COMBINING SVMS FOR FACE CLASS MODELING

Julien Meynet, Vlad Popovici, Matteo Sorci and Jean-Philippe Thiran

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Institute
CH-1015 Lausanne, Switzerland
{Julien.Meynet,Vlad.Popovici,Matteo.Sorci,JP.Thiran}@epfl.ch
http://itswww.epfl.ch

ABSTRACT

We present a method for combining a number of Support Vector Machines trained independently in the eigenface space and we apply it to face class modeling. We first train several SVMs on subsets of some initial training set and then combine their expertise using various probabilistic combining rules. This approach is compared to a classical SVM classification as well as Multiple SVM classification[1].

1. INTRODUCTION

Automatic face detection is the first step of face analysis and needs to be precise and robust as it directly affects the performance of the next stages of processing. The complexity of modeling the face class is due to the large intra-class variability, as faces are highly deformable objects whose appearances depend on numerous factors (lighting conditions, presence or absence of occluding objects, and so forth). Moreover, the non-face class is a very broad concept and modeling it proves to be very difficult.

In the last years, many methods have been proposed and we give hereafter a brief overview of some of the most significant of them.

There are two important aspects in a good face detection system: the speed of the detection and the classification performance. A fast algorithm, proposed by Viola and Jones[2], uses simple rectangular Haar-Like features boosted in a cascade structure. We have used this fast approach as a preprocessing step in order to reduce the search space. However, even if the cascade results in a very fast detector, the false positive rate remains too high.

The method reported by Rowley et. al. in [3] is one of the most representative for the class of neural network approaches. It comprises two modules: a classification module which hypothesizes the presence of a face and a module for arbitrating multiple detections.

Sung and Poggio have developed a clustering and distribution-based system for face detection [4]. There are two main components in their system: a model of the face/non–face patterns distribution and a decision making module. The two class distributions are each approximated by six Gaussian clusters.

Osuna et. al. developed a face detector based on SVM trained directly on the intensity patterns [5]. A brief description of the SVM is given in this paper also.

In [6], Popovici and Thiran proposed to model the face class using a SVM trained in eigenfaces space. They showed that even a very low dimensional space (compared with the original input space) suffices to capture the relevant information when used in conjunction with a powerful classifier, like a non linear SVM.

Then some studies tried to deal with large datasets by using experts trained on the original dataset. For example Bengio and al. [7] used parallel SVMs trained on subsets of large scale problems.

An improvement applied to face detection was presented in [1]. They use a mixture of SVMs (MSVM) to reduce the complexity of the problem for both training and testing. The idea is to split the original face training set into several subsets chosen either by random sampling or clustering. They train a SVM on each of the subsets and then train a second layer SVM to combine the outputs of the first layer SVMs. They showed that it also improves the generalization capabilities compared to a single SVM trained on the complete dataset.

We now propose another way of combining the expertise of the first layer SVMs. Instead of using directly the margins we use a probabilistic approach based on the estimation of the posterior probabilities output by the first layer SVMs. We then use basic probabilistic combination rules for taking the final decision. This idea of using simple combination rules to combine the decision of several classifiers has been studied in [8].

We will introduce the motivation for combining SVMs and we will justify its use both from a theoretical perspective and its efficiency in the case of face class modeling. In section 2 we will briefly review the SVM theory, the estimation of the posterior probabilities and then we will describe in detail the combination approach. The classifier will be trained on face and non face examples pre-processed by PCA, as described on section 2.1. Finally, in sections 3 and 4 we present some experiments and comparisons with classical SVM and we draw some conclusions.

2. MIXTURES OF SVMS

2.1 Construction of the Eigenfaces

As each image is made of a large number of pixels, we use Principal Component Analysis (PCA) to decrease the dimensionality of the image space. We first recall the definition of PCA complemented by the distance from feature space (DFFS) as discussed in [1].

2.1.1 Principal Component Analysis (PCA) and Eigenfaces

Let $\mathbf{x}_1, \ldots, \mathbf{x}_l \in \mathbb{R}^n$ be a set of $n-$dimensional vectors and consider the following linear model for representing them

$$\mathbf{x} = W_{(k)}\mathbf{z} + \mu$$

where $W_{(k)}$ is a $n \times k$ matrix, $\mathbf{z} \in \mathbb{R}^k$ and the columns of $W_{(k)}$ are given by the dominant k eigenvectors of the sample covariance matrix[1] $S = \frac{1}{l}\sum_l (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)'$ such that $S\mathbf{w}_j = \lambda \mathbf{w}_j$ and where μ is the sample mean. Some details about the estimation of the eigenfaces space dimensionality such as classification in eigenfaces space using SVMs are shown in [6]. This dimensionality reduction technique is very popular in face analysis where the principal directions are called *eigenfaces* [9],[10].

The distance between a given image and the face class is decomposed in two orthogonal components: the *distance in feature space* corresponding to the projection onto the lower dimensional space and the *distance from feature space (DFFS)(see Eq. 1)* accounting for the reconstruction error.

$$DFFS = \sqrt{\|\mathbf{x} - \mu\|^2 - \|\mathbf{z}\|^2} \tag{1}$$

Given this and considering that the DFFS still contains some useful information for classification, we can improve the discrimination power by adding the value of the DFFS to the projection vector. Thus considering that we keep 85% of total variance with the k first eigenvectors, we use the following vectors to perform the classification.

$$X = (x_1, \ldots, x_k, x_{k+1}),$$

where $x_1, \ldots, x_k$ represent the projection onto the $k-$ dimensional eigenfaces space and x_{k+1} the DFFS.

2.2 An overview of Classical SVM

Let us begin with a brief overview of the classical SVM algorithm. More information about SVM can be found in [11], [12].

Let $\{(\mathbf{x}_i, y_i) | i = 1, \ldots, l\} \subset \mathbb{R}^n \times \{-1, +1\}$ be a set of examples. From a practical point of view, the problem to be solved is to find that hyperplane that correctly separates the data while maximizing the sum of distances to the closest positive and negative points (i.e. *the margin*). The hyperplane is given by[2]:

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = 0$$

and the decision function is

$$f(\mathbf{x}) = \operatorname{sgn}(h_{\mathbf{w},b}(\mathbf{x})) = \operatorname{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b)$$

In the case of linearly separable data, maximizing the margins means to maximize $\frac{2}{\|\mathbf{w}\|}$ or, equivalently, to minimize $\|\mathbf{w}\|^2$, subject to $y_i(\langle \mathbf{w}, \mathbf{x} \rangle + b) \geq 1$. Suppose now that the two classes overlap in feature space. One way to find the optimal plane is to relax the above constraints by introducing the *slack variables* ξ_i and solving the following problem (using 2-norm for the slack variables):

$$\begin{aligned} &\min_{\xi, \mathbf{w}, b} \|\mathbf{w}\|^2 + C \sum_{i=1}^{l} \xi_i^2 \\ &\quad \text{subject to} \qquad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \quad \forall i = 1, \ldots, l \end{aligned}$$

where C controls the weight of the classification errors ($C = \infty$ in the separable case).

This problem is solved by means of Lagrange multipliers method. Let $\alpha_i \geq 0$ be the Lagrange multipliers solving the problem above, then the separating hyperplane, as a function of α_i, is given by

$$h_{\alpha_i, b}(\mathbf{x}) = \sum_{i:\alpha_i > 0} y_i \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b$$

Note that usually only a small proportion of α_i are non-zero. The training vectors $\mathbf{x}_i$ corresponding to $\alpha_i > 0$ are called *support vectors* and are the only training vectors influencing the separating boundary.

In practice however, a linear separating plane is seldom sufficient. To generalize the linear case one can project the input space into a higher–dimensional space in the hope of a better training–class separation. In the case of SVM this is achieved by using the so–called "kernel trick". Basically, it replaces the inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ with a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$. As the data vectors are involved only in this inner products, the optimization process can be carried out in the feature space directly. Some of the most used kernel functions are:

$$\begin{aligned} \text{the polynomial kernel} \qquad & K(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + 1)^d \\ \text{the RBF kernel} \qquad & K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|^2) \end{aligned}$$

2.3 Conditional Probabilities Estimation

The decision function $f(\mathbf{x}) = \operatorname{sgn}(h_{\mathbf{w},b}(\mathbf{x}))$ of the SVM is directly based on the uncalibrated value of the margin. However in order to efficiently use the quantity output by the SVM it would be interesting to have a posterior probability output. This is exactly what we need for the combination of the SVMs in our study. One way of transforming the margins in posterior probabilities consists in training directly a kernel classifier using maximum likelihood. A more appropriate method was proposed by Platt in [13]. He uses a sigmoid function to map the margins into probabilities. The advantage of this technique is that we directly obtain the posterior probabilities $P(c = +1|f)$ instead of estimating the class conditional densities. Eq. 2 shows the form of such a sigmoid:

$$P(c = +1|f) = \frac{1}{1 + \exp(Af + B)}. \tag{2}$$

The parameters A and B are trained using maximum likelihood estimation from the training set, see [13] for details. An interesting and important point is that the error function is not changed by this calibration process.

2.4 Combining SVMs

SVM techniques are well known since a few years for many reasons, among them their generalization capabilities. However, as explained in the previous subsection, training a SVM usually requires solving a quadratic optimization problem, which means it also varies quadratically with the number of training examples. We know by experience that due to the large variability of both face and non face classes, building a face detection system requires a large amount of examples. In order to improve the speed of the process, we use a parallel structure of SVMs similar to the one introduced in [7]

[1] We denote with a prime symbol the transpose of a matrix or a vector.

[2] We use $\langle \cdot, \cdot \rangle$ to denote the inner product operator

and [1]. We decompose the initial training dataset into several subsets (either by random sampling or by clustering, as in [1]) and train a SVM with each of these subsets. Then the final decision is made accordingly to some combination rules based on the posterior probabilities estimated from the margins output by the SVMs using Eq. 2.

By reducing the size of the training data we also decrease significatively the complexity of the optimization problem and also reduce the influence of eventual outliers or noise in the initial database.

We now focus our work on the combination of the first layer classifications. In [1] a second layer SVM was used to combine the decisions. It was directly trained on the margins output by the first layer SVMs. For this another independent subset was also needed. However in this scheme a useful information has been ommited: We know that more than the margins, we can estimate the posteriori probabilities of the first layer SVMs. This calibration of the outputs allows us to directly combine them using basic rules in order to make the final decision.

Let us recall the context. Consider that we want to classify a pattern $\mathbf{x}$ in one of the C classes $(c_1,...,c_C)$. We model each of the C classes by the probability density functions $p(\mathbf{x}|c_k)$ and its a priori probability by $P(c_k)$. Assume that we have N SVMs. Thus denote by $p_j(c_k|\mathbf{x})$ the posterior probability estimated from the j-th SVM that $\mathbf{x}$ belongs to class c_k. The Bayes decision rule acesses that an example $\mathbf{x}$ is assigned to the class c_i if:

$$P(c_i|\mathbf{x}) > P(c_k|\mathbf{x}), \text{ for } k = 1,...,C; k \neq i \tag{3}$$

Eq. 3 relies on the theoretical framework of the classification task but its computation is very difficult in practice. That is why we simplify the problem by using some basic combination rules easier to compute. In our study we focus on the six following rules:

- Product rule: Example $\mathbf{x}$ is assigned to the class c_i if for $k = 1,...,C; k \neq i$:

$$P(c_i)\prod_{j=1,..,N} p_j(\mathbf{x}|c_i) > P(c_k)\prod_{j=1,..,N} p_j(\mathbf{x}|c_k) \tag{4}$$

  This rule derives directly from Bayes theorem by assuming that the measurements of the different classifiers are conditionally independent;
- Sum rule: Example $\mathbf{x}$ is assigned to the class c_i if for $k = 1,...,C; k \neq i$:

$$(1-N)P(w_i) + \sum_{j=1,..,N} P_j(c_i|\mathbf{x}) > (1-N)P(w_k) + \sum_{j=1,..,N} P_j(c_k|\mathbf{x}) + \sum_{j=1,...,N} P_j(c_k|\mathbf{x}) \tag{5}$$

Then from 4 and 5 we derivate four other combination rules. In all the cases, Example $\mathbf{x}$ is assigned to the class c_i if for $k = 1,...,C; k \neq i$:

- Max rule:

$$\max_{j=1,...,N} P_j(c_i|\mathbf{x}) > \max_{j=1,...,N} P_j(c_k|\mathbf{x})$$

- Min rule:

$$\min_{j=1,...,N} P_j(c_i|\mathbf{x}) > \min_{j=1,...,N} P_j(c_k|\mathbf{x})$$

- Median rule:

$$\text{median}_{j=1,...,N} P_j(c_i|\mathbf{x}) > \text{median}_{j=1,...,N} P_j(c_k|\mathbf{x})$$

- Majority vote:

$$\sum_{j=1,...,N} d_{i,j}(\mathbf{x}) > \sum_{j=1,...,N} d_{k,j}(\mathbf{x})$$

  where $d_{i,j} = 1$ if classifier j assigns measurement $\mathbf{x}$ to class c_i and 0 otherwise.

In the following, we only consider the binary classification problem ($C = 2$)with two classes: face images and non face images.

3. EXPERIMENTS AND RESULTS

In order to test the Combined SVMs using the previously introduced combining rules, we did some experiments in the framework of face detection. We first collected face images from some classical face databases: BANCA [14], XM2VTS[15], BioID[16], FERET[17].

From each of the database, we extracted roughly two images per identity and then cropped the images to 19x19 pixels grayscale images. We thus obtained 3708 face images for the training and 4295 faces for testing. The non faces examples were chosen by bootstrapping on randomly selected images. As face representation we used the PCA decomposition complemented by the distance from feature space. From the 361 input pixels reduce the dimensionality to 15 by PCA decomposition as described in section 2.1. The dimensionality of the eigenfaces space was chosen by keeping 85% of total variation. Then adding the DFFS value to the projection onto the eigenfaces space yields to a 16 dimensional classification vector.

Then as described in [1], we first splitted the face training data into 5 subsets for training the first layer SVMs. As the purpose of this paper is to show the efficency of the combination rules we simply extracted the subsets by random sampling on the original dataset. We also tested splitting the data by clustering, but then we noticed that each cluster represented a specific scenario in the databases. Thus the training focused more on the scenario properties than the face structure which resulted in a biased combination. In the following, we only consider the case of random sampling for generating the subsets.

On each of these subsets (700 faces and 2000 non faces) we trained a SVM by cross-validation using Radial Basis Functions as kernels. Table 1 reports the performances of each of the SVMs on the test set.

The results show the benefits of splitting the data: not only we reduced the complexity of the training stage, but we also obtain sparser models (models with less support vectors). It is important to remark that sparser models produce better generalization capabilities.

Then we compare the performances of the combination rules in Table 2.

We notice that the "max" and "min" rules are equal because we work in a binary classification task. On the other hand the "median" and the "majority" vote return the same labels because we consider an odd number of SVMs in the first layer such that the median classifier always belongs to the majority vote.

Classifier	Test(%)	Face(%)	NonFace(%)	# SV
SVM1	95.18	90.47	97.92	223
SVM2	95.96	90.57	97.10	325
SVM3	95.35	88.95	96.97	298
SVM4	95.92	91.70	95.30	255
SVM5	95.40	92.07	97.42	274
Single SVM	94.17	91.40	96.95	1068

Table 1: Total error rate, true positive rate and true negative rate of the first layer SVMs on the test set made of 4295 faces and 10000 non faces. Last column represents the number of support vectors selected for each SVM.

Rule	Test(%)	Faces (%)	Non Faces (%)
Max	95.17	92.20	98.15
Min	95.17	92.20	98.15
Median	95.27	92.57	97.98
Majority vote	95.27	92.57	97.98
Product	95.16	92.17	98.15
Sum	95.22	92.32	98.12

Table 2: Combination of the first layer SVMs.

The probability combination rules improve the classification power of the first layer SVMs and moreover they improve the classification rates of the faces which is something important in the context of face detection.

4. CONCLUSIONS

In this paper we presented a method for face class modeling using Combined Probabilistic SVMs. We propose a tools for learning a classifier that performs particularly well on large datasets, as it is needed for face detection. The decomposition of the training set into several parallel subsets yields parallel classifiers of lower complexity. Then combining the posterior probability estimations gives also better generalization results than the single SVM trained on the complete original set. However we noticed in this work that the technique used for splitting the initial dataset was a key decision that directly affects the classification rates. Thus depending on the training data, the random sampling or clustering should be used for generating the partitions. We are currently working on finding the best sampling technique according to a particular dataset, or for example using metrics more appropriated than the euclidian one for the clustering. Finally a more practical step will be to implement this classifier into a complete face detection system in order to test the accuracy in real world conditions.

5. ACKNOWLEDGMENTS

This work is supported by the Swiss National Science Foundation through the National Center of Competence in Research on "Interactive Multimodal Information Management (IM2)".

REFERENCES

[1] J. Meynet, V. Popovici, and JP. Thiran, "Face class modeling using mixture of svms," in *In Proceedings of International Conference on Image Analysis and recognition, ICIAR 2004, Porto, Portugal*, J. Bigun, Ed., Berlin, September 2004, Springer-Verlag.

[2] Paul Viola and Michael Jones, "Robust real-time object detection," *International Journal of Computer Vision - to appear*, 2002.

[3] H.A. Rowley, S.Baluja, and T.Kanade, "Human face detection in visual scenes," in *Advances in Neural Information Processing Systems*, David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, Eds. 1996, vol. 8, pp. 875–881, The MIT Press.

[4] Kah-Kay Sung and Tomaso Poggio, "Example-based learning for view-based human face detection," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, 1998.

[5] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines:an application to face detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1997 pp. 130–136.

[6] Popovici Vand Thiran J, "Face detection using svm trained in eigenfaces space," in *4th International Conf. AVBPA, Guildford, UK*, Berlin, June 2003, vol. 2688 of *Lecture Notes in Computer Science*, pp. 190–198, Springer-Verlag.

[7] R. Collobert, S. Bengio, and Y. Bengio, "A parallel mixture of svms for very large scale problems," in *Advances in Neural Information Processing Systems*. 2002, MIT Press.

[8] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. vol. 20, no. 3, pp. 226–239, 1998.

[9] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America*, vol. 4, pp. 519–524, 1987.

[10] Matthew Turk and Alex Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[11] Vladimir Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, 1995.

[12] N.Cristianini and J.Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, 2000.

[13] J. Platt, "Probabilistic outputs for support vector machines and comparison to regularized likelihood methods," in *Advances in Large Margin Classiers, eds., MIT Press*, A.J. Smola and al., Eds., 2000, pp. 61–74.

[14] Bailly-Bailliere E. and al., "The banca database and evaluation protocol," in *4th International Conference on AVBPA, Guildford, UK*, Berlin, June 2003, vol. 2688, pp. 625–638. http://banca.ee.surrey.ac.uk

[15] K Messer and al., "XM2VTSDB: The Extended M2VTS Database," in *In Proc. Int. Conf AVBPA*, 1999, pp. 72–77.

[16] R.W.Frischholz and al., "BioID: A multimodal biometric identification system," *j-COMPUTER*, vol. 33, no. 2, pp. 64–68, 2000.

[17] P.J. Phillips and al., "The FERET database and evaluation procedure for face-recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, 1998.