COMBINING SUPPORT VECTOR MACHINES FOR ACCURATE FACE DETECTION

I. Buciu[†] *C. Kotropoulos*[⋆] *and* *I. Pitas*[⋆]

[⋆] Department of Informatics, Aristotle University of Thessaloniki
Box 451, Thessaloniki 540 06, Greece
costas@zeus.csd.auth.gr
[†] Applied Electronics Department, University of Oradea
5 Armatei Romane str., Oradea, 3700, Romania

ABSTRACT

The paper proposes the application of majority voting on the output of several support vector machines in order to select the most suitable learning machine for frontal face detection. The first experimental results indicate a significant reduction of the rate of false positive patterns.

1. INTRODUCTION

Face detection is a prerequisite task in many applications including face recognition, teleconferencing, and face gesture recognition. The human face plays a central role in intelligent human computer interaction. The goal of face detection is to determine if there are any human faces in a test image or not. If a face exists, the objective is then to locate it in the test image regardless of the actual position, orientation, scale and pose of the head as well as the lighting variations. Due to the many above mentioned variable factors, developing a robust human face detector is a hard task.

Many approaches have been proposed for face detection. For instance, Yang and Huang have developed a system that attempts to detect a facial region at a coarse resolution and subsequently to validate the outcome by detecting facial features at the next resolution by employing a hierarchical knowledge-based pattern recognition system [1]. A probabilistic method to detect human faces using a mixture of factor analyzers has been proposed in [2]. Other techniques include neural networks [3], or algorithms where feature points are detected using spatial filters and then grouped into face candidates using geometric and gray level constrains [4]. Sung and Poggio report an example based-learning approach [5]. They model the distribution of human face patterns by means of few view-based face and non-face prototype clusters. A small window is moved over all portions on an image and determines whether a face exists in each window based on distance metrics. The application of Support Vector Machines (SVMs) in frontal face detection was first proposed in [6].

In several papers, false positive patterns are collected and are fed to the learning machine at the next iteration of the training procedure, a procedure that resembles bootstrap [5]. An alternative approach is proposed in this paper. More specifically, we propose to rank an ensemble of SVMs trained on the same training set by combining their outputs with majority voting in the decision making process. By doing so, we can define the most efficient SVM, i.e., that whose outputs appear most frequently in the set of the outputs produced by the ensemble of SVMs. We apply this technique to frontal face detection and report a significant reduction of rate of false positive patterns. Also, we apply bagging technique to each SVM and compare the results with the approach described in the paper.

The outline of the paper is as follows. A brief description of SVMs is presented in Section 2. The proposed method is described in Section 3 followed by a briefly presentation of bagging approach in Section 4. Experimental results are reported in Section 5 and conclusions are drawn in Section 6.

2. SUPPORT VECTOR MACHINES

SVMs is a state-of-the-art pattern recognition technique whose foundations stem from statistical learning theory [7]. However, the scope of SVMs is beyond pattern recognition, because they can handle also another two learning problems, i.e., regression estimation and density estimation. In the context of pattern recognition, the main objective is to find the optimal separating hyperplane, that is, the hyperplane that separates the positive and negative examples with maximal margin. SVM is a general algorithm based on guaranteed risk bounds of statistical learning theory, i.e., the so-called *structural risk minimization* principle. This principle is based on the fact that the error rate of learning machine on test data (i.e., the generalization error rate) is bounded by the sum of the training error rate and a term that depends on the Vapnik-Chervonenkis (VC) dimension [7]. We briefly describe linearly separable case followed by linearly non-separable case and the nonlinear one.

Consider the training data set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ of labeled training patterns, where $\mathbf{x}_i \in I\!R^d$ with d denoting the dimensionality of the training patterns, and $y_i \in \{-1, +1\}$. We claim that $\mathcal{S}$ is linearly separable if for some $\mathbf{w} \in I\!R^d$ and $b \in I\!R$,

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \qquad \text{for} \quad i = 1, 2, \ldots, l \tag{1}$$

where $\mathbf{w}$ is the normal vector to the separating hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$ and b is a bias (or offset) term [8]. The optimal separating hyperplane is the solution of the following quadratic problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ \text{subject to} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \ldots, l \end{aligned} \tag{2}$$

In Figure 1 the optimal separating hyperplane is drawn in the case

This work was supported by the European Union Research Training Network “Multi-modal Human-Computer Interaction (HPRN-CT-2000-00111).

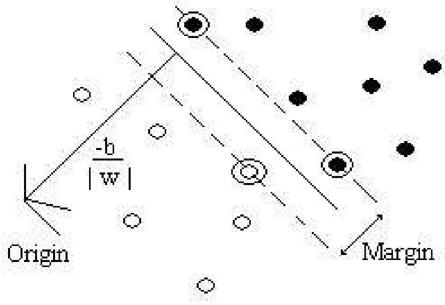


Fig. 1. Optimal separating hyperplane in the case of linearly separable data. Support vectors are circled.

of linearly separable data. The optimal $\mathbf{w}^*$ is given by

$$\mathbf{w}^* = \sum_{i=1}^{l} \lambda_i^* y_i \mathbf{x}_i \tag{3}$$

where $\boldsymbol{\lambda}^*$ is the vector of Lagrance multipliers obtained as the solution of the so-called Wolfe-dual problem

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{l} \lambda_i - \boldsymbol{\lambda}^T \mathbf{D} \boldsymbol{\lambda} \\ \text{subject to} \quad & \sum_{i=1}^{l} y_i \lambda_i = 0 \\ & \lambda_i \geq 0 \end{aligned} \tag{4}$$

where $\mathbf{D}$ is an $l \times l$ matrix having elements $D_{ij} = y_i y_i \mathbf{x}_i^T \mathbf{x}_j$.

Thus $\mathbf{w}^*$ is a linear combination of the training patterns $\mathbf{x}_i$ for which $\lambda_i^* > 0$. These training patterns are called *support vectors*. Given a pair of support vectors $(\mathbf{x}^*(1), \mathbf{x}^*(-1))$ that belong to the positive and negative patterns, the bias term is found by [7]

$$b^* = \frac{1}{2} \left[\mathbf{w}^{*^T} \mathbf{x}^*(1) + \mathbf{w}^{*^T} \mathbf{x}^*(-1) \right]. \tag{5}$$

The decision rule implemented by the SVM is simply

$$f(\mathbf{x}) = \text{sign} \left(\mathbf{w}^{*^T} \mathbf{x} - b^* \right). \tag{6}$$

If the training set $\mathcal{S}$ is not linearly separable, the optimization problem (4) is generalized to

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^{l} \xi_i \\ \text{subject to} \quad & y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, l \\ & \xi_i \geq 0 \end{aligned} \tag{7}$$

where ξ_i are positive slack variables [8], and C is a parameter which penalizes the errors. The situation is summarized schematically in Fig 2. The Lagrange multipliers now satisfy the inequalities

$$0 \leq \lambda_i \leq C. \tag{8}$$

The main difference is that support vectors do not necessarily lie on the margin.

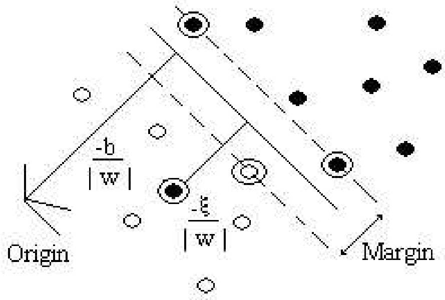


Fig. 2. Separating hyperplane for non-separable data. Support vectors are circled.

Finally, SVMs can also provide nonlinear separating surfaces by projecting the data to a high dimensional feature space $\mathcal{H}$ in which a linear hyperplane is searched for separating all the projected data, $\phi : \mathbb{R}^d \longrightarrow \mathcal{H}$. If the inner product in space $\mathcal{H}$ had an equivalent kernel in the input space $\mathbb{R}^d$, i.e.:

$$\phi^T(\mathbf{x}_i) \phi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j) \tag{9}$$

the inner product would not need to be evaluated in the feature space, thus avoiding the curse of dimensionality problem. In such a case, $D_{ij} = y_i y_i K(\mathbf{x}_i, \mathbf{x}_j)$ and the decision rule implemented by the nonlinear SVM is given by

$$f(\mathbf{x}) = \text{sign} \left(\sum_{\substack{i=1 \\ \lambda_i^* \neq 0}}^{l} \lambda_i^* \, y_i \, K(\mathbf{x}, \mathbf{x}_i) - b^* \right). \tag{10}$$

3. APPLICATION OF MAJORITY VOTING IN THE OUTPUT OF SEVERAL SVMS

Let us consider five different SVMs defined by the kernels indicated in Table 1. The following kernels have been used: (1) Polynomial with q equal to 2; (2) Gaussian Radial Basis Function (GRBF) with $\sigma = 10$; (3) Sigmoid with κ equal to 0.5 and θ equal to 0.2; (4) Exponential Radial Basis Function having σ equal to 10. The penalty, C, in (7)was set up to 500. In Table 1, $|| \cdot ||_p$ denotes the vector p-norm, $p = 1, 2$. For brevity, we index each SVMs by k, $k = 1, 2, \ldots, 5$. To distinguish between training and test patterns, the latter ones are denoted by $\mathbf{z}_j$. Let $\mathcal{Z}_k$ be the set of test patterns classified as face patterns by the kth SVM during

Table 1. Kernel functions used in SVMs.

k	SVM type	Kernel function $K(\mathbf{x}, \mathbf{y})$
1	Linear	$\mathbf{x}^T \mathbf{y}$
2	Polynomial	$(\mathbf{x}^T \mathbf{y} + 1)^q$
3	GRBF	$\exp(-\frac{\|\|\mathbf{x}-\mathbf{y}\|\|_2^2}{2\sigma^2})$
4	Sigmoid	$\tanh(\kappa \cdot \mathbf{x}^T \mathbf{y} - \theta)$
5	ERBF	$\exp(-\frac{\|\|\mathbf{x}-\mathbf{y}\|\|_1}{2\sigma^2})$

the test phase, i.e.,

$$\mathcal{Z}_k = \{\mathbf{z}_j : f_k(\mathbf{z}_j) = 1\}, \;\; k = 1, 2, \ldots, 5. \tag{11}$$

Let $\mathcal{Z} = \cup_{k=1}^{5} Z_k$. We define the histogram of labels assigned to all $\mathbf{z}_j \in \mathcal{Z}$ as

$$h(\mathbf{z}_j) = \#\{f_k(\mathbf{z}_j) = 1, \;\; k = 1, 2, \ldots, 5\} \tag{12}$$

where $\#$ denotes the set cardinality. We combine the decisions taken separately by the SVMs indexed by $k = 1, 2, \ldots, 5$ as follows:

$$g(\mathbf{z}_i) = \begin{cases} 1 & \text{if} \quad i = \arg\max_j \{h(\mathbf{z}_j)\} \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

Let us define the quantities:

$$\begin{aligned} F_k &= \#\{f_k(\mathbf{z}_j) = 1, \;\; \mathbf{z}_j \in \mathcal{Z}_k\} \\ G_k &= \#\{g(\mathbf{z}_j) = 1, \;\; \mathbf{z}_j \in \mathcal{Z}_k\} \end{aligned} \tag{14}$$

To determine the best SVM, we simply choose

$$m = \arg\max_k \{\frac{G_k}{F_k}\}. \tag{15}$$

4. BAGGING APPROACH

Bagging is a method for improving the prediction error of classifier learning system by generating replicated bootstrap samples of the original training set [9]. Given a training set described in Section 2 a $\mathcal{S}^\star$ bootstrap replicate of it is built by taking l samples with replacement from the original training set $\mathcal{S}$. The learning algorithm is then applied to this new training set. This procedure is applied B times yielding $\mathcal{S}^{\star 1}$,..., $\mathcal{S}^{\star B}$. Finally, those B new models are aggregating by uniform voting and the resulting class is that one having the most votes over the replicas. Notice that in the bootstrap replica an original pattern may not appear on it while others may appear more than once, on average 63% of he original patterns appearing in the bootstrap replica.

5. EXPERIMENTAL RESULTS

For all experiments the Matlab SVM toolbox developed by Steve Gunn was used [10]. For a complete test, several auxiliary routines have been added to the original toolbox.

5.1. Data set and pattern extraction

A training data set of 96 images, 48 images containing a face and another 48 images with non-face patterns, is built. The images containing face patterns have been derived from the face database of IBERMATICA where several sources of degradation are modeled, such as varying face size and position and changes in illumination. All images in this database are recorded in 256 grey levels and they are of dimensions 320 × 240. These face images correspond to 12 different persons. For each person four different frontal images have been collected. The procedure for collecting face patterns is as follows. From each image a bounding rectangle of dimensions 160 × 128 pixels has been manually determined that includes the actual face. The face region included in the bounding rectangle has been subsampled four times. At each subsampling, non-overlapping regions of 2×2 pixels are replaced by their average. Accordingly, training patterns $\mathbf{x}_i$ of dimensions 10×8 are built. The ground truth, that is, the class label $y_i = +1$ has been appended to each pattern. Similarly, 48 non-face patterns have been collected from images depicting trees, wheels, bubbles, and so on, by subsampling four times randomly selected regions of dimensions 160×128. The latter patterns have been labeled by $y_i = -1$.

5.2. Performance assessment

We have trained the five different SVMs indicated in Table 1. The trained SVMs have been applied to six face images from the IBERMATICA database that have not been included in the training set. Each test image corresponds to a different person. The resolution of each test image has been reduced four times yielding a final image of dimensions 15×20. Scanning row by row the reduced resolution image, by a rectangular window 10×8, test patterns are classified as non-face ones (i.e., $f(\mathbf{z}) = -1$) or face patterns (i.e., $f(\mathbf{z}) = 1$). When a face pattern is found by the machine, a rectangle is drawn, locating the face in image.

We have tabulated the ratio G_k/F_k in Table 2. From Table 2, it can be seen that ERBF is found to maximize the ratio in (15) for the five test images. On the contrary the machine built using the sigmoid kernel attains the worst performance with respect to (15). Interestingly, the ERBF machine experimentally yields the greatest number of support vectors, as can be seen in Table 3.

Table 2. Ratio G_k/F_k achieved by the various SVMs.

SVM type	Test Image numbers					
k	1	2	3	4	5	6
1	0.83	0.20	0.57	0.66	**1**	0.74
2	0.52	0.28	0.57	0.44	**1**	0.71
3	0.67	0.25	0.44	0.44	0.80	0.83
4	0.64	0.14	0.15	0.11	0.22	0.13
5	**1**	**0.50**	**0.80**	**0.80**	0.80	**1**

Table 3. Number of support vectors found in the training of the several SVMs studied.

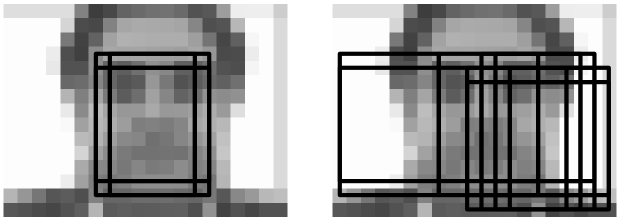
SVM type	Test Image numbers					
k	1	2	3	4	5	6
1	11	11	11	11	10	11
2	14	13	14	14	14	13
3	12	10	12	16	12	12
4	13	11	11	11	11	11
5	39	41	41	40	39	40

To assess the performance of the majority voting procedure, we have manually annotated each test pattern $\mathbf{z}_i$ with the ground truth that is denoted as $z_{i,81}$. Two quantitative measurements have been used for the assessment of the performance of each SVM, namely, the *false acceptance rate* (FAR) (i.e., the rate of false positives) and the *false rejection rate* (FRR) (i.e., the rate of false negatives) during the test phase. We have measured FAR and FRR for each SVM individually as well as after majority voting. We have found that FRR is always zero while FAR varies. For each of the five different SVM we used bagging. The number of bootstrap replicas was 21. Unfortunately, for this set of data, the method did not work well. Moreover, perturbing the distribution of the original data bagging slightly degrades the performance of the initial classifier. The values of FAR attained by each SVM individually and after applying majority voting along with the values obtained with bagging are shown in Table 4. The FAR after bagging are in parentheses. It is seen that application of majority voting reduces the number of false positives in all cases and particularly when $F_k \neq G_k$.

Table 4. False acceptance rate (in $\%$) achieved by the various SVMs individually, with bagging and after applying majority voting. In parentheses are the values corresponding to bagging

SVM type	Test Image numbers					
k	1	2	3	4	5	6
1	3.9 (4.7)	10.5 (12.1)	6.5 (7.6)	5.2 (6.5)	2.6 (3.5)	6.5 (7.8)
2	6.5 (10.1)	6.5 (9.3)	6.5 (7.6)	9.2 (9.2)	2.6 (3.5)	6.5 (10.8)
3	5.2 (7.7)	7.8 (10.1)	9.2 (10.6)	9.2 (13.5)	3.9 (4.5)	5.2 (8.8)
4	7.8 (23.7)	17.1 (29.2)	31.5 (44.6)	44.7 (78.5)	21.0 (46.5)	47.3 (88.8)
5	2.6 (2.6)	2.6 (3.1)	3.9 (6.5)	3.9 (6.5)	3.9 (4.5)	3.9 (4.8)
combining	2.6	1.3	2.6	2.6	2.6	3.9

Figure 3 depicts 2 extreme cases observed during a test. It is seen that majority voting helps to discard many of the candidate face regions returned by a single SVM (Fig. 3(b)) yielding the best face localization (Fig. 3(a)).



(a) (b)

Fig. 3. (a) Best and (b) worst face location determined during a test.

6. CONCLUSIONS AND DISCUSSION

In this paper, we have attempted to improve the accuracy of SVMs by applying majority voting on the output of an ensemble of different machines. We have tested the aforementioned technique for frontal face detection. We also used bagging in the trial to reduce the misclassification error and compared the results with the method proposed in this paper. Note that the majority vote is related to each SVM in the case of bagging while it is applied to different SVM's in the case of the ensemble of SVM's. Ensembling different kernel machines turns out to be a better idea than using bagging for achieving more accurate classifier.

7. REFERENCES

[1] G. Yang and T.-S. Yang, “Human face detection in complex backround,” *Pattern Recognition*, vol. 27, no. 1, pp. 53 – 63, 1994.

[2] M.-H.Yang, N. Ahuja, and D. Kriegman “Face detection using a mixture of factor analyzers,” in *Proc. of the 1999 IEEE Int. Conf. on Image Processing*, vol. 3, pp. 612–616, 1999.

[3] R. Vaillant, C. Monrocq, and Y. Len Cun, “Original approach for the localisation of objects in images ,” *IEE Proc. Vis. Image Signal Processing*, vol. 141, no. 4, August 1994.

[4] K.-C Yow and R. Cipolla, “Feature-based human face detection ,” *Image and Vision Computing*, vol. 15, no. 9, pp. 713–735, 1999.

[5] K.-K. Sung, and T. Poggio, “Example-based learning for view-based human face detection,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, January 1998.

[6] E. Osuna, R. Freund, and F. Girosi, “Training support vector machines: An application to face detection,” in *Proc. of the IEEE Computer Society Computer Vision and Pattern Recognition Conf.*, pp. 130–136, 1997.

[7] V.N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer Verlag, 1995.

[8] C. Burges, “A Tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, pp. 1–43, 1998.

[9] L. Breiman, “Bagging Predictors,” *Machine Learning*, vol. 24 , pp. 123–140, 1996.

[10] S. Gunn, ”Support Vector Machines for Classification and Regression”, ISIS Technical Report ISIS-1-98, Image Speech & Intelligent Systems Research Group, University of Southapton, May. 1998.