Face Detection Using Adaboosted RVM-based Component Classifier

Ali Reza Bayesteh Tashk, Abolghassem Sayadiyan, SeyyedMajid Valiollahzadeh
Electrical Engineering Department, Amirkabir University of Technology,
15914 Tehran, Iran
Bayesteh_ar@yahoo.com,eea35@aut.ac.ir,valiollahzadeh@yahoo.com

Abstract

In this paper, a new Adaboosted Kernel Classifier algorithm is introduced for face detection application. However, most of the methods used to implement Relevance Vector Machine (RVM), need lengthy computation time when faced with a large and complicated dataset. A new pruning method is used to reduce the computational cost.

The kernel classifier parameters are adaptively chosen. In addition, using Fisher's criterion, a subset of Haar-like features is selected. As a result, our proposed algorithm with its previous counterparts i.e. Support Vector Machine (SVM) and RVM without boosting is compared, which results in a better performance in terms of generalization, sparsity and real-time behavior for CBCL face database.

1. Introduction

Nonlinear classification of data is always of special attention. Face Detection is a problem dealing with such data, due to large amount of variation and complexity brought about by changes in facial appearance, lighting and expression. Feature selection is needed beside appropriate classifier design to solve this problem, like many other pattern recognition tasks.

Tipping's Relevance Vector Machines (RVM) [1] [3] are a Bayesian approach leading to a probabilistic non-linear model with a prior on the weights that promotes sparse solutions. The advantage of RVM over non-Bayesian kernel methods comes from explicit probabilistic formulation that yields predictive distributions for test instances and allows Bayesian model selection [4].

One of the major developments in machine learning in the past decade is the Ensemble method, which finds a highly accurate classifier by combining many moderately accurate component classifiers. Boosting [15] and Bagging [16] are the most common techniques, used to construct Ensemble classifiers. In Comparison with Bagging, Boosting outperforms when the data do not have much noise [17] [18]. Among popular Boosting methods, AdaBoost [6] establishes a collection of weak component classifiers by maintaining a set of weights over training samples and adjusting them adaptively after each Boosting iteration the weights of the misclassified samples by current component classifier will be increased while the weights of the correctly classified samples will be decreased. To implement the weight updates in Adaboost, several algorithms have been proposed [19]. The success of Adaboost can be attributed to its ability to enlarge the margin [5], which could enhance Adaboost's generalization capability. All these factors have to be carefully tuned in practical use of Adaboost. Furthermore, diversity is known to be an important factor which affects the generalization accuracy of Ensemble classifiers [21][19]. In order to quantify the diversity, some methods are proposed [19] [22]. It is also known that in Adaboost there exists an accuracy/diversity dilemma [9], which means that the more accurate two component classifiers become, the less they can disagree with each other. Only when the accuracy and diversity are well balanced, the Adaboost can demonstrate excellent generalization performance. However, the existing Adaboost algorithms do not yet explicitly taken sufficient measurement to deal with this problem.

In this paper we propose a new kernel classifier for face detection. Applying boosted RVM has an advantage of getting accuracy and being Sparse. Boosting will reduce the sparsity in nature, while RVM will compensate this fact. Obtaining accuracy and sparsity will allow the system operate very fast. The rest of the paper is organized as follows Sections 2 describes the feature selection method. In Section 3 we introduce RVM and Adaboost, and then we develop AdaboostRVM. In Section 4, we apply the proposed method for face detection. Finally, we provide discussions and conclusions in Section 5.

2. Feature selection

To find out which features to be used for a particular problem, is referred as feature selection. In this paper, like Viola and Jones [10], we use four types of Haar-like basis functions for feature selection which have been used by Papageorgiou et al [9].

Like their work, we use four types of haar-like feature to build the feature pool. The features can be computed efficiently within integral image. The main objective to use these features is that they can be rescaled easily which avoids to calculate a pyramid of images and yields to fast operation of the system on these features. These features can be seen in Figure 1. Given that the base resolution of the detector is 19x19, the exhaustive set of rectangle features is quite large. Note that unlike the Haar basis, the set of rectangle features is over complete. Like viola, we use image variance σ to correct lighting, which can be got using integral images of both original image and image squared.

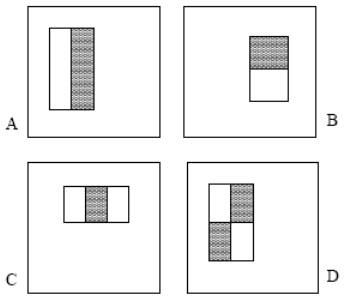

Figure 1. Example rectangle features shown relative to the enclosing detection window. The sum of the pixels which lie within the white rectangles is subtracted from the sum of pixels in the grey rectangles. Two-rectangle features are shown in (A) and (B). Figure (C) shows a three-rectangle feature, and (D) a four-rectangle feature.

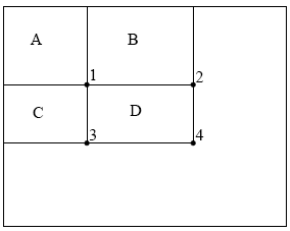

Figure 2.The sum of the pixels within rectangle D can be computed with four array references. The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 correspond to the area A+B and so on

Using the integral image any rectangular sum can be computed in four array references (see Figure 2). Clearly the difference between two rectangular sums can be computed using eight references. Since the two-rectangle features defined above involve adjacent rectangular sums they can be computed in six array references, eight in the case of the three-rectangle features, and nine for four-rectangle features.

we use Fisher's score for between-class measurement as:

$$S_i = \left| \frac{m_{i,face} - m_{i,nonface}}{\sigma^2_{i,face} + \sigma^2_{i,nonface}} \right| \qquad (1)$$

By selecting the feature with highest Fisher's scores and smallest spatial correlation, we can retain the most discriminative feature between face and non-face classes

3. Statistical Learning

In this section, we describe boost based learning methods to construct face/nonface classifier, and propose a new boosting algorithm which improves boosting learning.

3.1. AdaBoost Learning

Given a set of training samples, AdaBoost [7] maintains a probability distribution, W, over these samples. This distribution is initially uniform. Then, AdaBoost algorithm calls a WeakLearn algorithm repeatedly in a series of cycles. At cycle T, AdaBoost provides training samples with a distribution w^t to the WeakLearn algorithm.

AdaBoost, constructs a composite classifier by sequentially training classifiers while putting more and more emphasis on certain patterns.

For two class problems, we are given a set of N labeled training examples $(y_1, x_1), ..., (y_N, x_N)$, where $y_i \in \{+1, -1\}$ is the class label associated with example x_i .

For face detection, x_i is an image sub-window of a fixed size containing an instance of the face $(y_i = +1)$ or non-face $(y_i = -1)$ pattern. In the notion of AdaBoost see table 1, a stronger classifier is a linear combination of M weak classifiers.

In boosting learning [15], each example x_i is associated with a weight w_i, and the weights are updated dynamically using a multiplicative rule according to the errors in previous learning so that more emphasis is placed on those examples which are erroneously classified by the weak classifiers learned previously.

Greater weights are given to weak learners having lower errors. The important theoretical property of AdaBoost is that if the weak learners consistently have accuracy only slightly better than half, then the error of the final hypothesis drops to zero exponentially fast. This means that the weak learners need be only slightly better than random.

Furthermore, since proposed AdaBoost with RVM invents a convenient way to control the classification accuracy of each weak learner, it also provides an opportunity to deal with the well-known accuracy/diversity dilemma in Boosting methods. This is a happy accident from the investigation of AdaBoost based on RVM weak learners.

Table 1. The AdaBoost with RVM Algorithm [3] .

1. Input: Training sample
Input: a set of training samples with labels $(y_1, x_1), ..., (y_N, x_N)$, ComponentLearn algorithm, the number of cycles *T*.

2. Initialize: the weights of training samples: $w_i^1 = 1/N$, for all $i = 1, ..., N$

3. Do for $t = 1, ..., T$
(1)Use ComponentLearn algorithm to train the component classifier *ht* on the weighted training sample set.
(2)Calculate the training error of h_t :

$$\varepsilon_t = \sum_{1=1}^{N} w_i^t, y_i \neq h_t(x_i) \quad (2)$$

(3)Set weight of component classifier h_t :

$$h_t : \alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \quad (3)$$

(4)Update the weights of training samples:

$$w_i^{t+1} = \frac{w_i^t \exp\{\alpha_t y_i h_t(x_i)\}}{C_t} \quad (4)$$

where C_t is a normalization constant, and

$$\sum_{i=1}^{N} w_i^{t+1} = 1 \quad (5)$$

4. Output: $f(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$.

3.2. RVM for classification

$$y(X;W) = \sum_{i=1}^{N} w_i K(X,\mathrm{X}_i) + w_0 \quad (6)$$

Where $K(X,\mathrm{X}_i)$ is a kernel function, effectively defining one basis function for each example in the training set.

Relevance vector machine (RVM) is a Bayesian framework for achieving the sparse linear model (6). In sparse model, the majority of the W s are zero. The sparsity of model is based on a hierarchical prior, where an independent Gaussian prior is defined on the weight parameters in the first level:

$$p(W|\alpha) = \prod_{i=1}^{N} N\left(w_i|0,\alpha_i^{-1}\right) \quad (7)$$

Where $\alpha = (\alpha_1,\alpha_2,...,\alpha_N)$ is a vector consisting of N hyper parameters. An independent Gamma hyper prior is used for the variance parameters in the second level:

$$p(\alpha_i) = Gamma(a,b) \quad (8)$$

Where a and b are constants. The key point of this method is using the maximum a posteriori (MAP) instead of maximum likelihood (ML) for the Weight estimation.

Given the N pairs of training data $\{\mathrm{X}_l,t_l\}\Big|_{l=1}^{N}$, the dataset likelihood is defined by applying the logistic sigmoid link function $\sigma(y) = 1/1+e^{-y}$ to $y(X)$ and adopting the Bernoulli distribution for $P(t|X)$:

$$P(t|W) = \prod_{n=1}^{N} \sigma\{y(X_n;W)\}^{t_n}\left[1-\sigma\{y(X_n;W)\}\right]^{1-t_n} \quad (9)$$

Where class label is denoted by $t_l \in \{0,1\}$. The parameters w_i are then obtained by maximizing the posterior distribution of the class labels given the input vectors with respect to prior information. For this maximization, a numerical method is suggested as follows:

1. For the current, fixed, values of α, the most probable' weights W_{MP} are found, giving the location of the mode of the posterior distribution.

Since $P(W|t,\alpha) \propto P(t|W)P(W|\alpha)$ this is equivalent to finding the maximum, over W, of:

$$\log\left\{P(t|W)P(W|\alpha)\right\} = \sum_{n=1}^{N} t_n \log y_n + (1-t_n)\log(1-y_n) - \frac{1}{2}W^T AW \quad (10)$$

With $y_n = \sigma\{y(X_n;W)\}$

2. Laplace's method is simply a quadratic approximation to the log-posterior around its mode. The quantity (10) is differentiated twice to give:

$$\nabla_W \nabla_W \log P(W|t,\alpha)\Big|_{W_{MP}} = -\left(\Phi^T B\Phi + A\right) \quad (11)$$

Where

$B = diag(\beta_1,\beta_2,...,\beta_N)$ $\beta_n = \sigma\{y(X_n)\}\left[1-\sigma\{y(X_n)\}\right]$

The posterior is approximated around W_{MP} by a Gaussian approximation with Covariance:

$$\Sigma = \left(\Phi^T B\Phi + A\right)^{-1} \quad (12)$$

And mean

$$\mu = \Sigma\Phi^T Bt \quad (13)$$

3. Using the statistics Σ and μ of the Gaussian approximation, the hyper parameters α are updated as follows:

$$\alpha_i^{new} = \frac{\gamma_i}{\mu_i^2} \quad (14)$$

where μ_i is the i-th posterior mean weight from(14) and $\gamma_i \equiv 1 - \alpha_i^{old} N_{ii}$ which N_{ii} is the i-th diagonal element of Σ .Since computing the μ and Σ based on above mentioned steps takes so much time, we use incremental DFT-RVM for simplicity on implementation.

3.3. Data Pruning

When we are faced to a large and complicated dataset, the accuracy of RVM classification is not as high as expected and the computation time increases rapidly. Therefore, improving the efficiency of RVM is one important area of study.

Now, we present a simple statistical algorithm to identify the most crucial points of the training data. The basic idea is to model the face class as a multivariate normal distribution, which is especially reasonable if one, models only the upright frontal faces that are properly aligned to one another. Note that the training

face images are all upright, frontal, and properly aligned. Therefore, the density function of the face class is modeled as a multivariate normal distribution as follows:

$$p\left(Y\left|w_f\right.\right) = \frac{1}{\left(2\pi\right)^{N/2}\left|\Sigma\right|^{1/2}} \times \exp\left\{-\frac{1}{2}\left(Y-M\right)^t \Sigma^{-1}\left(Y-M\right)\right\} \quad (15)$$

Where $Y \in \mathbb{R}^N$ the discriminating is feature vector and $M \in \mathbb{R}^N, \sum \in \mathbb{R}^{N\times N}$ are the mean vector and the covariance matrix of the face class w_f, respectively.

Afterwards, we model non-face class PDF with a Gaussian mixture model.

$$p\left(Y\left|w_n\right.\right) = \sum_{i=1}^{M} W_i N\left(Y; M_i, \Sigma_i\right) \quad (16)$$

As a result, the crucial data are introduced as follows:

$$\varepsilon_1 \le Log\left(\frac{p(Y|w_n)}{p(Y|w_f)}\right) \le \varepsilon_2 \quad (17)$$

Where the remaining points obtained above, are the ones hardly separable.

The data obtained according to aforementioned scheme, can now be applied to a learning machine

3.4. Adaboosted RVM-Based Classifier

We combine RVM with Adaboost to improve its capability in classification. A polynomial RVM with kernel $K\left(X, X_l\right) = \left(1 + s\, X \cdot X_l\right)^d$ is used in our experiments [2].

RVM weak classifiers are obtained by selecting the polynomial parameters, s and d, then these weak classifiers (classifier error place in range of %55 to %65) are used for optimizing strong classifiers (Adaboost classifier).

3.5. Face Detection System

We explain our face detection system and show how to construct an Adaboosted RVM-based component classifier for face detection. The learning of a detector is done as follows:

1. A set of simple Haar wavelet features are used as candidate features. There are tens of thousands of such features for a 19x19 window.

2. A subset of them based on fisher's score are selected and the corresponding weak classifiers are constructed, using Adaboosted RVM-based component classifier learning. Data pruning is applied to reduce the number of effective samples but it helps to get higher training speed without losing the accuracy in general.

3. A strong classifier is constructed as a linear combination of the weak ones.

4. Experimental results

We adopt a face image database from the Center for Biological and Computational Learning at Massachusetts Institute of Technology (MIT), which contains 2429 face training samples, 472 face testing samples, and 23,573 non-face testing samples. We randomly collected 15,000 non-face training samples from the images that do not contain faces.

We compared RVM and SVM with the same input vectors and 2nd polynomial kernel without boosting. In this stage we generated the input vector by applying a mask on images in our database.

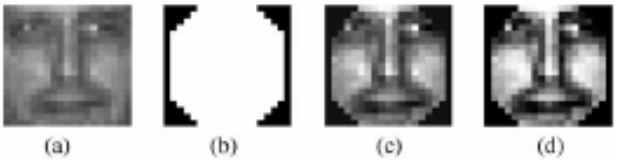

Figure 3.(a) Original face image, (b) The mask, (c) Normalized image and (d) Histogram equalized image

Next we performed normalization and the histogram equalization on resulted image. Figure 3 shows these steps [2]. Then we used the face training samples to calculate 50 Principal Analysis Component (PCA) features.

In the other experiment we calculated 50 Fisher's features and used them as the features of the 2nd polynomial kernel RVM and SVM classifier without boosting.

As we can see in the Figure 4, 50 PCA features outperforms in the terms of accuracy than 50 Fisher's features. This experiment showed RVM is better than SVM classifier.

Our experiment showed that the sparseness of RVM is more than SVM classifier and in testing phase it makes the RVM work fast. Table 2. Compares the sparseness of this approach. Another reason that this method works fast is the advantageous usage of Fisher's feature instead of PCA features. The number of multiplications required for computing Fisher's features are very less than PCA features. Also Figure4 shows that AdaboostRVM by applying pruning performs nearly to AdaboostRVM in accuracy but it reduces the number of samples greatly. Our methods used the highest 50 Fisher's scores features. Figure 4 shows the ROC graph of our method. According to this Figure, it is clear that the performance of the proposed method is much better than the SVM and RVM without boosting.

5. Conclusions

An Adaboosted method is proposed in this paper in order to combine a group of week RVMs which adaptively adjusts the kernel parameters of RVM classifier to get the best result. Experimental results on CBCL database for Face Detection demonstrated that

the proposed AdaboostRVM algorithm performs better than other approaches such as SVM and RVM without being Adaboosted in accuracy and speed.

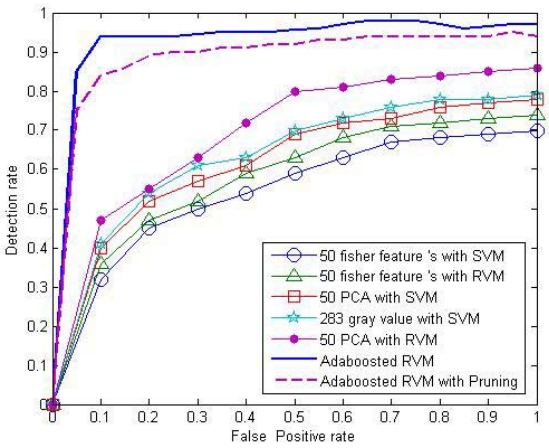

Figure 4. RVM and SVM Comparison

Table 2. Comparison of the sparseness

	SVM	RVM	Adaboosted -RVM	Adaboosted RVM with Pruning
283 gray level	792	--	--	--
50 PCA	766	185	--	--
50Fisher 's feature	529	107	586	427

Experimental results show that AdaboostRVM with pruning, results in a better performance in terms of computational cost and sparsity. Due to this fact that by applying pruning, number of effective samples will be reduced without losing the accuracy noticeably.
Besides these, it is found that proposed AdaboostRVM algorithm demonstrated a better performance on imbalanced classification problems. Based on the AdaboostRVM, an improved version is further developed to deal with the accuracy/diversity dilemma in Boosting algorithms, in raising a better generalization performance. Experimental results indicate that the performance of the Adaboost classifier with RVM is overlay superior to those obtained by the SVM and RVM.

6. References

[1] M.E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine", J. Machine Learning Research, vol. 1, 2001, pp. 211-244.

[2] Frank Y., Shouxian Cheng, Gravano, "Improved feature reduction in input and feature spaces", Pattern Recognition 38, 2005, 651-659

[3] Tipping M. E., Faul A., "Fast Marginal Likelihood Maximization for Sparse Bayesian Models", Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, Jan 3-6, 2003.

[4] Catarina Silva , Bernardete Ribeiro., " Two-level hierarchical hybrid SVM-RVM classification model", Proceedings of the 5th International Conference on Machine Learning and Applications, 2006

[5] Schapire, R. E., Freund, Y., "Boosting the margin: a new explanation for the effectiveness of voting methods", The Annals of Statistics, 26(5), pp.1651–1686, October 1998.

[6] Freund, Y., Schapire, R., Aug 1997 "A decision-theoretic generalization of on-line learning and an application to boosting". Journal of Computer and System Sciences, 55(1):119–139.

[7] Schapire R. E., Y. Singer, "Improved boosting algorithms using confidence-rated predictions,Machine Learning, 37(3), pp.297–336, Dec 1999.

[8] Friedman, J., Hastie, T., R. Tibshirani,. "Additive logistic regression: a statistical view of boosting", Technical report, Department of Statistics, Sequoia Hall, Stanford University, July 1998.

[9] Dietterich, T. G., Aug 2000, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization", Machine Learning, vol. 40, no. 2, Aug 2000, pp. 139–157.

[10] Papageorgiou, C., Oren, M., Poggio, T., "A general Framework for object detection", In International Conference on Computer Vision, 1998.

[11] Viola, P., Jones, M., "Rapid Object Detection Using a Boosted Cascade of Simple Features," Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, Dec. 2001

[12] Rowley, H., Baluja, S., Kanade, T., "Neural network-based face detection", In IEEE Patt. Anal. Mach. Intell., volume 20, pp.22–38,1998.

[13] Li, S. Z., EE, Zhang, Z. Q. ,"FloatBoost Learning and Statistical Face Detection", In IEEE Patt. Anal. Mach. Intell., vol. 26, no. 9, sept. 2004,

[14] Haykin, S., Neural networks: A comprehensive foundation. Prentice Hall, July 1998.

[15] Lienhart, R., Kuranov, A., Pisarevsky, V.,"Empirical analysis of detection cascades of boosted classifiers for rapid object detection", 2003.

[16] schapire. R. E., "The boosting approach to machine learning: An overview", In MSRI Workshop on Nonlinear Estimation and Classification, 2002.

[17] Breiman. L, "Bagging predictors", Machine Learning, 24, pp.123–140, 1996.

[18] Opitz, D., Maclin, R. "Popular ensemble methods: An empirical study", Journal of Artificial Intelligence Research, 11, pp.169–198, 1999.

[19] Bauer, E., Kohavi, R., "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants", Machine Learning, 36(1),pp.105–139 Jul 1999

[20] Kuncheva, L. I., Whitaker, C. J., "Using diversity with three variants of boosting: aggressive, conservative, and inverse", In Proceedings of the Third International Workshop on Multiple Classifier Systems, 2002.

[21] Schwenk , H. and Bengio. Y, "Boosting neural networks", Neural Computation, 12, pp.1869–1887, 2000.

[22] Melville P., Mooney. R. J, "Creating diversity in ensembles using artificial data", Information Fusion, 6(1), pp.99–111, Mar2005.