Pattern Recognition Letters 34 (2013) 884–894

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

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

Coarse-to-fine multiclass learning and classification for time-critical domains

Teo Susnjak *, Andre Barczak, Napoleon Reyes, Ken Hawick

Massey University Albany, Private Bag 102904, North Shore 0745, New Zealand

ARTICLE INFO

Article history:
Available online 4 February 2013

Keywords:
Coarse-to-fine learning
Multiclass classification
Classifier ensembles
Boosting
Classifier cascades
Training runtime constraints

ABSTRACT

This paper presents a coarse-to-fine learning algorithm for multiclass problems. The algorithm is applied to ensemble-based learning by using boosting to construct cascades of classifiers. The goal is to address the training and detection runtime complexities found in an increasing number of classification domains. This research applies a separate-and-conquer strategy with respect to class labels, in order to realize efficiency in both the training and detection phases under limited computational resources, without compromising accuracy. The paper demonstrates how popular, non-cascaded algorithms like AdaBoost.M2, AdaBoost.OC and AdaBoost.ECC can be converted into robust cascaded classifiers. Additionally, a new multiclass weak learner is proposed that is custom designed for cascaded training. Experiments were conducted on 18 publicly available datasets and showed that the cascaded algorithms achieved considerable speed-ups over the original AdaBoost.M2, AdaBoost.OC and AdaBoost.ECC in both training and detection runtimes. The cascaded classifiers did not exhibit significant compromises in their generalization ability and in fact produced evidence of improved accuracies on datasets with biased-class distributions.

1. Introduction

In many complex classification domains, the performance bottleneck occurs in both the learning and the object detection phases. This is frequently experienced in fields like computer vision, natural language parsing and translation and in ever increasing real-time streaming settings. Often, it is the sheer volume of data needing to be processed which primarily contributes to protracted runtimes; but, in numerous occasions, the feature extraction component presents the highest computational costs.

Coarse-to-fine approaches are becoming more prevalent as solutions to these problems (Petrov et al., 2010), with the ever increasing number of dense domains. Coarse-to-fine strategies gradually increase complexity into its learning process, while minimizing loss of accuracy. Each subsequent model refines the previous one by using the outputs of the coarser stages. The classification phase is subsequently accelerated, since only a subset of features undergoes evaluation. Extensive research has shown, since the original contribution by Viola and Jones (2001), how this strategy can be applied to multi-layer classifiers in the form of cascades.

A large proportion of these kinds of real-world problems involve multiclass classification. A considerable corpus of research has attested to the ability of boosting and ensemble-based learning to provide robust and efficient solutions to binary class problems. Subsequently, effective and theoretically proven extensions of the popular AdaBoost (Freund and Schapire, 1995) algorithm have been proposed for multiclass problems. The modified versions of AdaBoost solve multiclass problems by reformulating them into series of binary class problems.

1.1. Related research

However, the limitations of these algorithms in their naive form, is that they are not designed to take advantage of coarse-to-fine approaches and are therefore ill suited for computationally demanding domains. The most current assessment by experts in the field concludes that the application of coarse-to-fine approaches. in the form of cascaded classifiers, to the challenges of multiclass learning is still an open problem (Zhang and Ma, 2012). The most common methods have involved either constructing separate parallel cascades for each class or building cascaded detector trees (Lienhart et al., 2003). Recently, the most notable contributions to the design of integrated multiclass cascaded classifiers have been by Verschae and del Solar (2010). In their research, a multiclass boosting algorithm called VectorBoost (Huang et al., 2007) was combined with a domain-partitioning weak classifier in order to produce compact and robust multi-view face detection classifiers.

In this paper, we propose a coarse-to-fine multiclass learning method that decomposes the training and detection task into cascades of boosted ensembles. We show how it can be used to convert single-layer multiclass algorithms like AdaBoost.M2 (Freund and Schapire, 1995), AdaBoost.OC (Schapire, 1997) and AdaBoost.ECC (Guruswami and Sahai, 1999) into multi-layer cascaded

* Corresponding author. Tel.: +64 9 4140800.
E-mail address: T.Susnjak@massey.ac.nz (T. Susnjak).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.01.011

classifiers. In order to maintain lower computational demands, we demonstrate how effective classifiers can be trained on difficult datasets using only decision stumps. We also present a new weak learner and show how it can be combined with the cascaded architecture to attain arbitrarily low training errors and accurate classifiers compared to current multiclass AdaBoost approaches.

The succeeding section describes the details of our cascaded multiclass architecture and the proposed weak learner. We subsequently describe the implementation of the experiments on 18 benchmark multiclass University of California at Irvine (UCI) (Frank and Asuncion, 2010) datasets, whose results we discuss in the remainder of the paper.

2. Multiclass cascade learning

The underlying principle of the proposed multiclass cascade is the *separate-and-conquer* strategy. As such, it bears some conceptual similarities to the rule-based RIPPER (Cohen, 1995) algorithm. The algorithm refines the cascaded classifier in a stepwise fashion by continuously removing class labels which it has learned best, in order to focus on more difficult classes.

Given a training set $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ where x_i is a sample vector $x_i \in X$ and y_i is a class label $y_i \in Y$ and $Y \in \{1, \ldots, k\}$, the multiclass algorithm constructs a cascade classifier H_{k-1} that consists of $k-1$ number of layers. Each layer i is specifically trained to predict a single class label c, as $H_i^c(x) \in Y$, otherwise the prediction is passed onto $H_{i+1}^c(x)$.

The cascaded classifier H_{k-1} is also two dimensional, whereby each layer contains within it a further *nested* cascade $H_{i,j}^c$ (Fig. 1). The nested cascade facilitates the coarse-to-fine learning of each layer, which enables the convergence to low training error rates. For clarity we will refer to each layer j of a *nested* cascade denoted as a pair of (M_j, B_j), as *nodes*. $M_j^\gamma(x)$ denotes a multiclass node whose prediction $\gamma \in Y$ and $\gamma \neq H_{i,j}^c(x)$, while B_j denotes a binary predictor whose output is $B_j(x) \in \{-1, 1\}$.

The cascade training proceeds as follows: in the first layer H_i, the initial multiclass node $M_{i,j}$ is trained on all samples using a generic multiclass algorithm until a predefined number of boosting iterations Φ are completed. Once this criterion is met, node $M_{i,j}$ is assessed for accuracy based on individual class error rates. The most separable class label γ is then identified for separation and assigned to node $M_{i,j}^\gamma$ as its target class for prediction.

All *correctly* predicted samples $(M_{i,j}^\gamma(x_i) = \gamma) \wedge (y_i = \gamma)$ are then trained against all the *incorrectly* predicted samples as class $(M_{i,j}^\gamma(x_i) = \gamma) \wedge (y_i \neq \gamma)$, using a binary learning approach. The resulting binary node $B_{i,j}$ functions as an auxiliary node to the multiclass node. The auxiliary node $B_{i,j}$ is trained until all the false positive samples with respect to class γ have been correctly learned. All correctly learned samples $(M_{i,j}^\gamma(x_i) = \gamma) \wedge (y_i = \gamma) \wedge (B_{i,j}(x_i) = 1)$, belonging to class γ are then removed from subsequent training of layer H_i. The training of node $M_{i,j+1}$ proceeds as with the initial node; howbeit, with $k-j$ class labels to learn.

The coarse-to-fine learning continues until all the most separable class labels and their instances have been removed from layer H_i. If at the end of the layer training, there remain instances which have not been correctly learned and associated with correct class labels, then a final binary node $FB_{i,j}$ is trained, whose output is $\{-1, 1\}$. In the training of the $FB_{i,j}$ node, the incorrectly learned samples are designated as negatives, while the instances belonging to the last remaining class label c as the positives. The binary training proceeds until a predetermined error rate is achieved. Subsequently, the class label c is assigned as the predictor for layer H_i^c. All instances belonging to class label c are eliminated from further training of layers H_{i+1}. In turn, each succeeding H_{i+1} layer with $k-i$ class labels reduces the computational demands and the complexity of class separability.

Removing class instances with the lowest error rate directly affects the rate at which the complexity of computation subsides and the class separability of the remaining sub-tasks are reduced. Class labels with a high accuracy result in the removal of a larger num-

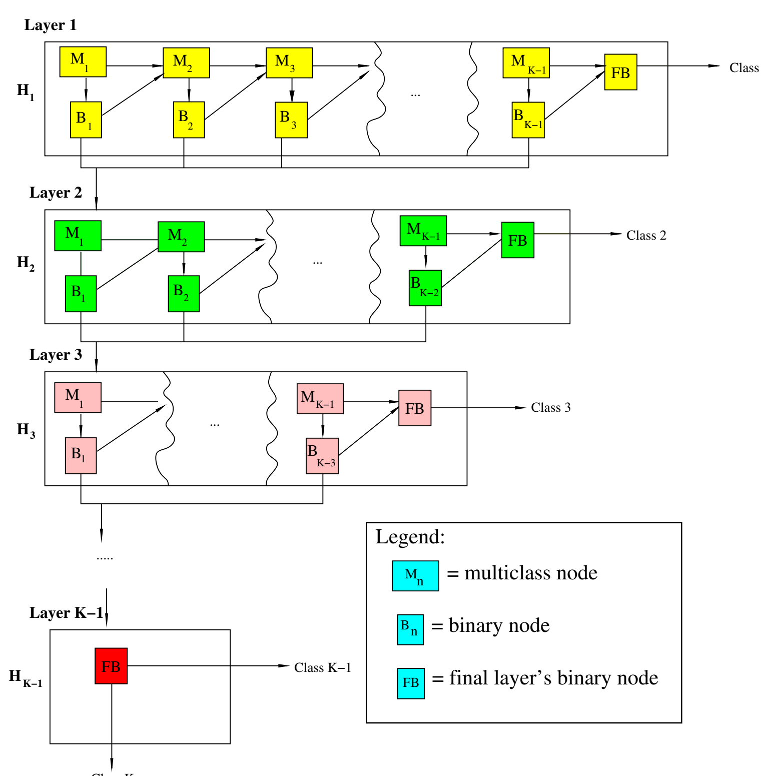

Fig. 1. Architecture of the proposed multiclass cascaded framework.