Pattern Recognition Letters 40 (2014) 11–18

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A biased selection strategy for information recycling in Boosting cascade visual-object detectors [☆]

Chensheng Sun, Jiwei Hu, Kin-Man Lam [*]

Center for Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, China

A R T I C L E I N F O

Article history:
Received 28 May 2013
Available online 12 December 2013

Keywords:
Boosting
Cascade
Object detection
Information recycling

A B S T R A C T

We study the problem of information recycling in Boosting cascade visual-object detectors. It is believed that information obtained in the earlier stages of the cascade detector is also beneficial for the later stages, and that a more efficient detector can be constructed by recycling the existing information. In this work, we propose a biased selection strategy that promotes re-using existing information when selecting weak classifiers or features in each Boosting iteration. The strategy used can be interpreted as introducing a cardinality-based cost term to the Boosting loss function, and we solve the learning problem in a step-wise manner, similar to the gradient-Boosting scheme. Our work provides an alternative to the popular sparsity-inducing norms in solving such problems. Experimental results show that our method is superior to the existing methods.

1. Introduction

Boosting cascade detector, since being advocated by Viola and Jones (2004), has received a lot of research attention and is now a standard technique in visual-object detection. Using the sliding-window method for detection, the cascade detector brings three benefits. First, since visual-object detection is a rare-event detection problem with numerous negative windows but only a few positive windows in a typical image, a practical detector should have a very low false-positive rate and a decent recall rate. The cascade detector solves this problem by connecting a number of classifiers, also called stages. Each stage lets through almost all of the positive windows, but filters out a considerable portion of the negative windows. Second, the cascade detector can be very computationally efficient, since most negative windows can be rejected in the first several stages without invoking the later stages. The earlier stages of a cascade are usually very simple classifiers, while the later stages are more powerful but also more costly. Third, the large number of negative samples can be effectively exploited for training, since a later stage is trained using those samples accepted by the earlier stages. Viola and Jones (2004) also used a Boosting method to train each stage. The Boosting method (Friedman et al., 2000) manages to build a strong classifier by selecting a few weak classifiers from a very large pool of weak classifiers. By limiting each weak classifier to use a single feature, the Boosting method performs feature selection, and therefore improves the efficiency of the detector: Only those selected features need to be extracted at detection time. The feature-selection property of Boosting is well accepted in the field, as is illustrated in Yin et al. (2005) and Frst et al. (2008), etc.

Although a later stage in a cascade detector is trained using only those samples accepted by the earlier stages, those samples are not accepted with equal confidence. For example, the previous stage is trained until a threshold on the classifier's score can be found to achieve a 99.5% recall rate and a 50% false acceptance rate at the same time, and a new training set is collected by drawing samples from the true positives and false positives of this classifier. It can be observed from Fig. 1 that, in evaluating the previous classifier on this new dataset, the positive set has a much higher average score than the negative set does. This fact, first observed in Xiao et al. (2003), implies that the earlier stages contain some information that could be beneficial for solving the learning problems of the later stages. Since this information is already extracted by the earlier stages, it can be recycled at a very small extra cost when evaluating the later stages. Hence, the complexity of the later stages can be reduced, thereby improving the detector's overall efficiency. This fact has been exploited in the Boosting chain (Xiao et al., 2003) and the multi-exit cascade (Pham et al., 2008) by using the strong classifier of the immediate previous stage as the first weak classifier in a new, later stage. The nested cascade (Huang et al., 2004) also recycles the previous strong classifier's score by using it as the feature for the first weak classifier in the new stage.

Recent advances interpret Boosting as a convex optimization problem (Demiriz et al., 2002; Shen and Li, 2010), but one that works with a semi-infinite number of features. In this formulation, a weak classifier's score is used as a feature, and a new feature

☆ This paper has been recommended for acceptance by A. Petrosino.
* Corresponding author. Tel.: +85 2 27666207; fax: +85 2 23628439.
E-mail address: enkmlam@polyu.edu.hk (K.-M. Lam).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.12.006

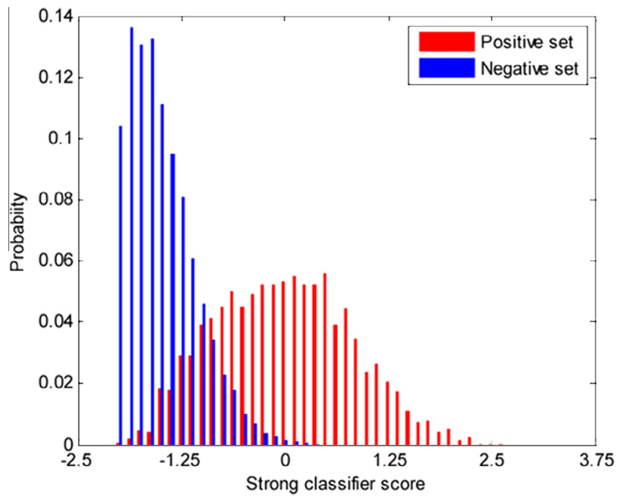

Fig. 1. The distribution of the previous-stage strong classifier's scores on the training set for the new stage.

space is spanned by the almost infinite number of possible weak classifiers. The goal is to find a linear classifier in this feature space to minimize an empirical loss function – for example, the hinge loss for LPBoost (Demiriz et al., 2002), and the exponential loss for AdaBoost (Shen and Li, 2010). The column-generation technique is employed to solve this semi-infinite programming problem, which divides a Boosting iteration into a feature-generation phase where new weak classifiers are trained and selected, and a problem-solving phase where the weights of the selected weak classifiers are adjusted to optimize the loss function. These methods are also referred to as "totally-corrective Boosting methods", since all the selected weak classifiers' weights are updated in every Boosting iteration. Therefore it is quite straightforward to re-use the weak classifiers obtained in the earlier stages of a cascade detector in a new stage, simply by optimizing their weights for the new training data. A recent work (Wang et al., 2010) implemented this idea and proved the efficacy of the method. Since the weights of the recycled weak classifiers are adjusted according to the new training data, fewer additional weak classifiers are needed in the new stage than by directly recycling the strong classifier scores, resulting in a more efficient detector. In addition, L-1 regularization is performed on the weak classifiers' weights such that the solution is a sparse vector, i.e. not all of the previous weak classifiers are actually used in the new stage.

In this work, we propose a step-wise method for recycling the weak classifiers and the features. The intuition behind this idea is very simple: In each iteration of the Boosting training process, we choose to either add a new weak classifier or adjust the weight of an existing weak classifier. A new weak classifier is introduced if and only if it shows a significant advantage over re-using an existing weak classifier, i.e. our choice in the Boosting iteration is biased towards the recyclable weak classifiers. This approach fits into the step-wise gradient-Boosting framework and is easy to implement, but the weights of all the selected weak classifiers can be adjusted in a similar way to the totally-corrective Boosting. Compared to the totally-corrective approach, the impact of our method is two-fold. First, the newly introduced weak classifiers are less affected by the recycled weak classifiers, since our approach interleaves the new-weak-classifier steps and the recycling steps, rather than introducing all previous weak classifiers at the beginning. Second, our approach naturally selects a subset of the previous weak classifiers, without relying on sparsity-inducing L-1 regularization as in the totally-corrective approach. Furthermore, the same idea is employed to recycle features, i.e. training new weak classifiers using features that have already been extracted. Recycling features cannot be achieved using the totally-corrective approach, and according to our knowledge, our proposed approach has not been studied in any previous work. In practice, the cost of evaluating a classifier is usually much lower than that of extracting a feature. Therefore, it is reasonable to fully exploit the potential of a feature, for example, by training a complex and more powerful classifier using that feature, or by including several weak classifiers in the Boosting strong classifier that share this feature. We have managed to recycle features using the latter approach, simply by biasing our selection of weak classifiers towards those using features that have already been extracted. Our biased selection strategy can be interpreted as optimizing a loss function with a cardinality regularization term. Rather than introducing sparsity using L-1 regularization, we directly penalize the number of unique weak classifiers or features in the detector. The resulting problem is approximately solved by our proposed step-wise approach.

The rest of this paper is organized as follows. In Section 2, we review several related works. Then, in Section 3 we introduce our algorithm, for recycling weak classifiers and features. Several interesting properties of our algorithm will also be discussed. We then experimentally prove the efficacy of our algorithm in Section 4. Section 5 concludes the paper with discussions, and also indicates our future work on this topic.

2. Related works

We take the most popular Boosting algorithm, the AdaBoost, as an example. Given a large set of available weak classifiers $\mathcal{F} = \{f_i\}$, the AdaBoost algorithm finds an optimal function $F(\mathbf{x})$ to optimize a loss functional $L(F)$. The function $F(\mathbf{x})$ is represented by a linear combination of a subset of the weak classifiers, and the loss functional is approximated by the empirical loss on the training set, such that the problem becomes:

$$\begin{aligned} \min_F \quad & L(F) = \sum_{j=1}^{M} \exp(-y_j F(\mathbf{x}_j)) \\ \text{s.t.} \quad & F(\mathbf{x}) = \sum_{i=1}^{N} w_i f_i(\mathbf{x}), f_i(\mathbf{x}) \in \mathcal{F}, \end{aligned} \tag{1}$$

where $\{(\mathbf{x}_j, y_j)\},\ j = 1, \dots M$, is a given training set with M training samples, each with a feature vector $\mathbf{x}_j$ and a ground-truth label $y_j \in \{+1, -1\}$. The gradient-Boosting scheme solves this problem using a step-wise approach. For example, the discrete AdaBoost assumes that the weak classifiers are binary, i.e. $f_i(\mathbf{x}_j) \in \{+1, -1\}$, and at the $n+1$st Boosting iteration, with the given current strong classifier $F^n(\mathbf{x}) = \sum_{i=1}^{n} w_i f_i(\mathbf{x})$, a new weak classifier f_{n+1} is selected to best approximate the functional gradient-descent direction. The weight of the new weak classifier w_{n+1} is also chosen to optimize the loss function:

$$\begin{aligned} & f_{n+1} = \arg\max_f \left(-f \cdot \frac{\partial L}{\partial F}\right), \quad w_{n+1} = \arg\min_w L(F + w f_{n+1}), \\ & \text{and} \quad \frac{\partial L}{\partial F} = \sum_{j=1}^{M} -y_j \exp\left(-y_j F(\mathbf{x}_j)\right). \end{aligned} \tag{2}$$

Then, the strong classifier is updated by $F^{n+1} = F^n + w_{n+1} f_{n+1}$. In a Boosting cascade visual-object detector, each stage is such an ensemble of weak classifiers. In an ordinary cascade, each stage is trained starting from $F^0(\mathbf{x}) = 0$, and new weak classifiers are added sequentially. To re-use the information from the previous stage, the current stage can be trained starting from the previous strong classifier's score instead of starting from 0. Denote $F_t(\mathbf{x})$ as the strong classifier for the tth stage, the strong classifier for the