Fast Classification in Incrementally Growing Spaces

Oscar Déniz-Suárez[1], Modesto Castrillón[2], Javier Lorenzo[2],
Gloria Bueno[1], and Mario Hernández[2]

[1] E.T.S.I.Industriales, Universidad de Castilla-La Mancha
Avda. Camilo Jose Cela s/n, 13071 Ciudad Real, Spain
[2] Universidad de Las Palmas de Gran Canaria. Dpto. Informatica y Sistemas.
Edificio de Informatica, Campus de Tafira, 35017 Las Palmas, Spain

Abstract. The classification speed of state-of-the-art classifiers such as SVM is an important aspect to be considered for emerging applications and domains such as data mining and human-computer interaction. Usually, a test-time speed increase in SVMs is achieved by somehow reducing the number of support vectors, which allows a faster evaluation of the decision function. In this paper a novel approach is described for fast classification in a PCA+SVM scenario. In the proposed approach, classification of an unseen sample is performed incrementally in increasingly larger feature spaces. As soon as the classification confidence is above a threshold the process stops and the class label is retrieved. Easy samples will thus be classified using less features, thus producing a faster decision. Experiments in a gender recognition problem show that the method is by itself able to give good speed-error tradeoffs, and that it can also be used in conjunction with other SV-reduction algorithms to produce tradeoffs that are better than with either approach alone.

Keywords: gender recognition, Support Vector Machines, Principal Component Analysis, Eigenfaces.

1 Introduction

One of the most frequent classification systems encountered in research is the combination of PCA (Principal Component Analysis) and SVM (Support Vector Machines). PCA is frequently used because of its simplicity and relative effectiveness, while SVM have already demonstrated impressing classification capabilities. The two techniques have been used together for face recognition and verification, face detection, biosignal (i.e. EEG, ECG, EMG, CT scans...) classification, operations research, part inspection, biochemistry, anomaly detection, text categorization, medicine composition analysis, etc. For a comprehensive list of SVM applications the reader is referred to [1].

Despite the power of SVMs, they are orders of magnitude more costly at query-time than other popular machine learning alternatives such as decision trees and neural networks [2]. Classification speed is crucial for learning problems that use a large number of samples, like in emerging data mining applications. In some domains the amount of data available is growing at exponential rates, especially with the advent of global networks and the possibility of ubiquitous generation of data. Human-computer and human-robot interaction applications also need to produce fast responses, as for example in phoneme classification. Low computational complexity is also required for embedded and mobile systems, where available resources are rather limited.

Most of the research carried out in fast classification kernel machines has involved reducing the number of support vectors [3]. Such reduction can be achieved by approximating the discriminating hypersurface to a user-specified accuracy. In [4] the approach taken was to reduce the complexity of the generated hypothesis by excluding some training samples, specifically subsets of the support vectors obtained in the first place. In a similar fashion, [5] is based on stopping the evaluation of support vectors of the hypothesis when the confidence of the result (measured by the partial classification result) is above a threshold. This requires the support vectors of the hypothesis to be ordered by decreasing importance. In [6] pairs of close support vectors are iteratively substituted by a new one. Similarly, in [7] the decision function is simplified by removing support vectors that contribute less to the decision.

As shown above, most research in fast kernel machines has involved selecting subsets of support vectors (or training samples in general). This paper describes a framework for fast classification in PCA+SVM systems in which classification is performed incrementally in increasingly larger feature spaces. As soon as the classification confidence is above a threshold the process stops and the class label is retrieved. Fast classification is not achieved by using less support vectors, but by classifying in simpler spaces, which reduces the number of computations. Section 2 explains the common PCA+SVM setting encountered in supervised learning problems and describes the method proposed. Experimental results are shown in Section 3. Finally, the main conclusions and lines of future work are outlined.

2 Fast Classification in PCA+SVM Settings

PCA is often used to project input samples to a (generally lower dimensional) space where classification is carried out. This is specially useful when the input samples are images. Basically, PCA gives a set of orthogonal dimensions that maximize the variance of the input samples. In face recognition, this set is called *eigenfaces*, see [8]. Not all of these dimensions (eigenfaces) are useful for classification. Only the first n eigenfaces are appropriate for classification, with the last eigenfaces typically encoding noise.

When a test sample X is to be projected with PCA, the operation to perform is: $Y = XW$, where W is the transform matrix. When working with vectorized images in the rows of X, the columns of W are the eigenfaces. As mentioned above, usually only the first n columns of W are used in the multiplication. This is thought to avoid the noise of the last eigenfaces. What should be a good value for n? There are reasons to believe that a large dimensional input space would be needed to separate difficult samples, while 'easy' samples could be separated in simpler spaces (i.e. lower value for n). In the method proposed, a different K value may be used for each specific test sample, instead of a fixed dimension n. *Easy* samples can be classified in a PCA space of a low number K of dimensions. The necessary number of dimensions to use will be ultimately given by the classifier output. For each test sample the system would classify it in a low dimensional space first. If the classifier output is large enough (i.e. above a fixed threshold) then classification will end and a class label will be retrieved. Otherwise the process should be repeated in a more informative space of a larger dimension, see Fig. 1.

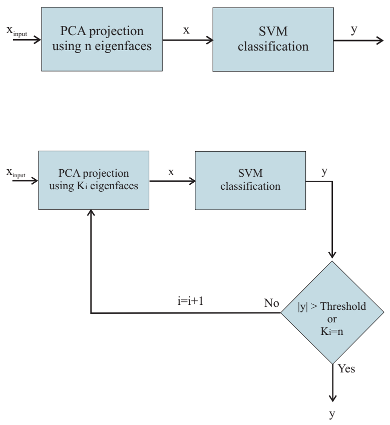

Fig. 1. Top: Typical PCA+SVM classification procedure for a test sample. Bottom: Fast PCA+SVM classification method for a test sample.

The loop of Figure 1 would have to be incremental in terms of computational cost. Otherwise there would not be any speed gain over the use of a fixed dimension. The PCA projection of the input sample can be done incrementally, since it is a matrix multiplication (see above).

It can be shown that SVM classification can be also made incremental in the input space dimension as long as the new dimensions at each step are orthogonal to the previous ones, which is the case when using PCA. Kernels typically used (like polynomial, RBF and sigmoid) are functions either of a dot product or a norm of samples. When classifying a sample, the cost of the kernel evaluations is therefore dependent on the space dimension. For a given input sample, let us suppose that classification has been already made in a space of dimension K_{i-1}. Therefore, we have already evaluated the kernel values $\kappa(\mathbf{x}, \mathbf{x_i})$. Let us suppose that we set to classify the same sample in a space of dimension $K_i > K_{i-1}$. Here the input and training samples can be respectively represented as $\mathbf{x} + \Delta\mathbf{x}$ and

$\mathbf{x_i} + \Delta\mathbf{x_i}$, where vectors $\mathbf{x}$ and $\mathbf{x_i}$ are augmented with zeros in order to have K_i components. $\Delta\mathbf{x}$ and $\Delta\mathbf{x_i}$ represent the values of the new $\Delta K = K_i - K_{i-1}$ dimensions, with the other components set to zero.

With the assumption imposed on the input space, in this space of K_i dimensions the following orthogonality relations hold: $\mathbf{x}\perp\Delta\mathbf{x}$, $\mathbf{x}\perp\Delta\mathbf{x_i}$, $\mathbf{x_i}\perp\Delta\mathbf{x}$, $\mathbf{x_i}\perp\Delta\mathbf{x_i}$. Using these orthogonality relations two cases are now possible:

- Dot product-based kernels (for example the polynomial kernel $\kappa(\mathbf{x}, \mathbf{x_i}) = (\mathbf{x} \cdot \mathbf{x_i} + 1)^p$):

$$\begin{aligned}(\mathbf{x} + \Delta\mathbf{x}) \cdot (\mathbf{x_i} + \Delta\mathbf{x_i}) = \mathbf{x} \cdot \mathbf{x_i} + \mathbf{x} \cdot \Delta\mathbf{x_i} + \Delta\mathbf{x} \cdot \mathbf{x_i} + \Delta\mathbf{x} \cdot \Delta\mathbf{x_i} = \\ = \mathbf{x} \cdot \mathbf{x_i} + \Delta\mathbf{x} \cdot \Delta\mathbf{x_i} \qquad (1)\end{aligned}$$

- Norm-based kernels (for example the RBF kernel: $\kappa(\mathbf{x}, \mathbf{x_i}) = \exp(-||\mathbf{x} - \mathbf{x_i}||^2/2p^2)$):

$$\begin{aligned}||(\mathbf{x} + \Delta\mathbf{x}) - (\mathbf{x_i} + \Delta\mathbf{x_i})||^2 = ||\mathbf{x} + \Delta\mathbf{x}||^2 + ||\mathbf{x_i} + \Delta\mathbf{x_i}||^2 - 2(\mathbf{x} + \Delta\mathbf{x}) \cdot (\mathbf{x_i} + \Delta\mathbf{x_i}) = \\ = ||\mathbf{x}||^2 + ||\Delta\mathbf{x}||^2 + 2\mathbf{x} \cdot \Delta\mathbf{x} + ||\mathbf{x_i}||^2 + ||\Delta\mathbf{x_i}||^2 + 2\mathbf{x_i} \cdot \Delta\mathbf{x_i} - 2(\mathbf{x} + \Delta\mathbf{x}) \cdot (\mathbf{x_i} + \Delta\mathbf{x_i}) = \\ = ||\mathbf{x}||^2 + ||\Delta\mathbf{x}||^2 + ||\mathbf{x_i}||^2 + ||\Delta\mathbf{x_i}||^2 - 2(\mathbf{x} \cdot \mathbf{x_i} + \mathbf{x} \cdot \Delta\mathbf{x_i} + \Delta\mathbf{x} \cdot \mathbf{x_i} + \Delta\mathbf{x} \cdot \Delta\mathbf{x_i}) = \\ = ||\mathbf{x} - \mathbf{x_i}||^2 + ||\Delta\mathbf{x}||^2 + ||\Delta\mathbf{x_i}||^2 - 2\Delta\mathbf{x} \cdot \Delta\mathbf{x_i} \qquad (2)\end{aligned}$$

It can be seen that the computations are based on the dot product or norm of the previous step plus some terms that can be computed with a constant cost proportional to ΔK. Thus, in both cases the computation can be done incrementally.

Note that the training cost of the proposed method is the same as in a non-incremental classifier, only one training stage is carried out using a space of whatever dimension n. Once we have a trained classifier, the proposed method only works at test time, where we have the incremental classification of samples.

3 Experiments

Since the speed gain in the proposed method is based on classifier confidence, we will have a trade-off between classification speed and error. In this respect, the main performance indicator that will be used here is the error-speedup curve, which represents test error as a function of classification speed gains. This curve is obtained by varying the classifier confidence threshold (see Figure 1), with values ranging from 0 to 1. An RBF kernel was used in all the experiments.

The question arises whether the proposed dimensionality reduction strategy can be compared with SV-reduction. Note that there are cases in which one reduction strategy will always be superior to the other and vice versa. For the dimensionality reduction approach the results will depend on the number n of dimensions used (i.e. the size of the feature space). On the other hand, the performance of SV-reduction methods depends on the number of support vectors which in turn depends on the parameters used for training the classifier (i.e. the kernel parameter 'p'). The best values for these parameters depend on the problem at hand (and also on the number of training samples available). For large values of n, for example, the proposed dimensionality reduction method should give better error-speedup curves than the SV reduction method. For small values of n the reverse will be true. For these reasons, a direct comparison is not appropriate. Instead, we focused on combining the two strategies to test whether better net results can be obtained.

The combination implies progressively reducing both the number of support vectors and dimensions, following some order (i.e. choosing at each step between SV-reduction or dimensionality reduction). Searching for the ordering that gives the optimal error-speedup curve is not feasible, since there is a factorial number of orderings.

Assuming independence between both reduction methods, approximations can be obtained. In our case, a simple greedy search was carried out in the validation set. The search involves choosing between reducing the dimension or reducing the number of support vectors, at each step of the curve. The selection is made according to the error decrease in the validation set produced by each option.

The proposed strategy was used in conjunction with the SV-reduction method described in [5], in which classification speed is improved by using only the most important support vectors in the classification function evaluation. In order to achieve this, the support vectors are ordered by the absolute value of the associated coefficient. With that algorithm, important computational savings can be achieved without significant degradation in terms of recognition accuracy.

A gender recognition scenario was used in the experiments, using the typical PCA+SVM combination. A number of face images were captured in our laboratory. These included talking and changes in facial expression, illumination and pose. A total of 7256 male+7567 female images were gathered, and later normalized in rotation and size to 39x43 pixels. In each run of the experiment, 300 of these images were randomly selected and randomly partitioned in a training set of 120 images (60+60), a validation set of 80 images (40+40) and a test set of 100 images (50+50).

PCA was previously applied over an independent set of 4000 face images taken from the Internet. The eyes in each image were located manually and then the image was normalized to 39x43. PCA was computed over this set of 4000 normalized images, retaining a number n of coefficients, see Figure 2. The collected images were all projected onto this PCA space previous to training and classifying.

Even though our goal in this work was not to obtain better absolute recognition values, we wanted to test the algorithms in an independent database, a subset of frontal faces of the FERET [9] data set was also considered. In this case the working set was made up of a total of 177 male+177 female faces, normalized to 52x60 pixels. In each experiment a random ordering of the samples of each class was performed. PCA was applied to the first 144 of them (77+77). The training set had 120 samples (60+60), the validation set 40 (20+20) and the test set the other 40 (20+20).

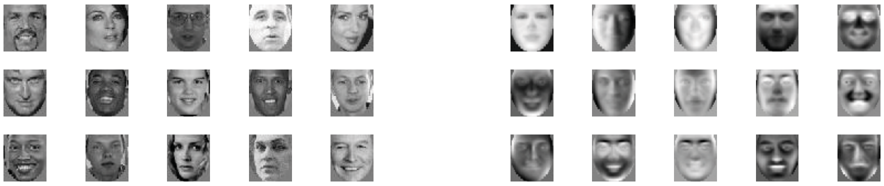

Fig. 2. Left: some Internet faces, as detected and normalized. Right: the first 36 eigenfaces obtained with PCA.

The experiments described here show the effects of the combination in two different cases: a) dimensionality reduction performing better than SV-reduction and b) SV-reduction performing better than dimensionality reduction. These cases were achieved by adjusting the parameter p of the support vector classifier and the value of n. The speedups were calculated as 100 minus the percentage of numerical operations used for classification, with respect to the initial case of dimensionality n and number of support vectors. Figure 3 shows the performance for case a), i.e. dimensionality-reduction method better than SV-reduction.

Note that for large values of n (as was the case in Figure 3) the dimensionality reduction curve is more horizontal, which makes it difficult to obtain significant improvements with the combination. This occurs because with the validation set the greedy algorithm could choose to reduce support vectors at a given point when in fact the best option is to keep on reducing dimensionality (thus keeping the error constant most of the time). This causes the performance of the combination to be worse than with either of the two methods, especially if the validation set is small. Since we have a validation set available, in such cases it

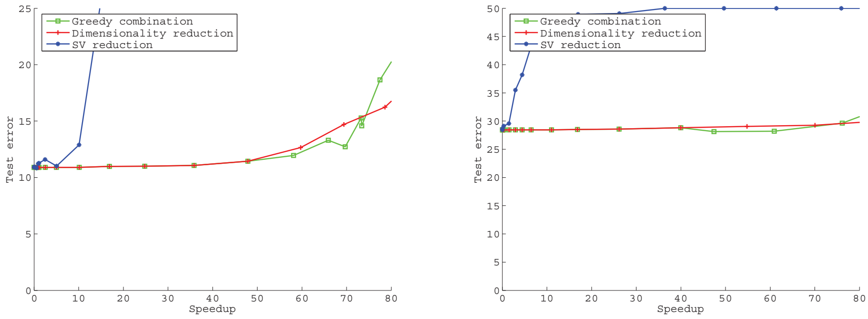

Fig. 3. Left: Error-speedup curves for the three methods considered, using our laboratory database. 40 runs, random distributions of the samples in training, validation and test sets. Kernel parameter $p = 3000$. The (initial) dimensionality n was calculated as that which accounted for a 90% of the total variance. Right: Error-speedup curves for the three methods considered, using the FERET database. Same conditions except for kernel parameter $p = 10000$.

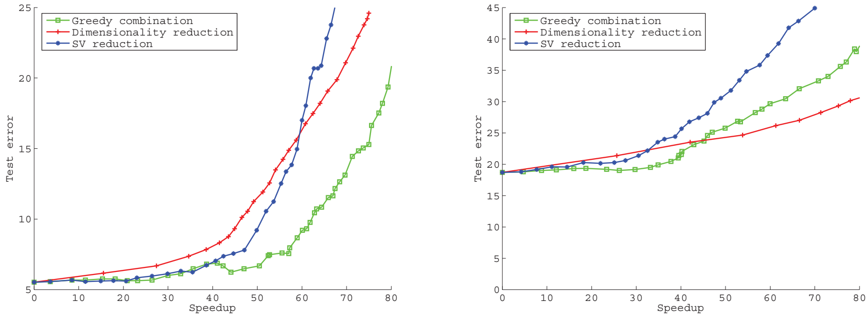

Fig. 4. Left: Error-speedup curves for the three methods considered, using our laboratory database. 25 runs, random distributions of the samples in training, validation and test sets. Kernel parameter $p = 500$. The (initial) dimensionality n was 4. Right: Error-speedup curves for the three methods considered, using the FERET database. 50 runs, random distributions of the samples in training, validation and test sets. Kernel parameter $p = 1500$. The (initial) dimensionality n was 25.

may be useful to set a threshold in the greedy search so that SV-reduction is used only when large errors begin to appear with dimensionality reduction. Alternatively, SV-reduction could be made to proceed only after the speedup gain has reached a given point, which can be estimated (manually) with the validation set. The latter option was used in Figure 3, where SV-reduction only acted after a speedup of 60% and 40% was reached using dimensionality reduction.

Figure 4 shows the performances for case b), i.e. SV-reduction better than dimensionality-reduction. For the FERET images it was very difficult to find a set of parameter values that made the SV-reduction method be clearly better than dimensionality reduction. We postulate that this was due to the fact that this data set was considerably more difficult (the images were larger, the PCA space was obtained with fewer samples, many races were present, more significant illumination variations, ...), which would have made the obtained support vectors more critical for classification. Still, the figure shows how the greedy algorithm allows to obtain an improvement for speedups between 10-40%, although after that point the performance of the combination obviously turns worse than with dimensionality reduction alone. Overall, the results shown above suggest that even with a simple greedy combination a better net performance can be achieved. With more computational effort better combinations could be used that take advantage of the (in)dependence between feature space size and classifier size.

4 Conclusions

The test speed of state-of-the-art classifiers such as SVM is an important aspect to be considered for certain applications. Usually, the reduction in classification complexity in SVMs is achieved by reducing the number of support vectors used in the decision function. In this paper a novel approach has been described in which the computational reduction is achieved by classifying each sample with the minimum number of features necessary (note that the typical setting is to use a fixed dimension for the input space). Experiments in a gender recognition problem show that the method is by itself able to give good speed-error trade-offs, and that it can also be used in conjunction with support vector-reduction algorithms to produce trade-offs that are better than with either approach alone.

Acknowledgments

This work was partially supported by project PII2I09-0043-3364 of Castilla-La Mancha Regional Government and the Spanish Ministry of Science and Innovation funds (TIN2008-06068).

References

1. Guyon, I.: SVM application list (2010), http://www.clopinet.com/isabelle/Projects/SVM/applist.html
2. Decoste, D., Mazzoni, D.: Fast query-optimized kernel machine classification via incremental approximate nearest support vectors. In: International Conference on Machine Learning, pp. 115–122 (2003)
3. Fehr, J., Zapien, K., Burkhardt, H.: Fast support vector machine classification of very large datasets. In: Proceedings of the GfKl Conference, Data Analysis, Machine Learning, and Applications. LNCS. Springer, University of Freiburg, Germany (2007)
4. Zhana, Y., Shen, D.: Design efficient support vector machine for fast classification. Pattern Recognition 38(1), 157–161 (2005)
5. Arenas-García, J., Gómez-Verdejo, V., Figueiras-Vidal, A.R.: Fast evaluation of neural networks via confidence rating. Neurocomput 70(16-18), 2775–2782 (2007), http://dx.doi.org/10.1016/j.neucom.2006.04.014
6. Nguyen, D., Ho, T.: An efficient method for simplifying support vector machines. In: Procs. of the 22nd Int. Conf. on Machine Learning, pp. 617–624 (2005)
7. Guo, J., Takahashi, N., Nishi, T.: An efficient method for simplifying decision functions of support vector machines. IEICE Transactions 89-A(10), 2795–2802 (2006)
8. Turk, M.A., Pentland, A.: Eigenfaces for Recognition. Cognitive Neuroscience 3(1), 71–86 (1991), ftp://whitechapel.media.mit.edu/pub/images/
9. Phillips, P., Moon, H., Rizvi, S., Rauss, P.: The FERET evaluation methodology for face-recognition algorithms. TPAMI 22(10), 1090–1104 (2000)