Learning to Recognize Faces Incrementally

O. Deniz, J. Lorenzo, M. Castrillon, J. Mendez, and A. Falcon

Dpto. Informatica y Sistemas. Universidad de Las Palmas de Gran Canaria
Campus de Tafira, Edificio Informatica. 35017 Las Palmas Spain

Abstract. Most face recognition systems are based on some form of batch learning. Online face recognition is not only more practical, it is also much more biologically plausible. Typical batch learners aim at minimizing both training error and (a measure of) hypothesis complexity. We show that the same minimization can be done incrementally as long as some form of "scaffolding" is applied throughout the learning process. Scaffolding means: make the system learn from samples that are neither too easy nor too difficult at each step. We note that such learning behavior is also biologically plausible. Experiments using large sequences of facial images support the theoretical claims. The proposed method compares well with other, numerical calculus-based online learners.

1 Introduction

Face recognition is becoming one of the most researched problems in Computer Vision. The available literature is increasing at a significant rate, and even the number of conferences and special issues entirely devoted to face recognition is growing. Access to inexpensive cameras and computational resources has allowed researchers to explore the problem from many different perspectives, see the survey [5].

Humans are very competent when it comes to recognize faces. A number of face recognition systems have been based, at least partially, on psychophysical or neurophysiological findings related to face recognition in humans. The use of biologically-inspired features for discrimination is a prominent example, with Gabor features topping the list.

Other work has tackled online face recognition. The interest, however, seems to have been mainly practical, rather than based on biological plausibility considerations. In particular, attempts have been made at alleviating the high computational cost of the most common feature selection model used in face recognition, namely Principal Component Analysis (PCA). Incremental PCA [6] aims at updating the PCA basis incrementally and is computationally efficient for large scale problems. Incremental algorithms have been also proposed for classification. Incremental SVM [4], for example, is a computationally efficient version of the successful SVM (Support Vector Machines) classifier, which typically requires solving a quadratic programming (QP) problem in a number of coefficients equal to the number of training samples.

The ideal situation would be to find a technique that is both sound from a theoretical viewpoint and plausible from a biological viewpoint. This work proposes a novel incremental learning algorithm that is heavily inspired by biological plausibility aspects. The paper is organized as follows. Section 2 describes the notion of *complexity minimization*, and proposes an incremental learning framework for face recognition. Experiments are shown in Section 3. Finally the main conclusions and ideas for future work are outlined.

2 Incremental Learning

As mentioned above, batch supervised learning is behind the vast majority of face recognition systems. A principled way to avoid overfitting in supervised learning is the use of complexity penalization. A well-known complexity penalization technique is Structural Risk Minimization (SRM) [2]. SRM is a procedure that considers hypotheses ranging from simple to complex. For each hypothesis the error in the training set is measured. Basically, the best hypothesis is that which minimizes the sum of its error in the training set and (a measure of) its complexity: $\arg\min_H \quad e_{training}(H) + Complexity(H)$.

Hypothesis complexity can be assimilated (although not strictly) to its number of parameters. The more parameters the more the discriminating power, but also the larger its complexity. Hypothesis complexity is commonly measured as a norm in the hypothesis parameter space. This form of complexity penalization has been used in face recognition for some time. In particular, Support Vector Machines [11], which is based on SRM, has been shown to give better results than other techniques for many tasks.

In a complexity penalization framework a search is made for the minimum hypothesis variation with respect to the ”zero” hypothesis. The zero hypothesis corresponds to the origin of the functional space, the hypothesis of zero complexity or capacity (i.e. that with no discriminating power). This complexity penalization approach can be made incremental if a search is made for the minimum variation with respect to the current hypothesis, while achieving consistency with a set of new training samples, see Figure 1.

In complexity penalization techniques a search is made for the simplest hypothesis that is consistent with the training samples $(x_y, y_i),\ y_i = \{\pm 1\},\ \ i = 1,..,n$. Hypotheses are generally represented in a functional Reproducing Kernel Hilbert Space (RKHS), a convenient tool from functional analysis [1]. In RKHSs, functions are represented by coefficients or coordinates. The function itself is reproduced as a sum of the coefficients multiplied by symmetric kernel functions centered at the training samples. For classification, the decision function is given by:

$$sig\left(f(x) = \sum_{i=1}^{n} c_i y_i K(x, x_i)\right) \tag{1}$$

The functional to minimize is, [8]:

$$J_n = e_n + \lambda ||f||^2 \tag{2}$$

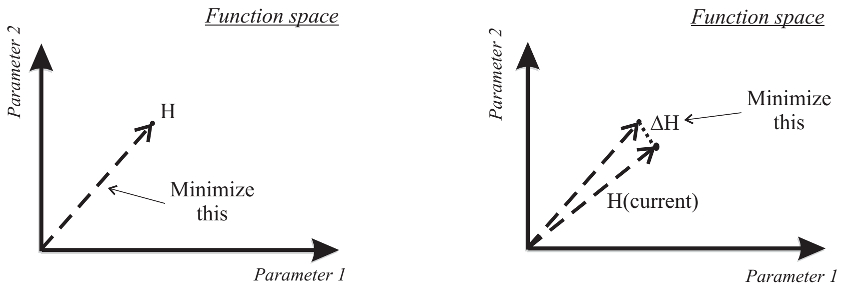

Fig. 1. One-step (batch) complexity minimization (left), incremental complexity minimization (right)

where $e_n = \frac{1}{n}\sum_{i=1}^{n} Err(y_i, f(x_i))$ is the error on the n training samples and $||f||^2$ is the norm of the hypothesis considered. This last term, sometimes called *regularizer*, penalizes the complexity of the hypothesis. For a given kernel and set of training samples, learning algorithms search for a set of non-negative c coefficients that minimize Eq. 2.

Let us divide the learning process in two stages. Let us suppose that we form a hypothesis $f' = f + \cdot f$, where f is the hypothesis obtained from the first $n-m$ samples and $\cdot f$ the hypothesis obtained for the m samples. Let e_{n-m}^{f} represent the training error in the first $n-m$ samples with hypothesis f, and e_n^f the training error with hypothesis f using all the n samples. Then:

$$\begin{aligned} J_n = e_n^{f'} + \lambda||f'||^2 = e_n^{f'} + \lambda||f + \cdot f||^2 = \\ = e_n^{f'} + \lambda||f||^2 + \lambda||\cdot f||^2 + 2\lambda < f, \cdot f >= \\ = e_n^{f'} + e_{n-m}^{f} - e_{n-m}^{f} + e_m^{\cdot f} - e_m^{\cdot f} + \lambda||f||^2 + \lambda||\cdot f||^2 + 2\lambda < f, \cdot f >= \\ = J_{n-m} + J_m + e_n^{f'} - (e_{n-m}^{f} + e_m^{\cdot f}) + 2\lambda < f, \cdot f > \end{aligned} \tag{3}$$

Note that both f and $\cdot f$ are vectors in a function space. $< \cdot, \cdot >$ is the dot product of that space. Summarizing Equation 3, we have:

$$J_n = J_{n-m} + J_m + \alpha + 2\lambda\beta \tag{4}$$

where:

$$\alpha = e_n^{f'} - (e_{n-m}^{f} + e_m^{\cdot f}) \tag{5}$$

$$\beta = < f, \cdot f > \tag{6}$$

$$\tag{7}$$

Our objective is to minimize J_n by minimizing the right hand side of Eq. 4, in an incremental fashion. That is, we want to minimize J_n in steps, first minimizing J_{n-m} and then J_m. The terms α and β would then have to be minimized too. Having minimized J_{n-m} in a previous step, suppose that we minimize J_m. If

m is sufficiently small, $e^{\cdot f}_{m}$ can be made arbitrarily close to zero. On the other hand, if m is kept fixed, the difference $e^{f'}_{n} - e^{f}_{n-m}$ decreases as n grows. Thus, if m is kept fixed α decreases as n grows. Note that $m << n$ does not hold at the beginning of learning, and so it may be necessary to start with a batch learning run in which more than m samples are used in the first step (this, however, was not necessary in the experiments reported below).

Finally, let us turn our attention to the term β. Recall that we are interested in minimizing J_n through the minimization of the right hand side of Eq. 4. Therefore, we have to enforce:

$$|\beta| = | < f, \cdot f > | \;\; \approx \; 0 \; , \tag{8}$$

Note also that:

$$f(\cdot) = \sum_{i=1}^{n-m} c_i y_i K(\cdot, x_i) \tag{9}$$

and

$$\cdot f(\cdot) = \sum_{j=1}^{m} d_j y_{j+n-m} K(\cdot, x_{j+n-m}) \; , \tag{10}$$

c_i and d_j being the coefficients obtained in the minimization of J_{n-m} and J_m, respectively. Samples $(x_1, y_1), ...\ (x_{n-m}\ , y_{n-m})$ are used for minimizing J_{n-m}, while $(x_{n-m+1},\ y_{n-m+1})$, ... (x_n, y_n) are used for minimizing J_m. Then (see [7]):

$$< f, \cdot f > = \sum_{i=1}^{n-m} \sum_{j=1}^{m} c_i y_i d_j y_{j+n-m} K(x_i, x_{j+n-m}) \tag{11}$$

Now let us suppose for simplicity that $m = 1$, then:

$$\cdot f(\cdot) = d_j y_{j+n-1} K(\cdot, x_{j+n-1}) \; , \tag{12}$$

and

$$\begin{aligned} < f, \cdot f > &= \sum_{i=1}^{n-1} c_i y_i d_j y_{j+n-1} K(x_i, x_{j+n-1}) = \\ &= d_j y_{j+n-1} \sum_{i=1}^{n-1} c_i y_i K(x_i, x_{j+n-1}) \end{aligned} \tag{13}$$

When sample x_{j+n-1} arrives, it would be classified by the (at that moment) current hypothesis using the sign of:

$$\begin{aligned} f(x_{j+n-1}) &= \sum_{i=1}^{n-1} c_i y_i K(x_{j+n-1}, x_i) = \\ &= \sum_{i=1}^{n-1} c_i y_i K(x_i, x_{j+n-1}) \end{aligned} \tag{14}$$

Then, from Eqs. 13 and 14 we see that:

$$< f, \cdot f >= d_j y_{j+n-1} f(x_{j+n-1}) \tag{15}$$

Our original requirement of Eq. 8 is therefore equivalent to:

$$|d_j y_{j+n-1} f(x_{j+n-1})| \approx 0 \tag{16}$$

Note that when the incoming sample is correctly classified by the current hypothesis the product $y_{j+n-1} f(x_{j+n-1})$ of Eq. 16 is larger than zero (in that case the signs are equal). On the contrary, when the incoming sample is incorrectly classified by the current hypothesis the product is negative [1]. This means that Eq. 8 only holds for samples that are neither too "easy" (i.e. a sample correctly classified, with $y_{j+n-1} f(x_{j+n-1}) >> 0$), nor too "difficult" (i.e. a sample incorrectly classified, with $y_{j+n-1} f(x_{j+n-1}) << 0$).

Above, $m = 1$ was used out of simplicity, although in practice m should be at least 2. It can be shown that for $m = 2$ the dot product is made up of two summands, each similar to the right hand side of 15. The interpretation is the same: the dot product will be low when the (two) new samples are neither too easy nor too difficult for the current hypothesis. Note that this theoretical requirement is in line with what occurs in human learning, where learning only progresses if there is scaffolding, see [12]. Consequently, this framework would work if we make the learner process samples that are neither too easy nor too difficult at each step. Such approach would be closely related to what is known as *active learning*.

Another form of achieving scaffolding will be used here. First, note that the left hand side of (16) could be kept low if the coefficient d_j of each new sample is adjusted: the larger the dot product the smaller the adjusted d_j to use. This way, the larger the dot product (which is a measure of similarity of the new sample to the previous ones) the less weight of the of the new sample in the hypothesis.

There is another possibility. Similarities depend on the kernel function $K(x, y)$ (see Eq. 1). In kernel learning this function is commonly considered a similarity measure ([9,10]), which has to be defined a priori. The larger its value, the larger the similarity between samples x and y. A typical kernel function is the RBF kernel:

$$K(x, y) = \exp\left(-\frac{||x - y||^2}{p^2}\right) \tag{17}$$

The larger p the larger the similarity values given by the RBF kernel. Now in this context, what are too-easy and too-difficult samples? The former are samples that are very similar to other (previously seen) samples of its same class, while the latter are samples that are very similar to other (previously seen) samples of the opposite class. Therefore, the value of p is important here: a large p will give large similarity values and thus too-easy and too-difficult samples.

[1] The d_j are always non-negative, it is a requirement imposed in the minimization process, see [2].

Scaffolding can be achieved by making the p values dependent on the absolute dot products. That is, the larger the absolute dot product of a new sample the smaller the p to use in the kernel associated to that sample. Such approach requires the dot product of Eq. 15 to be calculated for each new sample. The coefficients d_j and y_{j+n-1} are obtained after the minimization of $\cdot f$, which can be done in constant time (assuming m is kept fixed). The term $f(x_{j+n-1})$ requires evaluating f for each new sample, which has a cost $O(\sum_{i=1}^{n} i) = O(\frac{n(n+1)}{2}) = O(\frac{n^2+n}{2})$, n being the number of samples processed.

Once the dot product is calculated, the new sample will contribute to the hypothesis of Eq. 1 with a kernel p value dependent on the absolute dot product. The larger the absolute dot product the smaller the p used. The exact function used to achieve this will be shown below.

3 Experiments

The incremental learner introduced above was tested in a face recognition problem. The experiments required a large number of images per individual. The ENCARA2 system was used to collect a number of face image sequences. ENCARA2 is a face detection and normalization system that can detect and track people in real-time, see [3]. ENCARA2 tries to confirm that images actually contain a face and, if so, normalize them so as to be recognized. The final result is a set of frontal face images, normalized and ready to be recognized, see Figure 2.

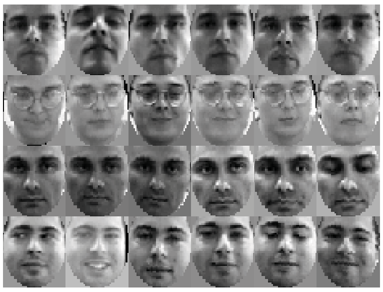

Fig. 2. Four (partial) face image sequences obtained with the ENCARA2 system

Twenty-five sequences were used, one for each individual. Each sequence had 300 normalized images of 39x43 pixels. Thus, a total of 7500 images were used in the experiments. PCA was initially applied (over the whole set of 40 training images per class, retaining 10 coefficients. Note that, in practice, PCA would be applied to an initial large set of labeled samples. The obtained basis images would then be used from that moment on to transform any incoming image to the new space, exactly as it would have to be done in a batch mode system. It is important to note that this paper is introducing an incremental classifier, not an incremental input space transform.

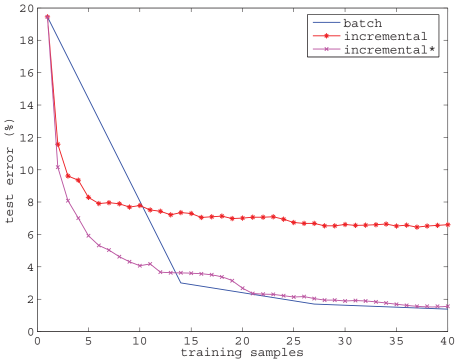

Fig. 3. Error obtained using the batch and incremental learning modes using $p = 800$ (i.e. the p which gave the minimum batch error). The horizontal axis is the number of training samples per class. In order to speed up the experiments, only four n values were calculated for the batch learner. Median of 10 runs.

An SVM classifier with Radial Basis Function (RBF) kernel (Eq. 17) was used. SVM is a binary classifier. $(N(N-1))/2$ binary classifiers were used for N-class classification. In a first experiment the test error rate of the batch learner was obtained for the values of $p = \{100, 200, 400, 800, 1600, 3200, 6400, 12800\}$, using 40 training samples. Error rates for $p = 100$ and $p = 12800$ were 59.4% and 6.4%, respectively. The best batch error (1.13%) was obtained for $p = 800$.

In the figures below, 'Incremental*' is the performance of the incremental learner using the strategy mentioned above. The strategy consists of making the p value associated to the new sample dependent on the corresponding absolute dot product: the larger the absolute dot product, the smaller the assigned p. When the current hypothesis is $f(x)$ and a new sample x_i is received, the p'_i value to use will be given by $p'_i = p_{initial} \cdot K^{-|dotproduct|}$, where $p_{initial}$ is the base p value (i.e. the one used in batch mode) and $K > 1$ is a constant. Note how the dot product is the same as that of Eq. 15. With this equation, the larger the absolute dot product, the smaller the assigned p. The value of K for p'_i equation was obtained using 130 samples -not used for training- per individual as a validation set. Figure 3 shows the results for $p = 800$. Note that in this case $p_{initial} = 800$. 'Incremental' is the performance of the incremental learner using always that $p_{initial}$ value.

More importantly, the 'Incremental*' approach gives lower error than 'incremental'. This difference is statistically significant. A t-test was made with the null hypothesis "means of Incremental and Incremental* errors are equal" vs. "mean of Incremental* is smaller". For $n = 40$ the t-test p-value was $1.2 * 10^{-11}$, a negligible support for the null hypothesis. This confirms the idea that the scaffolding strategy of penalizing too-easy and too-difficult samples has a positive effect. Therefore, the experimental results allow to infer that this incremental

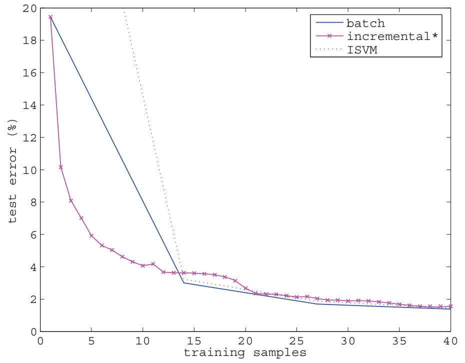

Fig. 4. Learning curves of the three compared learners

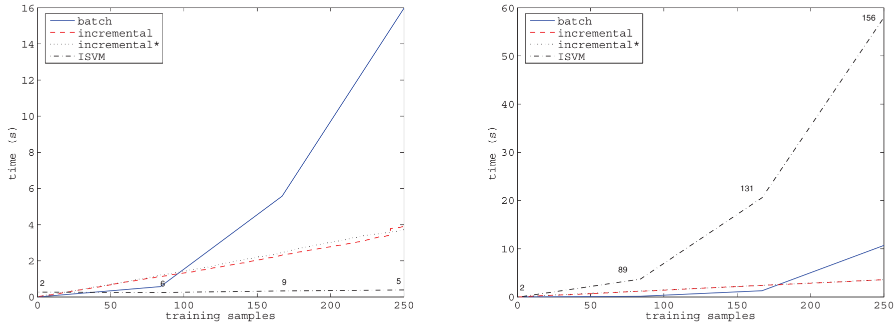

Fig. 5. Computational cost of the compared learners. Measures taken for 2-individual recognition. Left: $p = 800$, right: $p = 100$.

learning framework can work as long as the learner somehow processes samples that are not too-easy or too-difficult.

How well does the proposed learner compare with other incremental learners? In order to answer this we studied the learner proposed by Cauwenberghs and Poggio [4]. This state-of-the-art incremental SVM learner (ISVM in what follows) is based on retaining the *Karush-Kuhn-Tucker* conditions on all previously seen data, while adding a new sample to the hypothesis. According to their authors, ISVM is an *exact* online method. That is, it theoretically gives the same results as the equivalent batch learner. ISVM is an example of a number of incremental learners based on practicality considerations. These learners are generally based on properties of advanced numerical calculus. Figure 4 shows the learning curves of the three compared learners. The ISVM error at $n = 40$ is slightly closer to the batch error (batch=1.29%, incremental*=1.51%, ISVM=1.45%).

Figure 5 compares the computational cost of the three learners. The left figure shows that, for the best batch p value of 800, ISVM is much faster than the other learners. However, the theoretical computational cost of ISVM is $O(s^2)$, s being the number of support vectors. For $p = 800$ the number of support vectors is low (they appear as numbers in the figure). For $p = 100$ (the figure on the right) the number of support vectors is larger, and ISVM rates even worse than the batch learner. Incremental* has a computational cost of $O(\frac{n^2+n}{2})$. $\frac{n^2+n}{2} < n^2$, which suggests that Incremental* may outperform ISVM for particularly complex problems where a large number of support vectors are needed. Incremental* has a storage cost of $O(n)$ (i.e. just the hypothesis itself).

In batch learning, the learner has all of the training samples available from the beginning and thus it can select those that define a good discrimination boundary (i.e. the so-called support vectors). In incremental learning, this is not possible, for only some of the training samples have been given to the learner at a given moment. The natural approach in this case is to gradually span the input space with similarity functions centered in the received training samples. The last processed samples have smaller similarity radii than the first ones. This is what the proposed learner does. The similarity functions are the kernels. The p parameter acts as a radius. The assigned p values decrease with n because the dot products (which represent similarity to the previous samples) increase.

The proposed learner has at least three aspects of a strong biological plausibility. First, it is an online learner. Second, it requires scaffolding to learn. Third, it always classifies each incoming sample with the current hypothesis, being the result of that classification what can make the learner update the hypothesis.

4 Conclusions

A number of existing batch learners used in face recognition, including those based on Support Vector Machines, aim at minimizing both training error and (a measure of) hypothesis complexity. Inspired by biological plausibility considerations, especially those related to the learning process itself, in this work it has been shown that the same complexity minimization can be done incrementally as long as the learning process is aided by some form of "scaffolding", where samples processed by the learner are neither too easy nor too difficult. Within this framework, the feasibility of online learning, both in terms of error difference with respect to batch learning and computational cost at each step, crucially depends on scaffolding. Although there are other ways to achieve scaffolding, a gradually decreasing kernel parameter has been used here. The proposed method has been analyzed in experiments and compared with one state-of-the-art incremental learner. The results show that it compares favorably in terms of biological plausibility and computational and storage cost.

The proposed method seems to be a departure from mainstream approaches in face recognition. We note that this may be only a particular instantiation of a general class of learners that have features generally not found in previous face recognition research, notably a marked biological plausibility of the learning process. Further algorithms may be possible that, like this one, rely on such considerations.

Acknowledgements

This work was funded by research projects *UNI2005/18* and *TIN2004-07087*.

References

1. Aronszajn, N.: Theory of reproducing kernels. Trans. Am. Math. Soc. 68(3), 337–404 (1950)
2. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2(2), 121–167 (1998)
3. Castrillon, M., Deniz, O., Guerra, C., Hernandez, M.: ENCARA2: Real-time detection of multiple faces at different resolutions in video streams. Journal of Visual Communication and Image Representation (2007) (in press)
4. Cauwenberghs, G., Poggio, T.: Incremental and decremental support vector machine learning. In: NIPS, pp. 409–415 (2000)
5. Chellappa, R., Zhao, W. (eds.): Face Processing: Advanced Modeling and Methods. Elsevier, Amsterdam (2005)
6. Hall, P., Marshall, D., Martin, R.: Incremental Eigenanalysis for classification. In: Proceedings of the British Machine Vision Conference, vol. 1, pp. 286–295 (1998)
7. Hofmann, T., Scholkopf, B., Smola, A.J.: A tutorial review of RKHS methods in machine learning (2006), Available at `http://sml.nicta.com.au/~smola/papers/unpubHofSchSmo05.pdf`
8. Poggio, T., Smale, S.: The mathematics of learning: Dealing with data. Amer. Math. Soc. Notice 50(5), 537–544 (2003)
9. Scholkopf, B., Smola, A.: Learning with kernels. MIT Press, Cambridge, MA (2002)
10. Vanschoenwinkel, B., Manderick, B.: Appropriate kernel functions for support vector machine learning with sequences of symbolic data. In: Deterministic and statistical methods in machine learning (First international workshop), Sheffield, UK, September 2004, pp. 256–280 (2004)
11. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
12. Vygotsky, L.: Mind and society: The development of higher mental processes. Harvard University Press (1978)