Pattern Recognition Letters 68 (2015) 63–69

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Design of coupled strong classifiers in AdaBoost framework and its application to pedestrian detection☆

Kang-Kook Kong, Ki-Sang Hong*

Image Information Processing Laboratory, POSTECH, San 31 Hyoja-Dong, Pohang 790-784, Republic of Korea

ARTICLE INFO

Article history:
Received 19 January 2015
Available online 2 September 2015

Keywords:
AdaBoost
Coupled strong classifier
Complementarity
Pedestrian detection

ABSTRACT

In the AdaBoost framework, a strong classifier consists of weak classifiers connected sequentially. Usually the detection performance of the strong classifier can be improved increasing the number of weak classifiers used, but the improvement is asymptotic. To achieve further improvement we propose coupled strong classifiers (CSCs) which consist of multiple strong classifiers connected in parallel. Complementarity between the classifiers is considered for reducing intra- and inter-classifier correlations of exponential loss of weak classifiers in the training phase, and dynamic programming is used during the testing phase to compute efficiently the final object score for the coupled classifiers. In addition to CSC concept, we also propose using Aggregated Channel Comparison Features (ACCFs) that take the difference of feature values of Aggregated Channel Features (ACFs), enabling significant performance improvement. To show the effectiveness of our CSC concept, we apply our algorithm to pedestrian detection. Experiments are conducted using four well-known benchmark datasets based on ACFs, ACCFs, and Locally Decorrelated Channel Features (LDCFs). The experimental results show that our CSCs give better performance than the conventional single strong classifier for all cases of ACFs, ACCFs, and LDCFs. Especially our CSCs combined with ACCFs improve the detection performance significantly over ACF detector, and its performance is comparable to those of the state-of-the-art algorithms while using the simple ACF-based features.

1. Introduction

AdaBoost is a backbone framework to reach top performance and high speed in object detection, especially in pedestrian detection [4,15]. A strong classifier in the AdaBoost framework consists of weak classifiers connected sequentially. Usually the detection performance of the strong classifier can be improved by employing more weak classifiers, but its performance is asymptotic [13,23]. To improve the performance further, one can use multiple strong classifiers that are learned independently; in this case the relationship between the classifiers is ignored, so their detection results can overlap: some true positives missed by one classifier can be also missed by another classifier. For additional improvement we propose coupled strong classifiers (CSCs) to connect multiple strong classifiers in parallel.

To couple strong classifiers effectively, we introduce complementarity between the classifiers. Our CSCs consist of two kinds of classifiers: independent and dependent. An independent classifier is defined as the first classifier that is learned independently in the conventional way; the rest of the classifiers are dependent classifiers, and are learned to be complementary to previously-learned independent and dependent classifiers. Without the dependent classifiers, our CSCs are equivalent to a conventional single strong classifier in the AdaBoost framework. Based on the complementarity, our independent and dependent classifiers of CSCs are coupled, and an efficient method is used to compute the final object score for the coupled classifiers.

The complementarity concept is not entirely new; it is related to the diversity concept of ensemble learning [31]. However, the diversity in ensembles is achieved by applying several strategies that use different subsets of training data, different subsets of the available features, different parameters of the classifier, or different weak classifiers. In contrast, our CSCs are naturally related to many variants of AdaBoost [31] from the viewpoint of improving detection performance of AdaBoost. However, most of them focus on design of weak classifiers, on methods to select weak classifiers, or on methods to update the training sample weights, where a single strong classifier is assumed.

From the viewpoint of using multiple classifiers, our CSCs are related to multi-models such as occlusion-specific models [21], multiple object-scale (resolution-specific) models [2,3,29], and pose-aware models [1]. However, the goal of the multi-models is to learn

☆ This paper has been recommended for acceptance by R. Davies.
* Corresponding author. Tel.: +82- 054- 279- 2881; fax: +82- 054- 279- 5588.
E-mail address: hongks@postech.ac.kr (K.-S. Hong).

http://dx.doi.org/10.1016/j.patrec.2015.07.043
0167-8655/

multiple classifiers each of which corresponds to a sub-category that is defined explicitly or implicitly in a given object category, whereas our goal is to design a classifier that is complementary to the previously-learned classifiers and to devise a method to couple those classifiers. In our framework each classifier of multi-models can be considered as an independent classifier of our CSCs.

To show the effectiveness of our CSC concept, we apply our algorithm to pedestrian detection. Pedestrian detection in natural scenes is one of the most active research areas, but it is also a challenging problem owing to the non-rigid properties of pedestrians, to their variations of pose, and to the presence of multiple object scales, occlusions, and cluttered backgrounds. For the last decade, many algorithms [4,15,19] have been proposed to solve these problems, but research continues.

Implementation of our CSCs is based on Aggregated Channel Features (ACFs) [11] and Locally Decorrelated Channel Features (LDCFs) [22]. ACFs are one of the best features in terms of the detection performance and feature computation speed, among the features proposed so far including Haar-like features [27], HOG [7], HOG+LBP [28], channel features [13,32], motion features/optical flow [8,24] and covariance features [26]. Many recent features are also based on ACFs, including spatially pooled features [23], combined features [4], and LDCFs. The LDCF detector uses extended-channel features obtained by applying an efficient feature transform that removes local correlations in ACFs. It also shows state-of-the-art performance. Based on the experimental results on top of ACFs and LDCFs, we can expect improvement in detection performance of our CSCs when using other features.

The main ideas of our CSCs are explained in Section 2. Experimental results of pedestrian detection and implementation details are given in Section 3. The paper concludes in Section 4.

2. Proposed algorithm

2.1. Training phase

Our CSCs consist of two kinds of classifiers, independent and dependent classifiers. Before explaining these classifiers, we review conventional AdaBoost briefly. In the AdaBoost framework the strong classifier, H_M, consisting of M weak classifiers takes the form

$$H_M(x) = \sum_{m=1}^{M} \beta_m h_m(x), \tag{1}$$

where β_m and h_m $(m = 1, 2, \ldots, M)$ represent the mth voting weight and weak classifier respectively, and H_M also represents an accumulated object score from the first to the Mth weak classifier. The goal of the conventional AdaBoost in the training phase is to minimize the sum of certain losses of each training sample [5,20]. Given the pairs of training samples and their labels, (x_n, y_n) $(n = 1, 2, \ldots, N)$, the objective function J is defined as

$$J = \sum_{n=1}^{N} L\big(y_n, H_M(x_n)\big), \tag{2}$$

where $L(\cdot)$ represents some loss function. For each iteration in the AdaBoost framework, optimal voting weight and weak classifier are learned additively by minimizing the objective function in Eq. (2). In the mth iteration, the objective function to find the mth voting weight β_m and weak classifier h_m is defined as

$$\begin{aligned} J(\beta_m, h_m) &= \sum_{n=1}^{N} L\big(y_n, H_m(x_n)\big) \\ &= \sum_{n=1}^{N} L\big(y_n, H_{m-1}(x_n) + \beta_m h_m(x_n)\big), \end{aligned} \tag{3}$$

where $H_m = H_{m-1} + \beta_m h_m$ and H_{m-1} was learned in the $(m-1)$th iteration. If an exponential loss function is used, i.e., $L(y, H_m(x)) = \exp(-yH_m(x))$, then the objective function in Eq. (3) can be written as

$$\begin{aligned} J(\beta_m, h_m) &= \sum_{n=1}^{N} \exp\Big(-y_n\big(H_{m-1}(x_n) + \beta_m h_m(x_n)\big)\Big) \\ &= \sum_{n=1}^{N} L\big(y_n, H_{m-1}(x_n)\big) L\big(y_n, \beta_m h_m(x_n)\big). \end{aligned} \tag{4}$$

Therefore the objective function in Eq. (4) can be regarded as a correlation function over the training samples between the exponential loss $L(y, H_{m-1})$ by H_{m-1} and the exponential loss $L(y, \beta_m h_m)$ by $\beta_m h_m$. This correlation can be interpreted as intra-classifier correlation, because $\beta_m h_m$ and H_{m-1} belong to the same strong classifier. Therefore the independent classifier of our CSCs is learned by minimizing the intra-classifier correlation over the training samples. To learn the dependent classifier we expand this correlation, from intra-classifier correlation to inter-classifier correlation.

First, we consider the case of CSCs using one independent and one dependent classifiers. If we define the objective function of β_m^i and h_m^i for the independent classifier as

$$J\big(\beta_m^i, h_m^i(x)\big) = \sum_{n=1}^{N} L\big(y_n, H_{m-1}^i(x_n)\big) L\big(y_n, \beta_m^i h_m^i(x_n)\big), \tag{5}$$

then the update equation of the training sample weights [5,20] is given by

$$w_{m+1,n}^i = \exp\big(-y_n H_m^i(x_n)\big). \tag{6}$$

Extending this to the intra-classifier and the inter-classifier correlations, the objective function of β_m^d and h_m^d for the dependent classifier can be defined as

$$\begin{aligned} J\big(\beta_m^d, h_m^d(x)\big) &= \sum_{n=1}^{N} L\big(y_n, H_{m-1}^d(x_n)\big) L\big(y_n, \beta_m^d h_m^d(x_n)\big) \\ &\quad + \lambda \sum_{n=1}^{N} L\big(y_n, H_{m-1}^i(x_n)\big) L\big(y_n, \beta_m^d h_m^d(x_n)\big), \end{aligned} \tag{7}$$

where the first term corresponds to the intra-classifier correlation for the dependent classifier, the second term is the inter-classifier correlation between the independent and the dependent classifiers, and λ is a weighting parameter on the inter-classifier correlation. Using the objective function in Eq. (7), the optimal weak classifier of the dependent classifier is selected to minimize the intra-classifier and the inter-classifier correlations simultaneously. If $\lambda = \infty$, only inter-classifier correlation is considered, so the optimal weak classifier is learned by the same objective function as the independent classifier, regardless of its own dependent classifier. If $\lambda = 0$, only intra-classifier correlation is considered, so the dependent classifier is learned regardless of the independent classifier. The dependent classifier learned in this way is equivalent to the strong classifier learned independently.

The update equation of the training sample weights for the objective function of the dependent classifier in Eq. (7) is just simple modification given by

$$w_{m+1,n}^d = \exp\big(-y_n H_m^d(x_n)\big) + \lambda \exp\big(-y_n H_m^i(x_n)\big), \tag{8}$$

where the first term is from the intra-classifier correlation and the second is from the inter-classifier correlation. In fact the optimal weak classifier is selected to minimize the sum of classification errors of the training samples weighted by the training sample weights. The arrows in Fig. 1a represent update flows of the training sample weights. The weights for the independent classifier are updated along the conventional flows corresponding to Eq. (6), whereas the weights