International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 22, No. 5 (2008) 899–913

World Scientific
www.worldscientific.com

FACE RECOGNITION BY ARTIFICIAL VISION SYSTEMS: A COGNITIVE PERSPECTIVE

BOGDAN RADUCANU* and JORDI VITRIÀ†

Computer Vision Center, Edifici "O" — Campus UAB
08193 Bellaterra (Barcelona), Spain

†*Department of Applied Mathematics and Analysis*
University of Barcelona
08003 Barcelona, Spain
**bogdan@cvc.uab.es*
†*jordi@cvc.uab.es*

Cognitive development refers to the ability of a system to gradually acquire knowledge through experiences during its existence. As a consequence, the learning strategy should be represented as an integrated, online process that aims to build a model of the "world" and a continuous update of this model. Considering as reference the Modal Model of Memory introduced by Atkinson and Schiffrin, we propose an online learning algorithm for cognitive systems design. The incremental part of the algorithm is responsible of updating existing information or creating new data categories and the decremental part, to efficiently evaluate the system's performance facing partial or total loss of data. The proposed algorithm has been applied to the face recognition problem. More generally, the current approach can be extended to large-scale classification problems, to limit the memory requirements for optimal data representation and storage.

Keywords: Cognitive development; online learning; pattern classification; face recognition.

1. Introduction

The human visual cognitive system is very robust across large range of variations in environmental conditions. In contrast to this, a similar robustness of visual cognition is still far from being achieved by artificial vision systems. Despite the progresses reported in areas like vision sensors, statistical pattern recognition and machine learning, what for humans represents a natural process, for machines is still a far-fetched dream. One of the factors that limit these performances is the learning strategy that has been used. Most of today's approaches require the intervention

*Author for correspondence

of a human operator to collect, store and segment hand-picked images and train pattern classifiers with them.[a] It is unlikely that such a manual operation could meet the demands of many challenging cognition tasks that are critical for generating intelligent behavior, such as object recognition, in general, and face recognition, in particular. In view of this, our desired goal is to enable machines to learn directly from sensory input streams while interacting with the environment, including humans. During the interaction, the human is not allowed to interfere in the internal state of the system.[3]

Although there are several approaches in the pattern recognition literature dealing with online learning techniques, most of them refer only to the first aspect, i.e. the incremental phase. In some of these papers,[1,5,9] the Incremental Principal Component Analysis (IPCA) is presented. The update of the covariance matrix is achieved through a residual procedure, keeping only the learned coefficients of the eigenspace representation and discarding the original data. In the context of IPCA, Weng *et al.*[18] demonstrated that it is possible to build incrementally an eigenspace representation without the need to compute the covariance matrix at all. Some other incremental versions of Linear Discriminant Analysis (ILDA) are proposed.[7,11,13] The second phase of these learning techniques (the decremental one) has received very limited attention, compared to its incremental counterpart, being developed with predilection for Support Vector Machines.[4,14] The incremental phase corresponds to knowledge acquisition (for biological systems it is associated with evolution processes). On the other hand, the decremental phase (also referred as "unlearning"), resembles the forgetful property of human mind, i.e. how the degradation in system's retrieval performance is affected by sequentially subtraction of existing samples. The combination between incremental and decremental learning is very useful because this way the memory limitation can be overcome. Besides this, they can offer a possible solution to the "concept drift" problem,[15] i.e. when a model built on old data becomes inconsistent with the new one and an update of the model is required.

In a previous paper,[12] we introduced an online version of the Incremental Non-parametric Discriminant Analysis (from now on referred to as IncNDA) technique. Unlike the classical approach of Non-parametric Discriminant Analysis (from now on referred to as BatchNDA) which requires the complete set of data in order to build the knowledge representation subspace, the IncNDA needs only a limited number of samples to build the initial subspace. Later on, as new samples become available, this subspace is efficiently updated in real-time. Our choice for NDA technique was motivated by the fact that being a nonparametric method, its application is not limited to Gaussian distributions of data. Besides this, data representation is more effective in capturing the boundary structural information for different categories (classes). Another advantage provided by this

[a]In real world scenarios, it is unlikely to know beforehand the number of total classes or the exact number of instances per class.

method is that it extracts those features which work well with the nearest-neighbor classifier.

The current paper extends our previous work, by introducing the notion of Decremental Non-parametric Discriminant Analysis (referred to from now as DecNDA). Through the combination of IncNDA and DecNDA we propose a full cognitive developing system resembling the Modal Model of Memory proposed by Atkinson and Schiffrin.[2] This online learning strategy is applied to the cognitive face recognition problem.

The paper is structured as follows: in the next section, we will briefly review the Atkinson and Schiffrin's memory model fundamentals. Section 3 is dedicated to the introduction of the online Non-parametric Discriminant Analysis (with its incremental and decremental components). In Sec. 4, we discuss the application of our approach to the problem of cognitive face recognition. Finally, Sec. 5 contains our conclusions and the guidelines for future work.

2. Modal Model of Memory

A crucial factor in cognitive systems design is the information management, i.e. how the incoming information is perceived, processed and stored. Our approach for modeling the cognitive development is inspired from the Modal Model of Memory proposed by Atkinson and Schiffrin (see Fig. 1), which resembles human memory mechanisms and which is considered one of the most influential theories among psychologists.

This model involves a sequence of three stages of memory: sensory memory, the short-term (working) memory (from now on referred to as STM) and long-term memory (from now on to referred to as LTM). The sensory memory is in charge of transforming environmental stimuli into information that can be processed by STM. It can be actually seen like a sensorial mechanism buffer (mostly visual and auditory). Due to the high amount of information perceived, only the most relevant part of it is encoded and passed to STM. The rest is discarded.

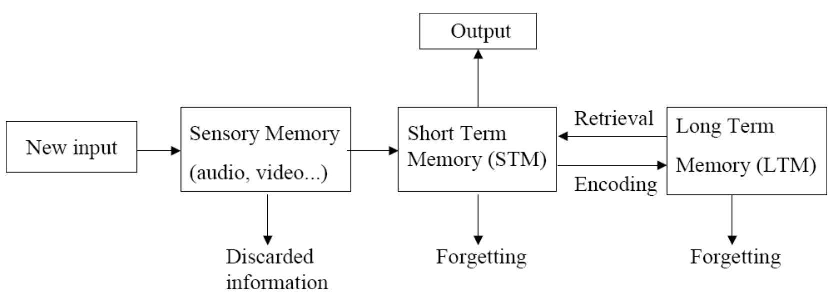

Fig. 1. Atkinson and Schiffrin's model of memory.

STM can hold information passed from either sensory memory (incoming data) or LTM (retrieved or recalled data[b]). It contains the structures and processes used for temporary storage and manipulation of currently active information. As its name suggests, it can keep the information for a very brief period of time, after which it is either passed along to the LTM or discarded. The process of consolidation (encoding and transfer of information from STM to LTM) is enhanced by the relationship, if any, of an item of STM to an item in LTM. In other words, this process is not only a function of time, but also depends on how relevant and meaningful this item is for its incorporation in LTM.

On the other hand, LTM represents the collection of information built over time. If the information passed by STM is relevant, then the knowledge content of LTM is updated. The long-term memory is responsible for guaranteeing the system viability over large period of times (as long as years). A phenomenon that can affect the LTM is the "forgetting" or "degradation" (partial or total loss of a data category). Sometimes, this process can be irreversible. In order to avoid or to postpone this situation, several retrievals/recalls of memory may be needed.

The rest of the paper is devoted to show how this cognitive model can be actually implemented. The LTM consists of feature vectors obtained from the projection of original data on its NDA-eigenspace representation. The STM has a similar structure and contains the perceived or retrieved/recalled data. Regarding the classification function, we employed the nearest neighbor rule. The "incremental learning" has been modeled through IncNDA algorithm, meanwhile "decremental learning" or "forgetfulness", through DecNDA. In the current implementation, both IncNDA and DecNDA perform a sequential update of LTM (this implies that STM contains only one data sample at any given time) and the learning strategy is supervised.

3. Online Nonparametric Discriminant Analysis

This section is devoted to the introduction of the online NDA. However, for the sake of completeness, we first briefly review the classical NDA (BatchNDA). After that, we present the proposed approaches for IncNDA and DecNDA.

As introduced in Ref. 8, 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 commonly used criteria is the one that maximizes the following expression:

$$\zeta = \mathrm{tr}(S_b S_w). \tag{1}$$

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

$$\hat{W} = \arg \max_{W^T S_w W = I} \mathrm{tr}(W^T S_b W). \tag{2}$$

[b]There is a subtle difference between the retrieval and recall processes. The former refers to the case when the action has been initiated at an external request and the latter when the action has been initiated by the system itself.

This problem has an analytical solution and is mathematically equivalent to the eigenvectors of the matrix $S_w^{-1}S_b$.

3.1. *BatchNDA*

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 Ref. 8, the S_w and S_b scatter matrices are defined as follows:

$$S_w = \sum_{i=1}^{C_N} \sum_{j \in C_i} (x_j - \bar{x}^{C_i})(x_j - \bar{x}^{C_i})^T \tag{3}$$

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

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. (6) 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.

3.2. *Sequential IncNDA*

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 will analyze these two situations below.

3.2.1. *The new training pattern belongs to an existing class*

Let us assume, for instance, that y belongs to one of the existing classes C_L (i.e. y^{C_L}, where $1 < L < N$). In this case, the equation that updates S_b is given by:

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

where $C_{L'} = C_L \bigcup \{y^{C_L}\}$, $S_b^{\text{in}}(C_L)$ represents the covariance matrix between the existing classes and the class that is about to be changed, $S_b^{\text{in}}(C_{L'})$ represents the covariance matrix between existing classes and the updated class $C_{L'}$ and by $S_b^{\text{out}}(y^{C_L})$ we denote the covariance matrix between the vector y^{C_L} and the other classes:

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

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

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

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

3.2.2. *The new training pattern belongs to a new class*

Let us assume that y belongs to a new class C_{N+1} (i.e. $y^{C_{N+1}}$).

In this case, the updated equations for the scatter matrices are given by:

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

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

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

$$S_b^{\text{in}}(C_{N+1}) = \sum_{j=1}^{C_N} \sum_{i=1}^{n_{C_j}} W(C_j, C_{N+1}, i)(x_i^j - \mu_{C_{N+1}}(x_i^j))(x_i^j - \mu_{C_{N+1}}(x_i^j))^T. \tag{14}$$

The new S_w' matrix remains unchanged i.e:

$$S_w' = S_w. \tag{15}$$

3.3. *Sequential DecNDA*

In order to describe the proposed algorithm, we assume that we have already generated an NDA eingenspace (as the result of a previous application of IncNDA or BatchNDA), i.e. we have already computed the S_w and S_b scatter matrices. The DecNDA algorithm provides the updated equations for S_w and S_b when a sample is removed from the database.

Let us denote by y, for instance, the pattern that we pretend to remove and assume that y belongs to one of the existing classes C_L, i.e. y^{C_L}, where $1 < L < N$ and $n_{C_L} > 1$. In this case, the equation that updates S_b is given by:

$$S'_b = S_b - S_b^{\text{in}}(C_L) + S_b^{\text{in}}(C_{L'}) - S_b^{\text{out}}(y^{C_L}) \tag{16}$$

where $C_{L'} = C_L \backslash \{y^{C_L}\}$ and the rest of the terms have exactly the same meaning as in the case of IncNDA and are given by Eqs. (17) and (18), respectively.

In the case when the class being updated contains only one sample, i.e. $n_{C_L} = 1$, the update equation for S_b is given by:

$$S'_b = S_b - S_b^{\text{out}}(y^{C_L}) - S_b^{\text{in}}(C_L). \tag{17}$$

Regarding the update of the S'_w matrix, in the case when $n_{C_L} > 1$, the expression is the same as given by Eq. (17), with the only difference that the $S_w(C_{L'})$ term is defined as:

$$S_w(C_{L'}) = S_w(C_L) - \frac{n_{C_L}}{n_{C_L} - 1}(y - \bar{x}^{C_L})(y - \bar{x}^{C_L})^T. \tag{18}$$

In the case when the class being updated contains only one sample, i.e. $n_{C_L} = 1$, the new S'_w matrix remains unchanged.

4. Cognitive Face Recognition: A Case Study

4.1. *Problem description*

The proposed online learning approach has been applied to a face recognition problem using a custom face database. The image acquisition phase was extended over several weeks and was performed in an automatic manner. For this purpose, we placed the camera in an open space and snapshots were taken each time a person appeared in front of it. The face was automatically segmented from the image using the Viola and Jones face detector.[17] We did not impose any restrictions regarding ambient conditions.

Overall, our database consists of 6882 images of 51 people (both male and female).[c] 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 48×48 pixels. Because of the particularity of the acquisition process, face images reflect the changes in appearance suffered by subjects over time. Furthermore, since our application run in real-time (and to give it a more *ad hoc* feeling), we did not perform any preprocessing step to face images before sending them to the classifier. That is why the faces used in our 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 were a little wider than the face region itself. Some samples of these images are presented in Fig. 2.

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

Fig. 2. Samples of face images from CVC custom database showing a certain degree of variation in illumination, pose and size.

Extensive experiments were carried out both in terms of classification accuracy and execution time (regarding IncNDA) and how "forgetfulness" degrades the retrieval/recall capability (regarding DecNDA).

4.2. *IncNDA: classification accuracy*

To test the IncNDA learning algorithm, we used 90% of the images (i.e. about 6000) 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 almost 1000 samples) to build the initial IncNDA eigenspace. In order to overcome the singularity problem, a PCA step was performed beforehand.[d] This way, data dimensionality was downsized from 2304 to 60. The remaining samples (around 5100) from the training set were added later on in a sequential manner (the samples were drawn randomly) and this way the NDA-eigenspace was updated.

In Fig. 3 (left), we depicted the evolution of the learning process after each update (a new sample added) of the initial IncNDA eigenspace. In the early stages, many new classes were presented at very short intervals. It can be appreciated that, with almost 50% of the remaining training samples introduced, all classes

[d]Because the dimensionality of a typical image (i.e. the number of pixels in it) is usually much larger than the number of available samples, the scatter matrices might be singular. In order to avoid this, a dimension reduction procedure (PCA) needs to be applied beforehand.

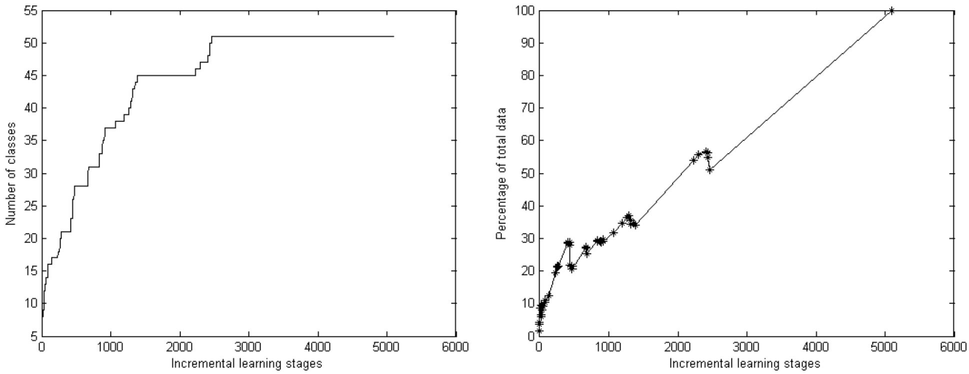

Fig. 3. Learning process: evolution of the number of classes function of learning stages (left) and the percentage of the training data function of learning stages (right).

have been represented. In Fig. 3 (right), we show the percentage of incremental training samples introduced since the start of the learning process (the stars represent the moment when a new class has been added). This graphic should be read in concordance with the left one.

As a final proof of accuracy, we compared IncNDA with the BatchNDA. In this case, a similar procedure has been followed, i.e. 90% of the data was used for training and the remaining data for test. In order to obtain meaningful and coherent results, the training and test were the same in both cases, i.e. at the end of the training process of IncNDA algorithm, the training set was identical with the one used for BatchNDA. In Fig. 4, 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 characteristics of the database. Both graphics were plotted after averaging the results obtained from a ten-fold cross-validation procedure (the training samples were randomly chosen in each run). We repeated the experiments considering a different number of neighbors (1, 3, 5, 7) when computing Eq. (4), but the best results correspond to a number of neighbors equal to 3. Figure 4 corresponds to this case.

Some instances of misclassified faces are represented in Fig. 5. From these results, we arrive of the conclusion that misclassification occurs in three situations: (1) when there are too few face instances per class, (2) when there are too few instances of a particular head pose/illumination conditions and (3) when the image presents a high level of distortion (the "blurring" effect due to person/camera movement).

4.3. *IncNDA: computation complexity*

In terms of computational complexity, the most "critical" aspect (time consuming) is the calculation of the S_b matrix, which, on its turn, is related to finding the nearest neighbor. In order to show the efficiency of our proposed technique, we measured

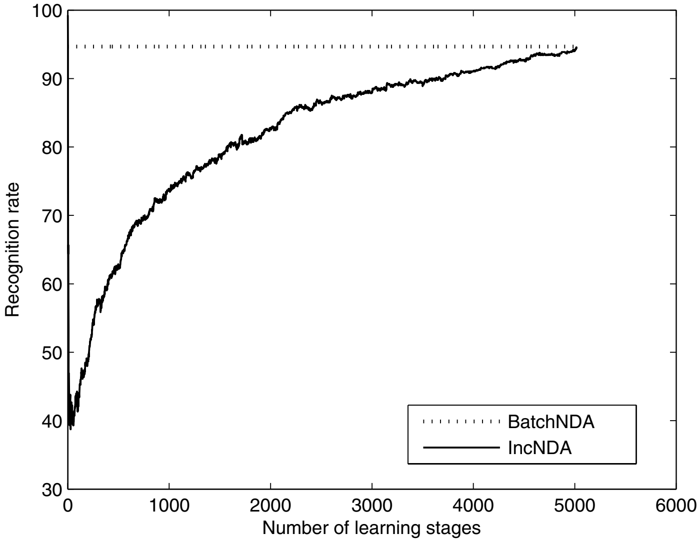

Fig. 4. IncNDA versus BatchNDA curves. IncNDA converges towards BatchNDA at the end of the learning process.

Fig. 5. Some instances of misclassified faces.

the time needed by IncNDA to update the S_b scatter matrix and compared it with time needed by BatchNDA to recalculate S_b each time from the beginning. We sequentially updated the S_b matrix with about 5000 samples (presented randomly) of dimensionality 60 (corresponding to the projection of an image on the NDA eigenspace), belonging to the 51 classes. The values in 1 represent "seconds" (CPU processing time) and were obtained considering 1 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. Alternatively, we present only the time computed every 500 samples. The experiment was done in MATLAB on a 3 GHz Pentium IV computer, with 1 GB of memory.

Table 1. Comparison between IncNDA and BatchNDA in terms of computational complexity: execution time to calculate the S_b matrix (in s).

	Number of Samples									
Method	500	1000	1500	2000	2500	3000	3500	4000	4500	5000
BatchNDA	3.64	6.21	12.46	17.37	21.48	29.78	42.54	51.32	62.21	70.52
IncNDA	0.05	0.12	0.18	0.26	0.31	0.37	0.42	0.50	0.56	0.63

Table 1 shows that the time needed to update the S_b matrix using the IncNDA approach is sensibly lower than using BatchNDA.[e] In summary, this makes our approach suitable for applications which need real-time update (for instance, for classification from video or live streams).

4.4. *DecNDA: degradation of recall/retrieve performance through "forgetfulness"*

In order to test the DecNDA performance, we randomly removed the samples from the final NDA-eigenspace built previously (the order of the samples extracted was different from the order in which they were introduced). We followed the same approach as in the case of IncNDA, i.e. applying a ten-fold cross-validation procedure. The results of the decremental learning ("unlearning") phase are depicted in the Fig. 6. It can be appreciated, that at the end of the "unlearning" process, the curve reaches the initial position (this figure should be interpreted in concordance with Fig. 6).

In the simulations described above, the samples have been removed randomly. However, in a real-world scenario, several criteria can be employed. In one of them, each sample can have an associated weight representing the time stamp of when it was acquired. After some time, the oldest ones can be completely replaced by new ones. In another scenario, the weight can represent a counter measure to how many times the sample has been selected by the nearest neighbor rule. Samples with low values can be dismissed, since they are not considered statistically "relevant".

4.5. *Discussion*

Our approach to online learning could seem somehow "expensive" in terms of memory requirements, since we store not only the projected data, but also the original one. (It is this strategy, in particular, that allow us to introduce the DecNDA algorithm). Otherwise, only the incremental phase of the online learning would have been possible. Related to this problem, things are not as bad as they might seem at first sight. Veres *et al.*[16] addressed the problem of how much biometrics data is needed from the point of view of keeping the error rates below a certain limit such that the statistical

[e]Similar results are expected to be obtained also by DecNDA, since the update equations are very similar.

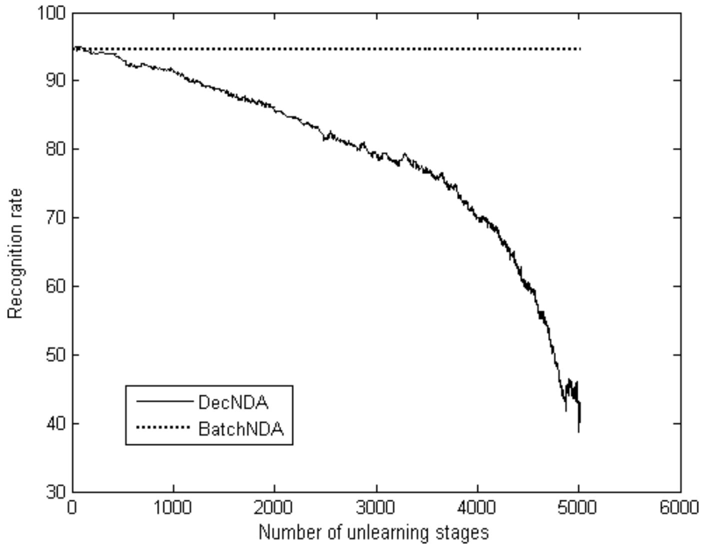

Fig. 6. DecNDA versus BatchNDA curves. DecNDA converges towards the initial status towards the end of the decremental process, when all the samples have been removed.

significance of the classification results is not affected. The estimates of sample size and the number of samples per subject are mathematically proven.

As mentioned before, one of the most time consuming steps is represented by the nearest neighbor search. An optimization step is proposed in (see Ref. 6), in order to reduce the complexity burden. The idea is to simply remove "useless" prototypes. In order to so, we must perform a Voronoi tesselation in order to obtain a partition of the class. All the samples falling in one of these cells can be thus replaced by the most representative one. The advantage of using a Voronoi tesselation is that the nearest neighbor searching time is reduced while keeping the class boundary (and hence the error) unchanged.

In order to postpone the effect of a total loss of a category which characterizes LTM, a recall (action initiated by the system itself) strategy could be implemented which can take place in increasing intervals in accordance with the principle of "spaced repetition" (increasing intervals of time are used between subsequent reviews).

5. Conclusions and Future Work

In the current paper, we have presented an online learning strategy aimed at improving the design of cognitive developmental systems. This online learning technique is built using the Atkinson and Schiffrin's model of memory as reference. The update of the LTM can be achieved in two directions: either adding samples/categories to the existing ones (IncNDA) or removing the exisiting sample/categories (DecNDA). This online learning strategy has been applied to a cognitive face recognition system.

Simulation performed on a custom face database show that IncNDA learning curve converges towards BatchNDA at the end of the learning process. Furthermore, the time needed by both IncNDA and DecNDA to sequentially update the NDA eigenspace representation is two orders of magnitude lower than BatchNDA (if we compute this representation each time from scratch). This result is very relevant because it can be implemented on real-time systems.

In the future, we plan to perform the update of the LTM (for both IncNDA and DecLDA) in terms of data chunks. The motivation behind this is that in humans, the retrieval/recall capability of STM improves when the data is presented in chunks, rather sequential. Psychological research revealed that human mind is more robust in recalling long strings of data (numbers, letters or words) when these are grouped in chunks, rather than being isolated (see Ref. 10). Furthermore, it was shown that chunk size does depend on the category to be memorized (for instance, chunk size is around 7 for numbers, around 6 for letters and around 5 for words).

Although the results obtained so far are very promising, they are obtained on a closed data set. The real challenge will be to have this cognitive architecture implemented in a real system working for a very long period of time (days, maybe weeks), in order to assess the real performances of the proposed approach. This would open up a new line of research that consists in changing the learning type, from supervised to semi-supervised (i.e. taking into account both labeled and un-labeled data) and would represent the first step towards a complete unsupervised learning strategy.

Acknowledgments

This work is supported by MEC Grant TIN2006-15308-C02 and CONSOLIDER-INGENIO 2010 (CSD2007-00018), Ministerio de Educacin y Ciencia, Spain. Bogdan Raducanu is supported by the Ramon y Cajal research program, Ministerio de Educación y Ciencia, Spain.

References

1. M. Artač, M. Jogan and A. Leonardis, Incremental PCA for on-line visual learning and recognition, *Proc. 16th Int. Conf. Pattern Recognition*, Québec, Canada **3** (2002) 781–784
2. R. C. Atkinson and R. M. Schiffrin, Human memory: a proposed system and its control processes, *The Psychology of Learning and Motivation: Advances in Research and Theory*, eds. K. W. Spence and J. T. Spence, Vol. 2 (Academic Press, London, 1968).
3. R. A. Brooks, Intelligence without reason, *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI)* Sydney, Australia (1991), pp. 569–595.
4. G. Cauwenberghs and T. Poggio, Incremental and decremental support vector machine learning, *Adv. Neural Information Processing Systems (NIPS*2000)*, Vol. 13 (MIT Press, Cambridge, MA, 2001).
5. S. Chandrasekaran, B. S. Manjumath, Y. F. Wang, J. Winkler and H. Zhang, An eigenspace update algorithm for image analysis, *Graph. Mod. Imag. Process.* **59** (1997) 321–332.

6. R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, 2nd edn. (John Wiley and Sons, NY, USA, 2001).
7. T.-K. Kim, S.-F. Wong, B. Stenger, J. Kittler and R. Cipolla, Incremental linear discriminant analysis using sufficient spanning set approximations, *Proc. IEEE Conf. Comput. Vision Patt. Recog. (CVPR)*, Minneapolis, USA, 2007.
8. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd edn. (Academic Press, Boston, USA, 1990).
9. P. Hall, D. Marshall and R. Martin, Incremental eigenanalysis for classification, *Proc. British Machine Vision Conf.*, Southampton, UK, **1** (1998) 286–295.
10. G. A. Miller, The magical number seven, plus or minus two: some limits on our capacity for processing information, *Psychol. Rev.* **63** (1956) 81–97.
11. S. Pang, S. Ozawa and N. Kasabov, Chunk incremental LDA computing on data streams, *Advances in Neural Networks*, eds. Wang *et al.*, Lecture Notes in Computer Science, Vol. 3497 (Springer-Verlag, Berlin, 2005), pp. 51–56.
12. B. Raducanu and J. Vitrià, Incremental subspace learning for cognitive visual processes, *Advances in Brain, Vision and Artficial Intelligence*, eds. F. Mele *et al.*, Lecture Notes in Computer Science, Vol. 4729 (Springer-Verlag, Berlin, 2007), pp. 214–223.
13. D. Skočaj, M. Uray, A. Leonardis and H. Bischof, Why to combine reconstructive and discriminative information for incremental subspace learning, *Proc. Computer Vision Winter Workshop*, eds. Chum *et al.*, Telč, Czech Republic (2006).
14. D. M. J. Tax and P. Laskov, Online SVM learning: from classification to data description and back, *Proc. IEEE Workshop on Neural Networks for Signal Processing*, Toulouse, France (September 2003).
15. A. Tsymbal, The problem of concept drift: definitions and related work, Technical Report TCD-CS-2004-15, Computer Science Department, Trinity College Dublin (2004).
16. G. V. Veres, M. S. Nixon and J. N. Carter, Is enough enough? What is sufficiency in biometric data, *Image Analysis and Recognition*, eds. A. Campilho and M. Karmel, Lecture Notes in Computer Science, Vol. 4142 (Springer-Verlag, Berlin, 2006), pp. 262–273.
17. P. Viola and M. J. Jones, Robust real-time face detection, *Int. J. Comput. Vis.* **57** (2004) 137–154.
18. J. Weng, Y. Zhang and W.-S. Hwang, Candid covariance-free incremental principal component analysis, *IEEE Trans. Patt. Anal. Mach. Intell.* **25** (2003) 1034–1040.

Bogdan Raducanu received the B.Sc. in computer science from the Politechnical University of Bucharest (PUB), Romania, in 1995 and a Ph.D. cum laude from the University of The Basque Country (UPV/EHU), Spain, in 2001. Currently, he is a post-doc researcher at the Computer Vision Center, Barcelona, Spain.

His research interests include computer vision, pattern recognition, artificial intelligence and social robotics.

Jordi Vitrià received the Ph.D. degree from the Autonomous University of Barcelona (UAB), Barcelona, Spain, for his work on mathematical morphology in 1990. He joined the Computer Science Department, UAB, where he became an Associate Professor in 1991.

His research interests include machine learning, pattern recognition, and visual object recognition. He is the author of more than 40 scientific publications and one book.