Training Sequential On-line Boosting Classifier for Visual Tracking*

Helmut Grabner[1,3], Jan Šochman[2], Horst Bischof[1], Jiří Matas[2]

[1]*Institute for Computer Graphics and Vision, Graz University of Technology, Austria*
[2]*Center for Machine Perception, Czech Technical University, Prague*
[3]*Computer Vision Laboratory, ETH Zurich, Switzerland*

{hgrabner, bischof}@icg.tugraz.at {sochmj1, matas}@cmp.felk.cvut.cz

Abstract

On-line boosting allows to adapt a trained classifier to changing environmental conditions or to use sequentially available training data. Yet, two important problems in the on-line boosting training remain unsolved: (i) classifier evaluation speed optimization and, (ii) automatic classifier complexity estimation. In this paper we show how the on-line boosting can be combined with Wald's sequential decision theory to solve both of the problems.

The properties of the proposed on-line WaldBoost algorithm are demonstrated on a visual tracking problem. The complexity of the classifier is changing dynamically depending on the difficulty of the problem. On average, a speedup of a factor of 5-10 is achieved compared to the non-sequential on-line boosting.

1 Introduction

On-line boosting [2] proved its usefulness in many practical applications like object tracking [3, 14] and background modeling [4] and is used to improve object detectors over time (e.g. [5, 13]). Yet, there remain two important unsolved problems for practical applications: (i) optimization of the classifier evaluation speed, and (ii) automatic determination of the classifier complexity. Both problems are closely related, i.e., having the least complex classifier that solves the problem leads also to optimal speed. We show, how to solve both of these problems.

To overcome the first point, following the idea of Viola and Jones [11] who proposed a cascaded AdaBoost classifier, other authors tried to improve the evaluation speed of the classifier by e.g. introducing a FloatBoost (weak classifiers can be also removed from the strong classifier) [7], a vector boosting [6] or by using sequential decision making theory [10]. All these methods work in an off-line manner, meaning all the training samples are given in advance and the classifier is kept fixed after being trained. Recently, Wu and Nevatia [13] investigated to use the cascade approach in the on-line boosting framework. Their approach uses many heuristic decisions and is not well founded in the theory.

All current approaches for on-line learning need the number of weak classifier to be given in advance [9, 2]. However, in tasks where the decision problem changes over time, like in object tracking, it is impossible to specify the classifier complexity in advance. A common approach is to train complex classifiers which can handle all situations but this is less effective when the task becomes easier.

In this paper, we introduce Wald sequential decision theory in the on-line framework inspired by the WaldBoost algorithm [10]. Our approach overcomes both problems of classifier speed and complexity optimization in the on-line setting. Experiments on a visual object tracking task show that the method is able to automatically adapt the classifier complexity to changing problem difficulty.

2 Preliminaries

2.1 On-line Boosting for feature selection

The goal of training in both off-line [1] and on-line [9] boosting is to minimize the training error by selecting and combining a set of "weak" classification

*This work has been supported by the Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04 and the Austrian Science Fund (FWF) under the doctoral program Conuence of Vision and Graphics W1209, and by EC project FP6-IST-027113 eTRIMS, and by Czech Science Foundation Project 102/07/1317.

algorithms $\{h_n(x)|h_n(x) : \mathcal{X} \rightarrow \{+1,-1\}\}$ into a strong classifier $H(x)$

$$H(x) = \text{sign}(f_n(x)) \quad \text{where} \quad f_n(x) = \sum_{n=1}^{N} \alpha_n h_n(x) \ . \tag{1}$$

The main differences between off-line and on-line AdaBoost training is the way the data is obtained and how the strong classifier is built. In off-line training all the data is available in advance. The on-line training uses one training sample at a time. To build a classifier in the off-line training one weak classifier is added each training round while, in the on-line training, the strong classifier is initialized at the beginning and is updated by each training sample.

Here we describe the on-line boosting for feature selection proposed by Grabner and Bischof [2]. The main idea is to perform on-line boosting on *selectors* rather than on the weak classifiers directly. Each selector keeps a set of M weak classifiers $\mathcal{H}^n = \{h_1^n(x), \ldots, h_M^n(x)\}$ and the training procedure selects the one with the minimal estimated error to be included into the strong classifier. To estimate the weak classifiers error an importance/difficulty λ of a sample is propagated through the set of N selectors. The selectors are initialized randomly with weak classifiers. When a new training sample (x, y), where $y \in \{-1,+1\}$ is the label, arrives the selectors are updated sequentially. First, the importance weight λ of the sample is initialized to 1. The weak classifier with the smallest estimated error is selected by the selector. Then, the corresponding voting weight α_n and the importance weight λ of the sample are updated and passed to the next selector $\mathcal{H}^{n+1}$. The weight λ increases if the sample is misclassified by the current selector or decreased otherwise. Finally, a strong classifier is build as a linear combination of N weak classifiers selected in individual selectors.

2.2 WaldBoost

The WaldBoost algorithm [10] is an off-line training algorithm which combines the AdaBoost training and Wald's sequential decision theory [12]. Its training goal is to minimize the training error as in AdaBoost but at the same time to minimize the evaluation time.

The WaldBoost algorithm uses outputs of weak classifiers found by AdaBoost as measurements (i.e. it uses AdaBoost as a measurement selector). The classifier is evaluated after each measurement and the decision is drawn or another measurement is taken

$$H_n(x) = \begin{cases} +1, & f_n(x) \geq \theta_B^{(n)} \\ -1, & f_n(x) \leq \theta_A^{(n)} \\ \text{continue}, & \theta_A^{(n)} < f_n(x) < \theta_B^{(n)} \end{cases} \tag{2}$$

where $f_n(x)$ is defined in Eq. (1). The goal of training is to find the proper weak classifiers h_n and the thresholds $\theta_A^{(n)}$ and $\theta_B^{(n)}$. The thresholds can be computed given the classifier response function $f_n(x)$. From the Wald theory, we are looking for two thresholds on the likelihood ratio R_n. Unfortunately, R_n is difficult to estimate due to the high dimensionality of both probability densities. Instead, a projection to a one dimensional space is used [10]

$$R_n(x) \cong \hat{R}_n(x) = \frac{p(f_n(x)|y=-1)}{p(f_n(x)|y=+1)} \ . \tag{3}$$

However, the training process rebuilds repeatedly the training and the validation set using bootstrapping (i.e. already decidable training samples are replaced by those which could not be decided yet). As the validation set used for estimating the thresholds changes, direct density estimation gives $p(f_n(x)|y = C, \rightarrow n)$ where $C \in \{-1,+1\}$ and $\rightarrow n$ stands for the condition that the sample has not been decided up to training step n, instead of desired $p(f_n(x)|y = C)$. Using Bayes formula we get

$$\hat{R}_n(x) = \frac{p(f_n(x)|y=-1, \rightarrow n)p(\rightarrow n|+1)}{p(f_n(x)|y=+1, \rightarrow n)p(\rightarrow n|-1)} \tag{4}$$

which leads to estimation of the likelihood ratio taking into account the bootstrapping.

From Wald's theory the thresholds $\theta_A^{(n)}$ and $\theta_B^{(n)}$ are estimated using $\hat{R}_n$ by finding thresholds for which $\hat{R}_n \geq A$ or $\hat{R}_n \leq B$ respectively where $A = (1-\beta)/\alpha$ and $B = \beta/(1-\alpha)$, and α is allowed false negative rate and β allowed false positive rate of the classifier specified by the user. A practical way of estimating the thresholds is to look for such values of $f_n(x)$ for which the ratio of negative and positive samples multiplied by the correction factor which takes the discarded samples into account, fulfills the conditions.

3 On-line WaldBoost

The proposed on-line WaldBoost algorithm combines on-line boosting and Wald's sequential analysis described in Sec. 2. The general training scheme is shown in Fig. 1. As in Sec. 2.1 the selectors are updated using the actual training sample, the weak classifier is chosen as the best classifier in the selector, and

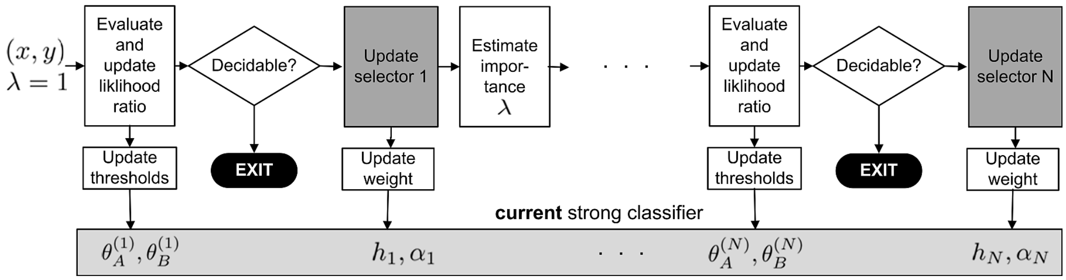


Figure 1. Training scheme of the on-line WaldBoost algorithm.

the sample importance weight λ reflects the difficulty of the sample. The main difference is that the training can be terminated earlier if the Wald conditions hold, i.e. the sample is used for updating only those selectors to which it is passed undecided.

In order to find the Wald thresholds $\theta_A^{(n)}$ and $\theta_B^{(n)}$ the likelihood ratio $\hat{R}(x)$ from Eq. 4 has to be estimated. In the off-line training the statistics are computed on an independent validation dataset. The on-line training offers an elegant way to compute an unbiased estimate of the statistics using the given sample only. The idea is to use the current training sample first as a test sample (not seen before) to update the Wald statistics before it is used for training the strong classifier. The probabilities $p(\rightarrow n|C)$ can be estimated by computing the portion of samples seen so far and not decided until n-th selector. The densities $p(f_n(x)|y = C, \rightarrow n)$ are estimated from the samples which are not decided until the n-th selector only. In our implementation they are approximated by Gaussians. Given these probabilities and α and β parameters the thresholds $\theta_A^{(n)}$ and $\theta_B^{(n)}$ are estimated as in Sec. 2.2. However, since a feature-switch in selector k causes a wrong estimate of the statistics of subsequent selectors, the statistics are reset where selectors $n \geq k$, i.e. $p(f_n(x)|y = C, \rightarrow n)$ is set to the uniform distribution and $p(\rightarrow n|C) = 0.5$ for $C \in \{-1, +1\}$.

This training scheme allows for classifier speedup in both training and evaluation compared to the original on-line boosting. Moreover, the number of selectors can be set to a high number and the real classifier complexity (i.e. number of weak classifiers used) is controlled automatically.

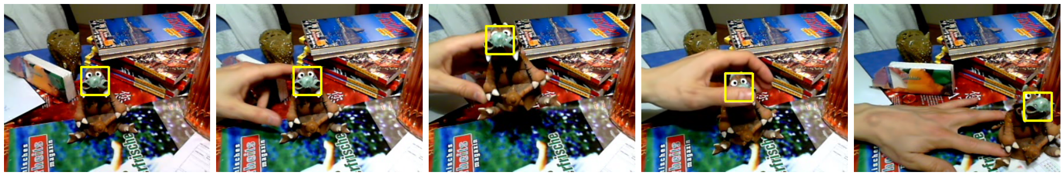
4 Experiments

The properties of the proposed on-line WaldBoost algorithm are demonstrated on the task of visual object tracking. It is formulated as a binary classification problem where the classifier is used to distinguish the object from the local background [3]. To be adaptive to the object appearance changes, updates of the classifier are performed – here we replaced the on-line boosting algorithm with the on-line WaldBoost algorithm proposed in this paper. For the on-line WaldBoost classifier $N = 50$ selectors were used which selects from a global feature pool of $M = 250$ weak classifier corresponding to Haar-like features (same parameters as in [3]). The Wald parameters were set to $\alpha = 0.02$ and $\beta = 0$.

Fig. 4 shows a challenging tracking sequence including appearance changes of the object as well as object occlusions on a complex background. The second row depicts the confidence maps of the classifier. Since we use no motion model (cf. [14]), a confidence map is computed by evaluating the classifier at all positions within a local search region. The position of the object corresponds to the maximum in the confidence map. The values equal to zero show the positions rejected before reaching the end of the classifier sequence.

These early decisions lead to the speedup shown in Fig. 5. The speedup is calculated as $N/\tilde{N}$, where $\tilde{N}$ is the average number of weak classifiers used before the decision is reached over the whole search region. If all values are equal to zero the object is considered to be lost. If the object is “stable” in the scene, the speedup is continuously increasing, since background patches can be discarded early. On average, we achieved a speedup of a factor of 5 to 10 without suffering a loss in tracking quality, i.e. we never discard the maximum peak of the confidence map, so the tracking results are exactly the same as reported in [3]. The same tracking results are reached also for the other tracking sequences used in [3, 8]. However, the speedup is not that big, since the object and the background changes a lot. Nevertheless, in the training the complexity of the classifier can still be determined automatically.

In general, the achieved speedup depends on dynamically changing problem difficulty and how often the Wald statistics have to be reseted (e.g. at frame 2706). Further, higher values of α lead to more speedup but having the risk of loosing the object if it changes its appearance too fast. The achieved speedup (≥ 1) can



(a) Frame 1 (b) Frame 400 (c) Frame 2000 (d) Frame 2759 (e) Frame 3659

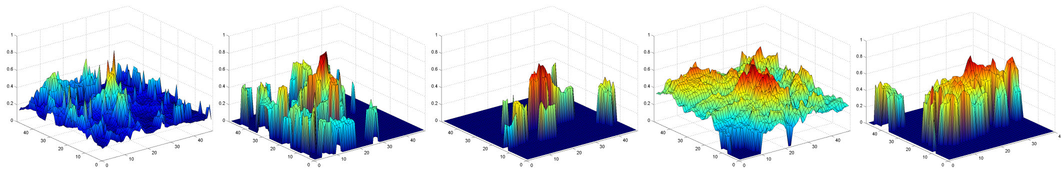


Figure 2. Tracking of an object (1st row) and the classifier response map within the search window (2nd row). Values equal to 0 mean early rejection, i.e. saving of the computation time.

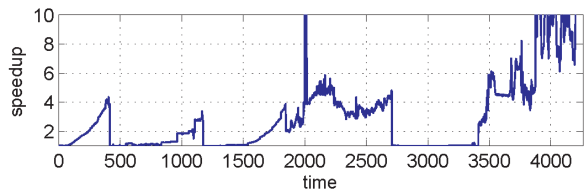


Figure 3. Speed-up compared to the non-sequential on-line boosting approach [3].

be used for instance for extending the search region to handle faster movements or to include more degrees of freedom like scale.

5 Conclusions

In this paper we have shown how to extend the on-line boosting algorithm using Wald's sequential decision theory. The proposed on-line WaldBoost training algorithm is able to control the classifier complexity depending on the problem difficulty. Moreover, the evaluation speed is increased through the sequential nature of the classifier. We tested the on-line WaldBoost algorithm on a visual tracking problem. The speedup is significantly improved compared to the non-sequential on-line boosting. We are confident that other applications (e.g. improving object detectors) can benefit as well. We are currently working on a refined statistic resetting strategy for a weak classifier switch and a method to adapt α on-line.

References

[1] Y. Freund and R. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 1999.

[2] H. Grabner and H. Bischof. On-line boosting and vision. In *Proc. CVPR*, volume 1, pages 260–267, 2006.

[3] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *Proc. BMVC*, volume 1, pages 47–56, 2006.

[4] H. Grabner, C. Leistner, and H. Bischof. Time dependent on-line boosting for robust background modeling. In *Proc. Int. Conf. on Comp. Vision Th. and App.*, 2007.

[5] H. Grabner, P. Roth, and H. Bischof. Is pedestrian detection realy a hard task? In *Proc. PETS*, 2007.

[6] C. Huang, H. Ai, S. Lao, and Y. Li. High-performance rotation invariant multiview face detection. *IEEE Trans. on PAMI*, 29(4):671–686, 2007.

[7] 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.

[8] J. Lim, D. Ross, R. Lin, and M. Yang. Incremental learning for visual tracking. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Neural Information Processing Systems*, number 17, pages 793–800. MIT Press, 2005.

[9] N. Oza and S. Russell. Online bagging and boosting. In *Proc. Artificial Intelligence and Statistics*, pages 105–112, 2001.

[10] J. Šochman and J. Matas. Waldboost - learning for time constrained sequential detection. In *Proc. CVPR*, volume 2, pages 150–157, 2005.

[11] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, volume I, pages 511–518, 2001.

[12] A. Wald. *Sequential analysis*. Dover, New York, 1947.

[13] B. Wu and R. Nevatia. Improving part based object detection by unsupervised, online boosting. In *Proc. CVPR*, pages 1–8, 2007.

[14] L. Xj, T. Yamashita, S. Lao, M. Kawade, and F. Q. Online real boosting for object tracking under severel appearance changes and occlusion. In *Int. Conf. on Acoustics, Speech and Signal Proc.*, pages 925–928, 2007.