AdaBoost with Totally Corrective Step for Face Detection

I. M. Anonymous

My Department
My Institute
City, STATE zipcode

M. Y. Coauthor

Coauthor Department
Coauthor Institute
City, STATE zipcode

Abstract

Abstract words...

1. Introduction

Face detection is important for many applications and various algorithms have been proposed [literatura]. In many applications, real-time performance is required. Recently, Viola and Jones [6] introduced an impressive face detection system capable of detecting faces in real-time with high detection rate and very low false positive rate. These properties are attributed especially to (1) very efficient features used, (2) AdaBoost learning algorithm, and (3) a cascade technique used for decision making. In this paper, an improvement of AdaBoost algorithm is proposed and its utility for cascade building is shown.

Viola and Jones' detector consists of several classifiers trained by discrete AdaBoost algorithm [1] and organised into a decision cascade. Each cascade stage classifier is set to reach a very high detection rate and an "acceptably" low false positive rate. Because it is trained on the data classified as a face by the previous stages, the final false positive rate is very low (multiplication of the previous stages' false positive rates, see Algorithm 3) and the final detection rate remains acceptably high. The cascade evaluation is equivalent to a degenerated decision tree. When the current stage classifier labels a region in an image as a non-face, the decision process is terminated. Otherwise, the next stage classifier is run. The region is declared a face if it is accepted by all classifiers in the cascade.

Face detection is done by scanning the cascade detector across the image at multiple scales and locations. A typical image contains only small number of face regions compare to the number of regions scanned. Due to early termination of the decision process in non-face regions, only few stages of the cascade are evaluated in average ([6, 4]). Hence, the speed of the cascaded classifier depends heavily on the computational complexity and rejection rates of the first few stages.

A question discussed in this paper is, whether the classifiers in the first few stages can be further shortened, while keeping the properties allowing reaching the same or better detection rates. A new algorithm based on AdaBoost is developed as an attempt to answer the question.

AdaBoost [1] constructs the classifier as a linear combination of "weak" classifiers chosen from a given, finite or infinite, set. Its goal is to chose small number (compare to the size of the set) of weak classifiers and assign them proper coefficients to represent a decision hyperplane sufficiently. Hence, AdaBoost can be viewed as an optimization procedure, which works in the space of weak classifiers' coefficients, starting with a zero vector and ending with a vector with only small number of non-zero elements.

Standard (discrete) AdaBoost is a greedy algorithm, which changes one zero coefficient to a non-zero value in each step. Because of its greedy character, neither the found weak classifiers nor their coefficients are optimal.

A totally corrective algorithm with coefficients updates (TCAcu) proposed in this paper differs from the standard AdaBoost in two main aspects. Firstly, in the standard AdaBoost, a newly added weak classifier can be shown to be "independent" in precisely defined way on the last added one. TCAcu finds a new weak classifier that is independent on *all* previous ones. Secondly, the coefficients of already found weak classifiers are updated repetitively during the learning process. It is shown, that these modifications minimise the classification error upper bound more greedily and that shorter classifiers are found.

The term "totally corrective algorithm" was first used by Kivinen and Warmuth [2]. They devised a variant of AdaBoost with the same independence property as in TCAcu. However, Kivinen and Warmuth's motivation was independence itself. In order to reach it, they omitted the original AdaBoost training error minimisation function. To keep this working properly, the algorithm has to be designed more carefully. Kivinen and Warmuth also give no empirical analysis of the algorithm.

Another attempt to shorten the final classifier was proposed by Li at al. [3] and was motivated by the feature selection view of AdaBoost. In case, the weak classifiers cor-

respond directly to the features as in Viola and Jones's face detection framework, changing one coefficient to a non-zero value effectively selects this feature [6]. Li at al. proposed a modification of AdaBoost, FloatBoost, where some of already non-zero coefficients are set back to zero when it leads to a lower upper bound on the classification error. Instead of the greedy feature selection, the sequential floating forward selection (SFFS) technique [literatura, jaka?] is used. Li at al. show that this modification leads to shorter classifiers.

The main contribution of this paper is (1) modification of totally corrective algorithm which leads to shorter classifiers and, (2) introduction of totally corrective algorithm to face detection. We show that resulting classifier gives results comparable to Viola and Jones' and runs faster.

The paper is structured as follows. In the Section 2 the totally corrective algorith with coeffiecients updates is described in the framework of the standard AdaBoost. Then, in Section 3, necessary details of Viola and Jones' work are given. Experimental results are shown in section 4 and the paper is concluded in Section 5.

2. Totally Corrective Algorithm

In this section, standard AdaBoost algorithm is described and motivation for the totally corrective step (TCS) is given. Then TCS is explained and its role in AdaBoost learning is discussed. Finally, another improvement, 0^{th} weak classifier, implied by different approach, is proposed to further enhance learning.

2.1. Standard AdaBoost

Totally corrective algorithm (TCA) is based on AdaBoost [1] and its basic scheme is depicted in Algorithm 1. Schapire and Singer's [5] notation is used and the algorithm differs from Schapire and Singer's one only in extra step 5.

The goal of AdaBoost is to train a classifier based on training set examples. First, AdaBoost takes a labeled training set and assigns a weight $D_1(i)$ to each training sample. Learning is done in a loop. In step t, a weak classifier h_t is selected, with the smallest weighted error on the training set. The loop is terminated if this error exceeds $1/2$. Then, a coefficient α_t is set and the weights are updated using the exponential update rule. Here, Z_t is normalization factor which assures D_{t+1} remains a distribution. The final decision rule is a linear combination of the selected weak classifiers weighted by their coefficients. The sign operation gives the class label.

As has been shown in [5], the algorithm minimises an upper bound on the classification error on the training set

$$\varepsilon_{tr}(H) \leq \prod_{t=1}^{T} Z_t = (1/2)^T \prod_{t=1}^{T} \sqrt{\epsilon_t(1-\epsilon_t)} \qquad (1)$$

Given: $(x_1, y_1), ..., (x_m, y_m)$; $x_i \in \mathcal{X}, y_i \in \{-1, 1\}$
Initialize weights $D_1(i) = 1/m$
For $t = 1, ..., T$:

1. Find $h_t = \arg\min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^{m} D_t(i) I[y_i \neq h_j(x_i)]$
2. If $\epsilon_t \geq 1/2$ then stop
3. Set $\alpha_t = \frac{1}{2}\log(\frac{1+\epsilon_t}{\epsilon_t})$
4. Update
$$D_{t+1}(i) = \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$
5. *Totally corrective step*

Output the final classifier:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

Algorithm 1: Totally corrective algorithm

This upper bound is minimised by selecting a weak classifier with the smallest weighted error on the training set and by setting its coefficient as in step 3.

In addition, used reweighting scheme assures successively selected weak classifiers to be maximally independent, since a new weak classifier is trained on a distribution fulfilling

$$\sum_i D_{t+1}(i) u_{t,i} = 0 \quad \text{where } u_{t,i} = h_t(x_i) y_i \qquad (2)$$

The new distribution is therefore uncorrelated with the mistakes made by the lastly added weak classifier ([5]).

2.2. Totally Corrective Step

The successive independence property is very attractive from the feature selection point of view. Viewing AdaBoost as a feature selector, a question arises whether this concept can be further extended to the independence of all found weak classifiers. A D_{t+1} distribution should then fulfill

$$\sum_i D_{t+1}(i) u_{q,i} = 0 \quad \text{for } q = 1, ..., t$$

where $u_{q,i} = h_q(x_i) y_i$. Unfortunately, there is no close-form solution to this system of equations and sometimes there does not even exist an exact solution ([2]). Therefore, an iterative optimization algorithm have to be used to find the best possible solution.

The proposed algorithm uses original successive independence to find the solution. The basic idea behind is

Initialize $\widehat{D}_0 = D_t$
For $j = 1, 2, ...$

1. Let q_j be such that $|\epsilon_{q_j} - 1/2|$ is maximised.
2. If $|\epsilon_{q_j} - 1/2| < \Delta_{min}$ exit the loop.
3. Let $\widehat{\alpha}_j = 1/2 \ln((1 - \epsilon_{q_j})/\epsilon_{q_j})$.
4. Define
$$\widehat{D}_{j+1}(i) = \frac{1}{Z_t} \widehat{D}_j(i) \exp(-\widehat{\alpha}_j u_{q_j,i})$$
5. $\alpha_{q_j} = \alpha_{q_j} + \widehat{\alpha}_j$

Assign $D_{t+1} = \widehat{D}_j$

Algorithm 2: Totally corrective step with coefficients updates

to repetitively "add" already used weak classifiers to make the new distribution uncorrelated with their mistakes. After number of iterations, this converges to a distribution maximally uncorrelated with all used weak classifiers.

Note that h_r is not decorrelated with $D_{r+2}, D_{r+3}, ...$, i.e. $\epsilon^r_{r+2}, \epsilon^r_{r+3}, ...$ differ from 1/2, where ϵ^r_q denotes error of h_r on D_q. In step $t > r$, h_r can be "added" again with α^r_t, if $\epsilon^r_t < 1/2$. It does not enlongate classification, because h_r can be evaluated only once and used with the coefficient $\alpha_r + \alpha^r_t$ in the final sum. Nevertheless, it makes D_{t+1} decorrelated with mistakes of h_r. If $\epsilon^r_t > 1/2$, negative α^r_t can be used. Moreover, upper bound on the classification error is decreased by this "virtual addition" (see Equation 1). Using this simple trick iteratively, D_{t+1} becomes uncorrelated with all used weak classifiers and besides the upper bound is tighten closer to the classification error which causes faster learning convergence.

Proof of convergence...

Algorithm 2 describes TCS more formally. At time t, a distribution D_t is used to initialize the algorithm. In each iteration a weak classifier is selected from already used ones such, that absolute difference of its error and 1/2 is maximised. Standard scheme is used to find a coefficient for selected weak classifier and to find a new distribution. Found $\widehat{\alpha}_j$ is added to already existing coefficient and the loop is repeated. Since the exact solution does not need to exist, the computation is terminated if close enough solution is found. The last computed distribution is used for the step $t + 1$ of AdaBoost learning.

TCA proposed by Kivinen and Warmuth is very similar to ours, but lacks some important properties. The main goal of their algorithm is also to find a distribution D_{t+1} maximally uncorrelated with all selected weak classifiers. However, they have not made connection to the upper bound minimisation and hence initialised TCS with D_0. In this setup, newly computed errors and coefficients cannot be used to update already used coefficients of the weak classifiers.

2.3. 0^{th} weak classifier

TCS can be seen as a perceptron update algorithm, with slightly modified update rule. Changing of the coefficients of the weak classifiers, can be viewed as shifting a decision hyperplane in the multidimensional space given by the weak classifiers. When no more improvement can be achieved, another dimension is added. Original AdaBoost conversely improves decision only by adding more dimensions.

Understanding this, another improvement can be offered. In the final sum in the final classifier, an absolute element is missing. A decision hyperplane must consequently go through the origin. Adding this element into the sum can be done very intuitively in TCA. The absolute element corresponds to "0^{th}" weak classifier which returns always the same class for any input does the work.

3. Face Detection and AdaBoost

Face detection framework described by Viola and Jones [6] is very good testing area for the proposed algorithm. Several classifiers are trained on the progressively more and more difficult training sets. Moreover, the stage classifiers shortening is connected to the speed of evaluation of the overall system. Applying the algorithm can therefore speed up the detection process.

Viola and Jones' detector is

3.1. Cascade building

The cascade building algorithm is described in the Algorithm 3. Input to the algorithm is desired false positive rate, f, detection rate, d, for each cascade stage, and the final false positive rate of the cascade. Each stage is trained until f and d can be reached. Since AdaBoost is neither designed to reach low false positive rates nor high detection rates. Hence, after

3.2. TCA

No difficult modifications needed. Shorter stages in cascade. Speedup.

4. Experiments

This section describes the experiments with TCAcu in face detection domain. The training dataset is described

Input: False positive, f, and detection rate, d, per stage
final false positive, f_{final}
$F_0 = 1$, $D_0 = 1$
Until $F_i > f_{\text{final}}$

1. Train the classifier until, by threshold change, $f_{\text{reached}} < f$ and $d_{\text{reached}} > d$ on validaton set
2. $F_{i+1} = F_i \times f_{\text{reached}}$
3. $D_{i+1} = D_i \times d_{\text{reached}}$
4. Throw away misclassified faces and generate new non-face data from non-face images

Algorithm 3: Algorithm for cascade building

4.1. Training Dataset

The data for the experiments were collected from various sources. Face images are taken from MPEG7 face images repository [literatura]. Images in this dataset are taken under different lightning conditions, with uniform or scattered background, a quality of images varies, people express different expresion. The pose of the people's heads is generally frontal with slight rotation in all directions. Eyes and nose tip are aligned in all images. The dataset contains 3176 images, from which one was removed for its too big distortion.

For learning purpose, the images were randomly rotated up to $\pm 5^\circ$, shifted up to one pixel a the bounding box was scaled by 1 ± 0.05. Two datasets, training and validation, of the same size as original dataset were created by these perturbations.

Non-face images were collected from the web. Variability was taken into account during downloading the images. The dataset contains images of animals, country site, man-made objects, etc.. More than 3000 images were collected and random sub-windows used as non-face examples.

4.2. Training Process

In order to compare standard AdaBoost and TCAcu, two cascaded classifiers were trained. The comparison is difficult because the stages of the cascades are trained on different data. The experiment is therefore driven by false positive and false negative rates only.

4.3. Results

Tests on MIT+CMU,

5. Summary and Conclusions

I kdyby to nevychazelo rychlejsi, tak pokud to bude kratsi, mohlo by to mit aplikace tam, kde je vyhodnoceni jednoho wc casove narocne nebo drahe.

References

[1] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.

[2] J. Kivinen and M. K. Warmuth. Boosting as entropy projection. In *Proc. 12th Annual Conference on Computational Learing Theory*, pages 134–144, ACM, New York, July 1999.

[3] S. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum. Statistical learning of multi-view face detection. In *ECCV*, page IV: 67 ff., 2002.

[4] R. Lienhart, A. Kuranov, and P. Vadim. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *DAGM*, Magdeburg, Germany, September 2003.

[5] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, pages 37(3): 297–336, 1999.

[6] P. Viola and M. Jones. Robust real time object detection. In *SCTV*, pages xx–yy, 2001.

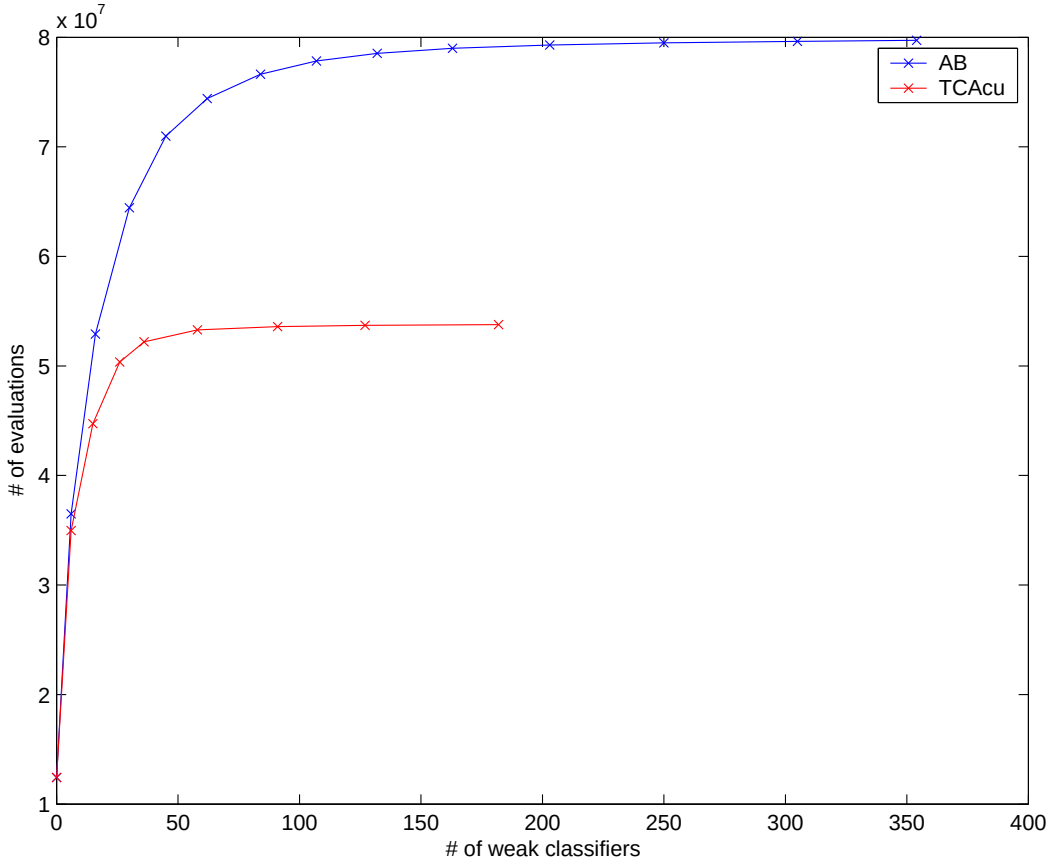

Figure 1: Speed comparison. Vertical axis shows the number of weak classifiers' evaluations made on MIT+CMU dataset given the length of the cascade (horizontal axis). Crosses mark the cascade stages.

	Classifier length		Number of detections		False negatives		False positives		Multiple detections	
Stage	AB	TCAcu	AB	TCAcu	AB	TCAcu	AB	TCAcu	AB	TCAcu
0			12431151	12431151						
1	6	6	4009205	3757632	0	0	3930473	3682307	78257	74850
2	10	9	1643072	1083123	0	0	1598933	1054019	43664	28629
3	14	11	823246	512004	0	1	795262	492881	27509	18649
4	15	10	435483	183962	2	5	415751	173341	19259	10151
5	17	22	201982	49814	4	16	189902	44287	11609	5068
6	22	33	100887	9052	11	39	92226	6488	8197	2129
7	23	36	52867	3173	17	83	46499	1584	5910	1198
8	25	55	27504	1149	26	143	22966	183	4089	622
9	31		14818		35		11262		3116	
10	40		7568		53		5070		2076	
11	47		4194		59		2193		1585	
12	55		2602		73		905		1295	
13	55		1828		84		426		1011	

Figure 2: Comparison of AB and TCAcu