MULTITASK LEARNING:
An application to Incremental Face Recognition

David Masip
Universitat Oberta de Catalunya, Rambla del Poblenou 156, 08018 Barcelona, Spain.
dmasipr@uoc.edu

Àgata Lapedriza
Computer Vision Center, Computer Science Depart., Universitat Autònoma de Barcelona, Edifici O Bellaterra, Barcelona 08193, Spain
agata@cvc.uab.es

Jordi Vitrià
Department of Appl. Math. and Analysis, University of Barcelona, Edifici Històric, Gran Via de les Corts 585, Barcelona 08007, Spain.
jordi@cvc.uab.es

Keywords: Face Classification, Incremental Learning, Boosting

Abstract: Usually face classification applications suffer from two important problems: the number of training samples from each class is reduced, and the final system usually must be extended to incorporate new people to recognize. In this paper we introduce a face recognition method that extends a previous boosting-based classifier adding new classes and avoiding the need of retraining the system each time a new person joins the system. The classifier is trained using the multitask learning principle and multiple verification tasks are trained together sharing the same feature space. The new classes are added taking advantage of the previous learned structure, being the addition of new classes not computationally demanding. Our experiments with two different data sets show that the performance does not decrease drastically even when the number of classes of the base problem is multiplied by a factor of 8.

1 Introduction

Face recognition problem can be stated as a machine learning process where we receive as input a high-dimensional data vector $\mathbf{X} \in \mathbb{R}^D$ (considering the $n_1 \times n_2 = D$ face image), and we must provide the identity or class membership $C \in \{C_1, \ldots, C_K\}$ of the subject. In real-world applications the number of classes is large, being most of the classic machine learning methods not suitable for the face recognition task. In addition, the number of available samples from each class is usually limited, making the estimation of the classifier parameters more difficult. Furthermore, in real-world applications the number of subjects to identify is variable in time: for example, an automatic face recognition system for presence checking should take into account that new subjects can be added to the system avoiding the need of retraining the whole learning scheme.

Most of the face recognition algorithms found on the literature focus on the problem of classification in high dimensional subspaces. Usually a feature extraction step is performed in order to reduce the problem complexity, and then a classifier is applied on the reduced space. Many unsupervised feature extraction methods have been applied to face recognition, being the seminal paper in this field the one proposing the "eigenfaces" approach, by Turk and Pentland (Turk and Pentland, 1991), that uses Principal Component Analysis (PCA) to find the optimal subspace under the reconstruction error criterion. On the other hand, supervised feature extraction techniques also take into account the labels of the data in the feature extraction task. Fisher Linear Discriminant Analysis (LDA) (Fisher, 1936) is the most known technique, and different extensions of the algorithm have been developed to relax some of the original assumptions. Some examples are the Nonparametric Discriminant Analysis (Fukunaga and Mantock, 1983) or more recently the boosted feature extraction (Masip et al., 2005). The main drawback of this techniques is that usually we have few training samples from each class. Moreover the system is difficult to scale when new classes join the system.

In this paper we introduce a face recognition scheme to deal with the above mentioned difficulties: the robustness against the small-size training set problem and the scalability to add new classes avoiding a new costly additional training step. For this purpose we consider the multitask learning (MTL) paradigm for the face recognition problem. The term MTL was firstly introduced by Caruana (Caruana, 1997), showing that simultaneously learning related tasks in an environment can achieve important improvements at two different levels: the number N of training samples needed to learn each classification tasks decreases as more related tasks are learned (parallel knowledge transfer), and it has been proved that under some theoretic conditions, a classifier trained on sufficiently related tasks is likely to find good solutions to solve novel related tasks (sequential knowledge transfer). Baxter (Baxter, 2000) proved that the number of training samples required to train each tasks decreases linearly as the number of tasks increases $O(\frac{1}{N}log(O(K)))$.

Torralba et al. (Torralba et al., 2004) extended the MTL principle to the ensemble classifiers field, introducing a new algorithm, called Joint Boosting, where they combine the use of an ensemble of simple classifiers (using the GentleBoost algorithm (Friedman et al., 2000)) with a feature sharing strategy that implements an structural approach for multitask learning.

In this work we use the Joint Boosting algorithm to build a robust classifier for face recognition using a fixed number of classes K. Then we extend the algorithm in order to incorporate new unseen classes. The main goal of our proposal is to incorporate new persons to recognize, avoiding the computationally expensive cost of retraining the whole system. Our experimental results show that the performance of the extended class problem does not degrade drastically.

In the next section we describe our proposed method, which is based on the Gentleboost algorithm (Friedman et al., 2000). We focus our work on extending this approach to the addition of new tasks without training the whole system, with minimum computational cost. Section 3 describes the experiments performed using two standard face databases, focusing on evaluating the scalability against the addition of new subjects to the system. Finally Section 4 concludes this work.

2 GentleBoost Applied to Face Recognition

The original Adaboost algorithm (Freund and Schapire, 1996) builds iteratively a binary classifier such that the final classification rule is a linear combination of weak classifiers. At each boosting step a new classifier is generated, and the training samples are reweighted according to the classification results. The weights are used to generate the next step classifier. In the literature, there are multiple implementations of the boosting scheme. Our approach is based on the “GentleBoost” algorithm proposed by Friedman et al. (Friedman et al., 2000) which has been shown to be more robust and appropriate for face classification tasks (Yokono and Poggio, 2006).

To extend the binary Adaboost classifier to the multiclass case, two different approaches have been found in the recent literature: extend the classification rule to the multiclass case (Ji Zhu and Hastie, 2006) or to combine different binary classifiers using error correction output codes (Schapire, 1997). Torralba et al. (Torralba et al., 2004) introduced the knowledge transfer concept on the gentleAdaBoost. The main idea is to see the multiclass classification problems as multiple binary classification tasks. They experimentally show that the obtained multiclass classifier needs less training examples and also less different features.

In this paper we use the shared feature boosting approach to build a global scheme where new classes can be added to the system.

2.1 Training the original Joint Boosting Algorithm

The algorithm takes as input the N training samples $\mathbf{X} = \{\mathbf{X_i} = (x_i^1, \ldots, x_i^d)\}$ and the corresponding labels $C \in \{C_1, \ldots, C_K\}$. A predefined number M of boosting rounds are performed [1]. At each boosting step, the multiclass classification problem is converted to a binary problem by grouping the classes in a positive and a negative cluster. A decision stumps classifier is trained on the new binary problem. The parameters of the weak learner are computed as:

$$\rho = \frac{\sum_{C\in Positive(n)} \sum_i \mathbf{W}_i^C b_i^C \delta(\mathbf{x}_i^j \leq \theta)}{\sum_{C\in Positive(n)} \sum_i \mathbf{W}_i^C \delta(\mathbf{x}_i^j \leq \theta)}, \qquad (3)$$

$$\alpha + \rho = \frac{\sum_{C\in Positive(n)} \sum_i \mathbf{W}_i^C b_i^C \delta(\mathbf{x}_i^j > \theta)}{\sum_{C\in Positive(n)} \sum_i \mathbf{W}_i^C \delta(\mathbf{x}_i^j > \theta)}, \qquad (4)$$

$$k^C = \frac{\sum_i \mathbf{W}_i^C b_i^C}{\sum_i \mathbf{W}_i^C}, \text{ if c} \notin \text{ Positive(n)} \qquad (5)$$

where k^c acts as a constant to prevent the effects of unbalanced training sets on the class selection and $\{W_i^c\}$ is the weights set.

In a first attempt, all the possible groupings could be made, $O(2^K)$. Nevertheless, when the number of classes is large this approach is not possible. Torralba et al.(Torralba et al., 2004) followed a best first search

[1] We assume that the reader is familiar with the Joint Boosting algorithm. For a detailed description the reader can see (Torralba et al., 2004)

```
1. Given the matrix $\mathbf{X_{1,\ldots,N+Q}}$ containing data samples $\mathbf{x}_i$, and the vector $\mathbf{c}$ with the corresponding labels $c_i \in \{C_1,\ldots,C_K,C_{K+1}\}$ $(i=1\ldots K+1)$
2. Initialize a set of weights: $\mathbf{W}_i^c(1)=1$, i=1,...,N+Q
3. For $t=1\ldots M$:
   (a) Assign the new samples to the Positive cluster, according to the optimal class grouping selected on the step M in the previous joint boosting algorithm.
   (b) Classify the training data $\mathbf{X}$ using the decision stumps generated at the step $t$ of the previous joint boosting algorithm.
       $$h_t^n(\mathbf{x}_i,c)=\begin{cases} \alpha\delta(\mathbf{x}_i^j>\theta)+\rho, & \text{when } C_i \in \text{Positive(n)} \\ k^c, & \text{when } C_i \notin \text{Positive(n)} \end{cases} \quad (1)$$
   (c) Compute the weighted error for the class grouping as:
       $$Err_p=\sum_{c=1}^{K+1}\sum_{i=1}^{N+Q}\mathbf{W}_i^c(b_i^c-h_t^m(\mathbf{x}_i,c))^2. \quad (2)$$
       where $b_i^c \in \{-1,+1\}$ is the label assigned to $C_i$ in the m optimal binary grouping.
   (d) Assign the new samples to the Negative cluster, according to the optimal class grouping selected on the step M in the previous joint boosting algorithm, and compute the error $Err_n$ as in 2.
4. Assign the new class to the clustering with minimum error: $m=min(Err_p,Err_n)$;
5. Update the data weights: $\mathbf{W}_i^c(t+1)=\mathbf{W}_i^c(t)exp^{-b_i^c h_t^m(\mathbf{x}_i,c)}, \quad i=1,\ldots,N.$
6. Update the estimation for each class: $\mathbf{H}(\mathbf{x}_i,c)=\mathbf{H}_i^c(t)+h_t^m(\mathbf{x}_i,c)$
7. Output the estimation of each sample for each possible class.
```

Figure 1: Algorithm to add new samples to a previously trained joint boosting algorithm.

approximation ($O(K^2)$), where the grouping is performed as follows:

1. Train a weak learner using a single class as positive. For each feature a decision stumps classifier is trained, keeping the one that minimizes the error criterion.
2. Select the class with minimum weighted classification error as the initial Positive cluster.
3. For a the remaining $K-1$ classes:
   - Train a classifier using the previous Positive cluster but adding another class from the Negative cluster.
   - Add to the previous Positive cluster the class from the Negative cluster only if the joint selection improves the previous classification error.

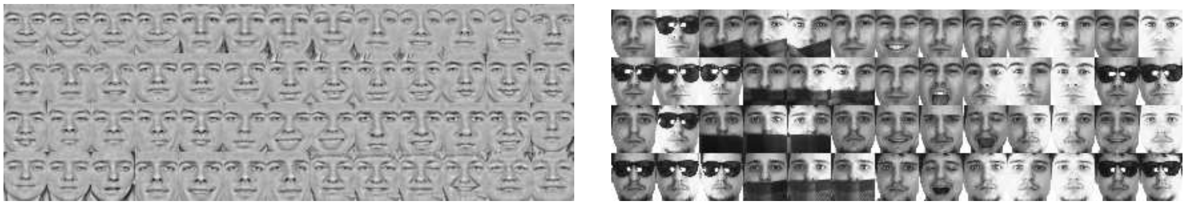
The class grouping with minimum error is selected, and at each boosting step the set of weights $\mathbf{W_i^c}$ are adjusted according to the partial classification results. Note that the optimal grouping is different at each step, given that the error criterion is computed taking into account the weights that focus the cluster selection on the most difficult samples. The grouping step allows the transfer of knowledge among several recognition tasks.

2.2 Adding new classes to the system

Once the joint boosting algorithm is trained, it can be used to classify a K-class problem. In a face recognition environment this fact means that it can only be used to recognize K people. When the subject $K+1$ is admitted in the system, the whole learning process must be retrained. We propose to take benefit of the class grouping performed in the sharing features step in order to incorporate online new classes to the system, avoiding the expensive relearning step.

The training algorithm can be divided in two steps: in the first one the joint boosting algorithm is run. The second step takes as input the Q samples of the new training class $\mathbf{X_{N+1,\ldots,N+Q}}$ and the corresponding label C_{k+1}, and runs M rounds of the algorithm shown in figure 1.

The idea is to add the new class to the system taking advantage of the previous shared feature space defined in the classification of the known tasks in the first step. Provided an optimal binary grouping at each step trained according a large enough number of samples and classes, we assign the new class samples to the positive or negative cluster minimizing the error criterion. The algorithm is iterated the same fixed



(a) Some faces from the FRGC. (a) Some faces from the AR Face.

Figure 2: Examples of faces from the FRGC and AR Face databases.

amount of times M. Notice that the method allows the inclusion of many new tasks, given that the same process can be iteratively repeated adding a new class each time. This approach is computationally fast, avoiding the most computationally expensive step of finding the optimal binary subgroup.

3 Experiments

The experiments have been performed using two different face databases: the Face Recognition Grand Challenge (Phillips et al., 2005), and the AR Face database (Martinez and Benavente, 1998). The idea of the experimental section is to show the evolution of the performance of our proposal as new classes are added to the system. We compare our proposal with a variation of the classic eigenface approach (Turk and Pentland, 1991) followed by a NN classification rule. We use PCA to extract 500 features from each data set, and then a discriminant analysis step is performed to obtain the 200 final features from each example. The NDA algorithm has been used for this purpose, which has been shown to improve the performance of other classic discriminant analysis techniques (Bressan and Vitria, 2003) under the NN rule. The new classes are added by projecting the training vectors on the reduced space, and using this projected features as a model for the new classification task.

Images from both data sets have been previously converted from the original RGB space to gray scale. Then we perform a geometric normalization using the center coordinates of each eye. Images have been rotated and scaled according to the inter-eye distance, in such a way that the center pixel of each eye coincides in all of them. The samples were then cropped obtaining a 37×33 thumbnail, therefore only the internal region of the faces has been preserved. The final sample from each image is encoded as a 1221 feature vector. In Figure 2 some examples from both databases are shown. From each data set, we have used only images from subjects that contain at least 20 samples (10 for training and the rest for testing).

3.1 Results

The experiments have been repeated 10 times, the results shown in table 1 are the mean accuracies of each method. The 95% confidence intervals are also shown near each value. The experimental protocol follows these steps for each database: (1) We randomly take 25 classes (people) from the data set. (2) We learn a classifier using 10 training samples from the 25 people. (3) We progressively add a new class without retraining the system. The remaining samples from each class are used for testing the resulting classifier.

The results with the FRGC database show an accuracy close to 98% using our boosted approach for the initial problem with 25 classes, while the application of feature extraction methods with the NN classifier obtains an initial 92%. This experiment suggests that for a perfectly acquired and normalized set, the use of shared boosting is the best option for multiclass face problems. Figure 3 shows the accuracies as a function of the number of classes added. The accuracy on the first 25 steps remains constant given that the classifier is initially trained on this subset. Notice that from that point the accuracy decreases, as expected, when new classes are added to the system. This fact is due to 2 reasons: first, usually the more classes has a classification problem the more decreases the accuracy, and second, when new samples are added to the system, there is an implicit error given that the classifier has not been retrained. Nevertheless, the accuracy does not decrease drastically, even when we increase the number of classes an 800%.

On the other hand, we also show the absolute and relative loss of accuracy as new classes are added (see Table 1). For each data set we add up to the maximum number of classes (160 and 86 for the FRGC and AR Face respectively) and take the resulting accuracy. The absolute decrease is computed as the accuracy using 25 classes minus the accuracy using the maximum number of classes. The relative decrease is computed as the absolute decrease divided by the initial accuracy considering the 25 classes. Using our

Table 1: Mean accuracy of the discriminant method and our proposal on two face databases. Only 25 classes are used for training, a total of 135 extra classes have been added in the FRGC case, and 59 in the AR Face. We show the decrease (absolute and relative percentage) in the mean accuracy from the first experiment with only 25 classes and the largest extended problem.

Data set	Acc. FE	Decrease	Relative	Acc. Adding-Shared	Decrease.	Relative
FRGC	0.8554 ± 0.0193	0.0548	6.0%	0.9214 ± 0.0024	0.0567	5.8%
AR Face	0.6013 ± 0.0020	0.2105	25.9%	0.7515 ± 0.0034	0.1062	12.4%

approach the accuracy decreases less, specially in the case of the AR Face data set, obtaining a more robust classification rule in presence of occlusions and strong changes in the illumination.

The main advantage using our adding-class approach, is the reduction on the computational needs. It has been shown experimentally that the use of joint boosting achieves high accuracies in face classification. Nevertheless, the computational cost makes the method unfeasible when the problem has too many classes. The clustering step to build binary problems at each boosting round is $O(K^2)$ using the BFS approach. This computational complexity is avoided when we use our proposal of adding new classes to the system. Typically, training the shared boosting algorithm using an initial set of 25 classes takes 8 hours on a Pentium IV computer (using the Matlab software). Training the same algorithm using 80 classes can take weeks, while extending the previous 25 class problem to the new 80 class problem using our approach takes a few minutes.

4 Conclusions

We propose a method to online add new classes to the joint boosting classifier in order to solve real world face recognition problems. We incrementally add a new class to the system extending the classifier in order to take into account the new binary classification task. The multiclass problem is seen as a set of multiple binary classification tasks that are trained sharing the feature space. The resulting classifier can be extended adding a new class avoiding the computationally expensive retraining process which could be computationally unfeasible in a large class problem. Nevertheless, it can be trained using less classes at first instance, and then extended using our approach to solve the real large problem.

We have experimentally validated our proposal using two different face databases: the FRGC database acquired in a controlled environment, and the AR Face database which contains important artifacts due to strong changes in the illumination and occlusions. The results show that the classification accuracy decreases less drastically than using the classic NN rule used in online learning methods when the original sets are extended to large class problems (up to 8 times the original class set size).

We plan as future work to analyze the importance of the classes chosen in the original trained algorithm. A diverse choice of the classes should allow a more general base for extending the classifier. The use of a training and an extra validation set could improve slightly the accuracies. The initial number of classes used to train the system influence the performance of the extended classifier.

Acknowledgments

This work is supported by the TIN2006-15308-C02-01 project, Ministerio de Educación y Ciencia, Spain.

REFERENCES

Baxter, J. (2000). A model of inductive bias learning. *Journal of Machine Learning Research*, 12:149–198.

Bressan, M. and Vitria, J. (2003). Nonparametric discriminant analysis and nearest neighbor classification. *Pattern Recognition Letters*, 24(15):2743–2749.

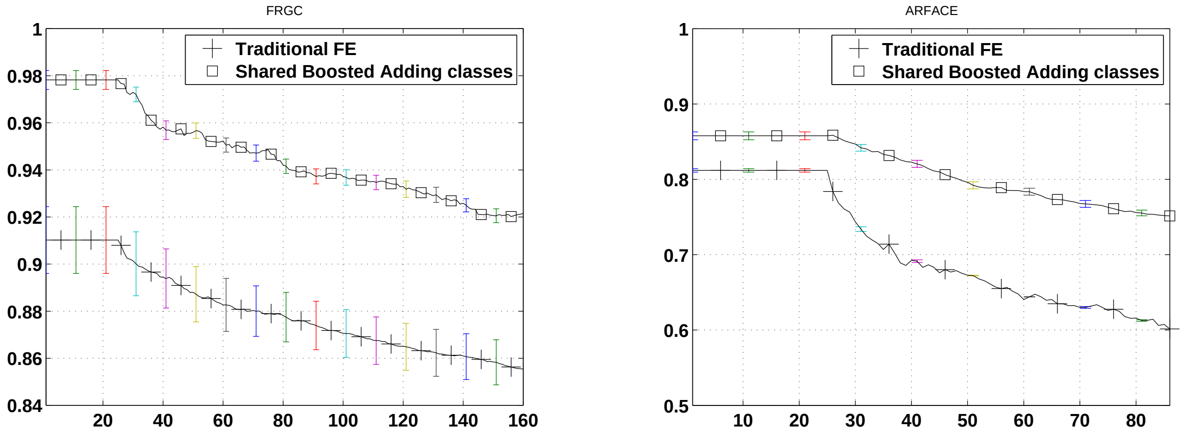
Caruana, R. (1997). Multitask learning. *Machine Learning*, 28(1):41–75.

Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Ann. Eugenics*, 7:179–188.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156.

Friedman, J., T.Hastie, and R.Tibshirani (2000). Additive logistic regression: a statistical view of boosting. *Annals of statistics*, 28:337–374.

Fukunaga, K. and Mantock, J. (1983). Nonparametric discriminant analysis. *IEEE Transactions*



(a) Accuracy using the FRGC data set.

(b) Accuracy using the ARFace data set.

Figure 3: Accuracy as a function of the number of classes

on Pattern Analysis and Machine Intelligence, 5(6):671–678.

Ji Zhu, Saharon Rosset, H. Z. and Hastie, T. (2006). Multi-class adaboost. Technical report, Standford University.

Martinez, A. and Benavente, R. (1998). The AR Face database. Technical Report 24, Computer Vision Center.

Masip, D., Kuncheva, L. I., and Vitria, J. (2005). An ensemble-based method for linear feature extraction for two-class problems. *Pattern Analysis and Applications*, 8:227–237.

Phillips, P. J., Flynn, P. J., Scruggs, T., Bowyer, K. W., Chang, J., Hoffman, K., Marques, J., Min, J., and Worek, W. (2005). The 2005 IEEE workshop on face recognition grand challenge experiments. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, page .45, Washington, DC, USA. IEEE Computer Society.

Schapire, R. E. (1997). Using output codes to boost multiclass learning problems. In *Proc. 14th International Conference on Machine Learning*, pages 313–321. Morgan Kaufmann.

Torralba, A., Murphy, K., and Freeman, W. (2004). Sharing features: efficient boosting procedures for multiclass object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86.

Yokono, J. J. and Poggio, T. (2006). A multiview face identification model with no geometric constraints. Technical report, Sony Intelligence Dynamics Laboratories.