA dim' refers to the dimensionality of the NDA eigenspace used in our experiments.

Table 1 Overview of evaluated UCI datasets: S1 = sonar, S2 = liver, S3 = iris, S4 = wine, S5 = vehicle, S6 = glass, S7 = segmentation, S8 = vowel

Idx	No. classes	Original dim	NDA dim	No. samples
S1	2	60	6	208
S2	2	6	3	345
S3	3	4	2	150
S4	3	13	7	178
S5	4	18	9	846
S6	6	9	6	214
S7	7	19	6	2,310
S8	11	10	7	528

[1] In order to overcome the singularity problem, a PCA has been applied beforehand.

The rest of the samples are added randomly, in a sequential order. The IncNDA eigenspace is updated after each sample is presented. We compared the results obtained using the IncNDA (in terms of classification accuracy and variance) with the ones obtained using BatchNDA. The results shown in Table 2 represent the average recognition rate and the confidence interval. We repeated the experiments, considering different number of neighbors in computing the Eq. 4: 1, 3, 5 and 7. In case of IncNDA algorithm, mean and variance value correspond to the last 'learning stage'. It can be appreciated that in all cases, these values, are very close to the ones corresponding to the BatchNDA, which demonstrates the convergence of our algorithm.

The best results in Table 2 are marked in bold. It shows that for datasets with a small number of classes or small number of samples (especially sets S1–S6), better classification results are obtained when using only one neighbor in computing the Eq. 4. It is the case that for these classes we also have a larger confidence interval. On the other hand, when we have datasets with more classes and more samples per class (dataset S7–S8), more stable results (also in terms of a reduced confidence interval) are achieved when we use more neighbors (five and seven, respectively) in computing the Eq. 4.

4.2 Face databases

We also tested the performance of our algorithm on some face databases: ORL-ATT, UMIST and AR[2]. The decision for choosing these databases was motivated by the fact that they complement very well with each other in the sense they show a wide variety of conditions: ORL-ATT contains frontal or slightly tilted face images, under small variations in illumination conditions; UMIST contains face images taken in very different head poses (ranging from frontal to profile), but with constant illumination; and finally, AR database which shows frontal face images under very significant changes in illumination, facial expression and occlusions. The characteristics of these databases are summarized in Table 3[3].

The recognition rates are presented in Table 4. Exactly as in the case of UCI datasets, we repeated the experiments considering different numbers of neighbors (1, 3, 5, 7) when computing the Eq. 4. The best results are marked in bold characters. It can be noticed that while the

[2] For this experiment we selected 85 classes from the total of 116, because these classes correspond to people who took part in both photographic sessions. We also removed those images showing occlusions.

[3] The values in the 'face size' column represent the actual values used in our experiments obtained after downsampling and cropping the face area from the original images.

Table 2 Comparison between IncNDA and BatchNDA in terms of classification accuracy on eight UCI datasets: S1 = sonar, S2 = liver, S3 = iris, S4 = wine, S5 = vehicle, S6 = glass, S7 = segmentation, S8 = vowel

Idx	NN-1		NN-3		NN-5		NN-7	
	IncNDA	BatchNDA	IncNDA	BatchNDA	IncNDA	BatchNDA	IncNDA	BatchNDA
S1	83.0357 ±2.3752	**83.0357 ±2.2640**	78.3929 ±2.3534	78.3929 ±2.3534	80.5357 ±2.8491	80.7143 ± 2.7063	80.1786 ± 2.5632	79.6429 ± 2.5442
S2	60.3333 ± 3.0354	**60.5556 ± 3.1579**	58.4444 ± 3.6455	58.3333 ± 3.5732	59.1111 ± 3.0863	59.1111 ± 3.0863	58.2222 ± 2.3773	58.0000 ± 2.3199
S3	**96.6667 ± 2.2232**	**96.6667 ± 2.2232**	96.0000 ± 1.9884	96.0000 ± 1.9884	96.0000 ± 1.9884	96.0000 ± 1.9884	94.6667 ± 2.7800	94.6667 ± 2.7800
S4	**96.4286 ± 1.6059**	96.2500 ± 1.6436	94.1071 ± 1.9848	94.1071 ± 1.9848	93.5714 ± 1.4000	93.7500 ±1.4251	94.6429 ± 1.7955	94.4643 ± 1.8642
S5	**68.2292 ± 1.5640**	68.2292 ± 1.5780	66.8750 ± 2.0124	66.7708 ± 2.0195	67.2917 ± 2.0758	67.3958 ± 2.1571	67.0833 ± 2.2725	67.0833 ± 2.2336
S6	71.1765 ± 2.6974	**71.9118 ± 2.9417**	68.5294 ± 2.8702	68.3824 ± 2.9535	68.0882 ± 3.8620	68.3824 ± 3.5456	69.7059 ± 3.6959	69.8529 ± 3.8073
S7	95.1774 ± 0.4666	95.1774 ± 0.4666	94.8871 ± 0.4570	94.9032 ± 0.4532	95.3871 ± 0.4256	95.3871 ± 0.4256	**95.7097 ± 0.4131**	95.6935 ± 0.4211
S8	98.3333 ± 0.5794	97.1154 ± 0.8908	98.0128 ± 0.8448	97.1154 ± 0.7263	**98.9103 ± 0.6888**	97.5641 ± 0.7713	98.2051 ± 0.5284	97.3077 ± 0.6289

Table 3 Overview of the face databases: ORL-ATT, UMIST, AR

Idx	No. classes	Face size (pixels)	NDA dim	No. samples
ORL-ATT	40	56 × 46	100	400
UMIST	20	56 × 46	100	564
AR	85	36 × 33	100	1,190

classification accuracy remains unchanged for the first two databases, for AR database in change, the classification accuracy increases with the number of nearest neighbors used to calculate the Eq. 4.

Additionally, we also give a graphical interpretation of some aspects related to the incremental learning process. For this purpose, we introduce the term ‘learning stage’ to refer to the number of samples from the training set that have been added up to a certain moment. In Fig. 3 above we depicted the evolution of the learning process after each update (a new sample added) of the initial IncNDA eigenspace. In Fig. 3 below, we depicted the percentage of incremental training samples introduced so far (the stars represent the moment when a new class has been added). The graph falls a couple of times, because the percentage of total data is actually computed relative to the number of classes presented up to a given moment, not with the total number of classes. For this reason, this graphic should be read in concordance with the one above. The plots correspond to the ORL-ATT face database.

In Fig. 4, we show that indeed the IncNDA is converging (at the end of the learning process) towards BatchNDA. Both graphics were plotted after averaging the results obtained from a tenfold cross-validation procedure (the training samples were randomly chosen in each run). The oscillation of the IncNDA has two explanations: new classes are presented one after the other, at short intervals and there are few instances available for each class. The plot corresponds also to the ORL-ATT face database.

In terms of computational complexity, the most ‘critical’ aspect (time consuming) is represented by the calculation of the S_b matrix (this, on its turn, is related to finding the nearest neighbor). In order to show the efficiency of our proposed technique, we measured the time needed to update S_b and compared it with time needed by the classical NDA to recalculate S_b each time from the beginning. We sequentially updated the S_b matrix with about 2,200 samples (presented randomly) of dimensionality 100 (corresponding to the projection of an image on the NDA eigenspace). The samples corresponding to the AR face database are divided in 85 classes[4]. The results are

Table 4 Comparison between IncNDA and BatchNDA in terms of classification accuracy on some face databases

Idx	NN-1		NN-3		NN-5		NN-7	
	IncNDA	BatchNDA	IncNDA	BatchNDA	IncNDA	BatchNDA	IncNDA	BatchNDA
ORL-ATT	96.5000 ± 1.1682	96.6000 ± 1.1760	97.1000 ± 0.7237	97.1000 ± 0.7237	**97.2000 ± 0.7194**	**97.1000 ± 0.7775**	96.7000 ± 0.9964	96.7000 ± 0.9116
UMIST	99.3333 ± 0.5844	99.4667 ± 0.5778	99.3333 ± 0.5844	99.3333 ± 0.7023	99.0667 ± 0.3992	99.3333 ± 0.4356	**99.3333 ± 0.4356**	**99.3333 ± 0.4356**
AR	77.4211 ± 1.4625	77.4211 ± 1.5489	79.8947 ± 1.1691	80.0000 ± 1.2302	79.3684 ± 1.8695	79.3158 ± 1.8584	**82.3684 ± 1.6933**	**82.3158 ± 1.7069**

[4] For this experiment, in order to have more data, we considered also the images affected by occlusions.

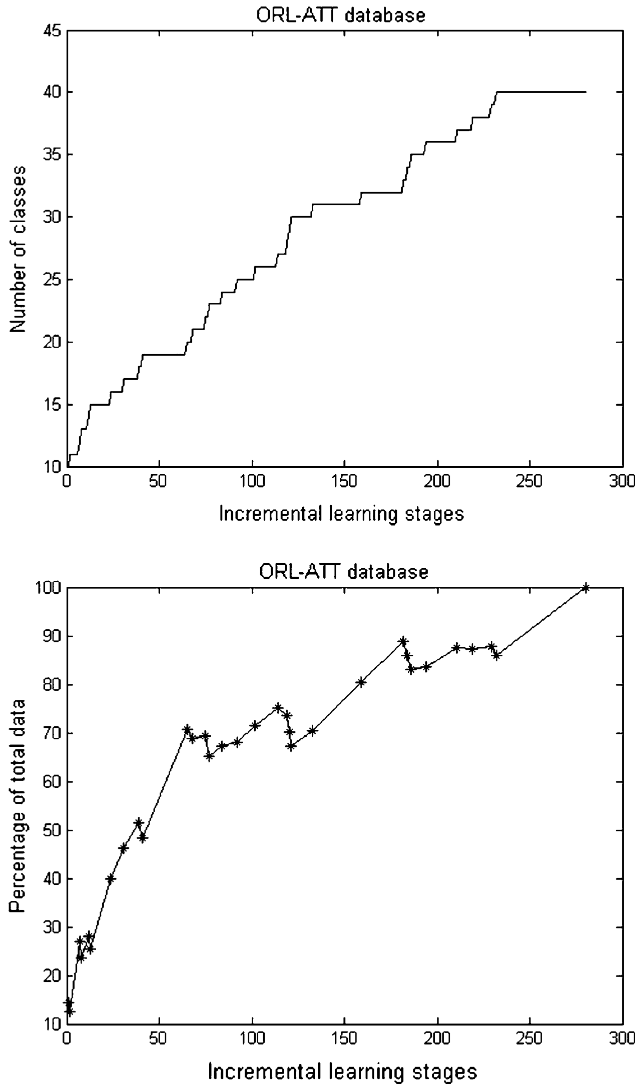

Fig. 3 Learning process: evolution of the number of classes function of learning stages (*above*) and the percentage of the training data function of learning stages (*below*)

presented in Table 5. The values represent ‘seconds’ and were obtained considering one neighbor in calculating the S_b matrix (according to Eq. 4). Due to very significative differences between the results of the two methods (two orders of magnitude), a graphical representation would not have been informative. For visualization purposes, we present only the time computed every 200 samples. The experiment was done in MATLAB on a 3 GHz Pentium IV computer, with 1 GB of memory.

From the Table 5, it can be appreciated that the time needed to update the S_b matrix using the incremental approach is sensibly much lower than using the classical one. In summary, this makes our approach suitable for applications which need real-time update (for instance, for classification from video or live streams).

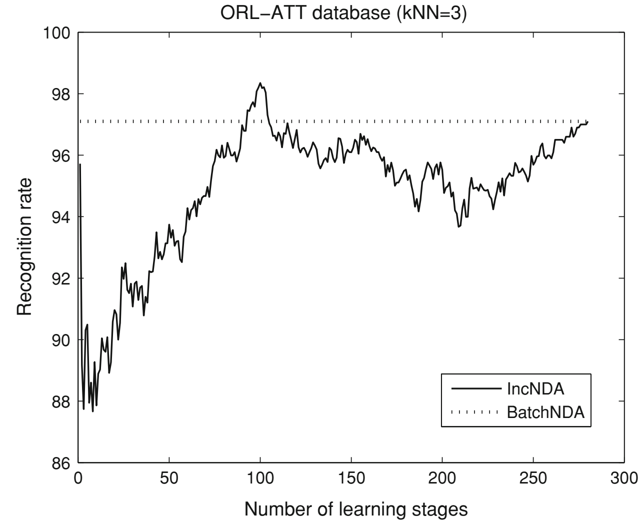

Fig. 4 IncNDA versus BatchNDA curves. IncNDA converges towards BatchNDA at the end of the learning process

4.3 Discussion

From the experimental results presented so far, it can be shown that incremental techniques are approximate ones, converging in the end towards the global solution given by their classical counterparts (this is also underlined in all the papers mentioned in Sect. 1). For instance, in the case of incremental PCA, based on the incremental update of the eigenvectors, a very rigorous mathematical proof about the convergence is given in [17]. For IncNDA, the reason for obtaining an approximate result can be explained by the order in which the samples are introduced. This factor, on its turn, has impact on the estimation of the nearest neighbor(s), which in the end is reflected in the computation of S_b matrix.

Another aspect that is related with the algorithm performance is represented by the memory usage. The memory usage itself should not be a problem (remember that we work with the projected version of our data, not with the original one), but the number of samples and the number of classes affects the execution time needed to update the NDA eigenspace. However, the experiments presented in this paper are limited and such a question does not represent a critical issue. It would be very interesting to consider the case of much larger datasets (with thousands of samples per class). With our incremental approach we can always keep an optimal number of samples per class and this way we can guarantee the real-time character of IncNDA. When the number of samples increases considerably, we can figure out some criteria upon which we can fix an upper-limit to them, without affecting the system's performance. This implies to replace some old samples by new ones. Such criteria would require to make some

Table 5 Comparison between IncNDA and BatchNDA in terms of computational complexity: execution time tocalculate the S_b matrix (in seconds)

Method	Number of samples										
	200	400	600	800	1,000	1,200	1,400	1,600	1,800	2,000	2,200
BatchNDA	0.46	1.51	3.17	4.53	6.57	8.54	10.65	13.03	16.09	18.73	22.14
IncNDA	0.03	0.04	0.07	0.09	0.12	0.15	0.17	0.20	0.21	0.26	0.29

modifications to our algorithm. Thus, we could think of a 'weight vector' associated with the data which would keep information about the time-stamp of the data was presented and how many times a sample has been selected by the nearest–neighbor rule. In this way, the criteria referred above could be stated as removing the oldest elements or removing the less relevant ones. These aspects are currently under investigation and will be the object of future work.

5 Face recognition in a real scenario

Next, we will discuss the results obtained applying IncNDA to a custom face database. The image acquisition phase was extended over several weeks and was performed in an automatic manner. For this purpose, we put the camera in an open space and snapshots were taken each time a person was passing in front of it. The face was automatically extracted from the image using the face detector based on [22]. We did not impose any restrictions regarding ambient conditions. Figure 5 shows a snapshot of the experimental setup.

Fig. 5 Snapshot of the experimental setup

Overall, our database consists of 6,882 images of 51 people (both male and female)[5]. Since no arrangements were previously made, some classes contain only a handful of images (as much as 20), meanwhile, the largest of them contains over 400. Segmented faces were normalized at a standard size of 48 $\times$ 48 pixels. Because of the particular acquisition process, face images reflect the changes in appearance suffered by subjects over time. Furthermore, since our application was thought to run in real-time (and to give it a more ad hoc flavor), we did not perform any pre-processing step to the face images before passing them to the classifier. That is why the faces used in the experiment show a certain degree of variation in pose and size and are not constrained to be exactly frontal. For the same reason, face images used to be a little wider than the face region itself. Some samples of these face images are presented in Fig. 6.

To test the IncNDA technique, we used 90% of the images (i.e. about 6,000) as training set and the remaining ones as test set. From the training set, we used 15% of the images (belonging to five classes and representing 900 samples) to build the initial IncNDA eigenspace. In order to overcome the singularity problem, a PCA step was performed beforehand. This way, data dimensionality was downsized from 2,304 to 60. The remaining samples (5,100) from the training set were added later in a sequential order (the samples were drawn randomly) and this way the NDA-eigenspace was updated.

In Fig. 7, we show that indeed the IncNDA is converging (at the end of the learning process) towards BatchNDA. The common recognition rate achieved is around 95%, which in our opinion is a very good result, taking into account the difficulty of the database. Both graphics were plotted after averaging the results obtained from a tenfold cross-validation procedure (the training samples were chosen in a random manner in each run). We repeated the experiments considering different number of

[5] In the current study, we put the accent in having a reasonable number of classes with many instances rather having an excessive number of classes with very few instances.

Fig. 6 Samples of face images from CVC custom database showing a certain degree of variation in illumination, pose and size

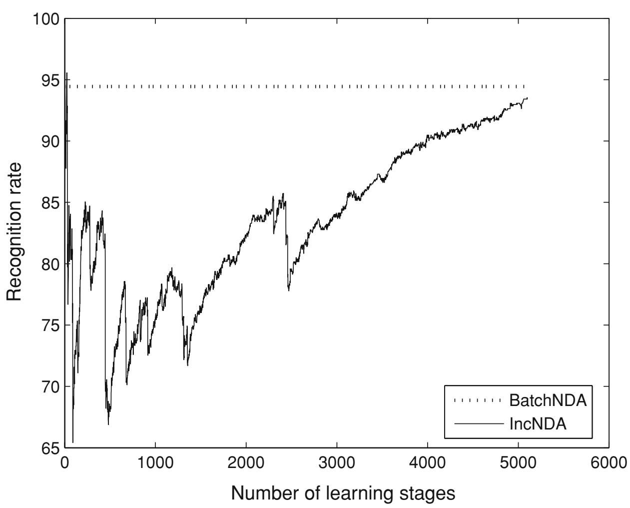

Fig. 7 IncNDA versus BatchNDA curves. IncNDA converges towards BatchNDA at the end of the learning process

neighbors (1, 3, 5, 7) when computing the Eq. 4), but the best results obtained correspond to a number of neighbors equal to 3. The Fig. 7 corresponds to this case. The oscillation of the IncNDA in its early stages corresponds to the situation when a significant number of new classes have been added at very short intervals and only a very few samples of those classes were available. After some learning stages, when enough samples for each class became available, we can appreciate that the evolution curve regulates its tendency and becomes constantly ascending.

Fig. 8 Some instances of misclassified faces

Some instances of misclassified faces are represented in Fig. 8. From the experiments performed, we arrive at the conclusion that misclassification occurs in three situations: when there are too few face instances per class, when there are too few instances of a particular head pose/illumination conditions and when the image presents a high level of distortion (the 'blurring' effect due to person movement).

6 Conclusions and future work

For some real-world applications, one-step (batch mode) learning techniques prove to be inadequate. For this reason, we proposed in this paper an incremental version of the NDA. We start to build the NDA-eigenspace representation in an incremental way, by adding sequentially new data. This new approach has been applied to a classification problem based on the nearest–neighbor rule. Extensive experiments were performed on some datasets from the UCI database as well as on some public face databases: ORL-ATT, UMIST and AR. The experiments were intended to assess the classification accuracy (it converges towards BatchNDA) and computational complexity (it is able to update the NDA eigenspace in real-time) of the proposed method. Finally, we presented the use of our approach in a real-world scenario.

In the future, we plan to extend the current approach, by allowing the update of the NDA-eigenspace in terms of data chunk. Another research direction is represented by the analysis of decremental learning, which emulates the 'forgetting' process in humans: those patterns which became irrelevant are removed from the knowledge representation depending on some criteria are replaced with new ones. This is necessary in order to avoid an excessive increase of the data, which could affect negatively the real-time running of the algorithm.

Acknowledgments This work is supported by MCYT Grant TIN2006-15308-C02, Ministerio de Educación y Ciencia, Spain. Bogdan Raducanu is supported by the Ramon y Cajal research program, Ministerio de Educación y Ciencia, Spain.

References

1. Artač M, Jogan M, Leonardis A (2002) Incremental PCA for on-line visual learning and recognition. In: Proceedings of the 16th international conference on pattern recognition, vol 3. Québec, Canada, pp 781–784
2. Blake C, Merz C (1998) UCI repository of machine learning databases. http://www.ics.uci.edu/ mlearn/MLRepositor. html
3. Bressan M, Vitrià J (2003) Nonparametric discriminant analysis and nearest neighbor classification. Pattern Recognit Lett 24(15):2743–2749
4. Chandrasekaran S, Manjumath BS, Wang YF, Winkler J, Zhang H (1997) An eigenspace update algorithm for image analysis. Graph Models Image Process 59(5):321–332
5. Duda RO, Hart PE, Stork DG (2001) Pattern classification, 2nd edn. Wiley, New York
6. Etemad K, Chellapa R (1997) Discriminant analysis for recognition of human face images. J Opt Soc Am 14(8):1724–1733
7. Fidler S, Skočcaj D, Leonardis A (2006) Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling. IEEE Trans Pattern Anal Mach Intell 28(3):337–350
8. Fisher RA (1938) The statistical utilization of multiple measurements. Ann Eugen 8:376–386
9. Fukunaga K (1990) Introduction to statistical pattern recognition, 2nd edn. Academic Press, Boston
10. Hall P, Marshall D, Martin R (1998) Incremental eigenanalysis for classification. In: Proceedings of British machine vision conference, vol 1. Southampton, UK, pp 286–295
11. Kotropoulos C, Tefas A, Pitas I (2000) Frontal face authentication using morphological elastic graph matching. IEEE Trans Pattern Anal Mach Intell 9(4):555–560
12. Li Z, Liu W, Lin D, Tang X (2005) Nonparametric subspace analysis for face recognition. In: Proceedings of 2005 IEEE conference on computer vision and pattern recognition, vol 2. San Diego, USA, pp 961–966
13. Martinez AM, Kak AC (2001) PCA versus LDA. IEEE Trans Pattern Anal Mach Intell 23(2):228–233
14. Martinez AM, Benavente R (1998) The AR face database. Computer Vision Center. Technical report #24
15. Masip D, Vitrià J (2005) Boosted linear projections for discriminant analysis. In: Vitrià et al (eds) Frontiers in artificial intelligence and applications series. Recent advances in artificial intelligence: Research and Development. IOS Press, Amsterdam, pp 45–52
16. Samaria F, Harter A (1994) Parameterisation of a stochastic model for human face identification. In: Proceedings of second IEEE workshop on applications of computer vision, Sarasota
17. Oja E (1983) Subspace methods of pattern recognition. Research Studies Press, Letchworth
18. Pang S, Ozawa S, Kasabov N (2005) Chunk incremental LDA computing on data streams. In: Wang et al (eds) International symposium on neural networks, vol LNCS 3497. Chongqing, China, pp 51–56
19. Skočaj D, Uray M, Leonardis A, Bischof H (2006) Why to combine reconstructive and discriminative information for incremental subspace learning. In: Chum et al (eds) Proceedings of computer vision winter workshop, Telč, Czech Republic, pp N/A
20. Swets DL, Weng JJ (1996) Using discriminant eigenfeatures for image retrieval. IEEE Trans Pattern Anal Machine Intell 18(8):831–836
21. Graham DB, Allinson NM (1998) Characterizing virtual eigensignatures for general purpose face recognition. In: Wechsler et al (eds) Face recognition: from theory to applications, NATO ASI series F. Computer and systems sciences, vol 163, pp 446–456
22. Viola P, Jones MJ (2004) Robust real-time face detection. Int J Comp Vis 57:137–154
23. Weng JJ (1996) Crescepton and SHOSLIF: towards comprehensive visual learning. In: Nayar et al (eds) Early visual learning. Oxford University Press, Oxford, pp 183–214
24. Weng J, Zhang Y, Hwang W-S (2003) Candid covariance-free incremental principal component analysis. IEEE Trans Pattern Recognit 25(8):1034–1040
25. Zhou SK, Chellappa R (2004) Multiple-exemplar discriminant analysis for face recognition. In: Proceedings of IEEE conference on pattern recognition, Cambridge, UK, pp N/A