Information Sciences 179 (2009) 1298–1318

Contents lists available at ScienceDirect

Information Sciences

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

Incremental construction of classifier and discriminant ensembles

Aydın Ulaş [a,*], Murat Semerci [b,1], Olcay Taner Yıldız [c], Ethem Alpaydın [a]

[a] Department of Computer Engineering, Boğaziçi University, TR-34342 Bebek, Istanbul, Turkey
[b] Department of Computer Science, Rensselaer Polytechnic Institute, 110 8th Street, Lally Hall, Troy, NY 12180-3590, USA
[c] Department of Computer Engineering, Işık University, TR-34980 Şile, Istanbul, Turkey

ARTICLE INFO

Article history:
Received 29 March 2008
Received in revised form 15 December 2008
Accepted 27 December 2008

Keywords:
Classification
Classifier fusion
Classifier ensembles
Stacking
Machine learning
Voting
Discriminant ensembles
Diversity

ABSTRACT

We discuss approaches to incrementally construct an ensemble. The first constructs an ensemble of classifiers choosing a subset from a larger set, and the second constructs an ensemble of discriminants, where a classifier is used for some classes only. We investigate criteria including accuracy, significant improvement, diversity, correlation, and the role of search direction. For discriminant ensembles, we test subset selection and trees. Fusion is by voting or by a linear model. Using 14 classifiers on 38 data sets, incremental search finds small, accurate ensembles in polynomial time. The discriminant ensemble uses a subset of discriminants and is simpler, interpretable, and accurate. We see that an incremental ensemble has higher accuracy than bagging and random subspace method; and it has a comparable accuracy to AdaBoost, but fewer classifiers.

1. Introduction

1.1. Current trends in classifier combination

It is well known that there is no single classification algorithm that is always most accurate, and methods have been proposed to combine classifiers based on different learning algorithms [20,31]. Each algorithm has a different inductive bias, that is, makes a different assumption about the data and makes errors on different instances and by suitable combination, the overall error can be decreased.

There are several methods for model combination: the simple method is to use voting which corresponds to the sum rule, or another fixed rule, i.e., median, product, minimum, or maximum [27]. Methods based on resampling from a single data set, such as bagging [8] and AdaBoost [16], are not used to combine different learning algorithms. In stacking, fusion is done using a trained, second layer classifier which estimates the real output from the outputs of base classifiers [58]. Ting and Witten [55] propose the Multiresponse Linear Regression (MLR) algorithm, which combines the outputs of base learners linearly. In a mixture of experts (MoEs) architecture, models are local and a separate gating network selects one of the local experts based on the input [24].

Model combination, however, is no panacea, and models in the ensemble should be carefully chosen for error to decrease. In particular, model combination through averaging reduces variance [8], and hence error, but only if bias does not increase

* Corresponding author. Tel.: +90 212 359 4523/4524; fax: +90 212 287 2461.
E-mail address: ulasmehm@boun.edu.tr (A. Ulaş).
[1] Tel.: +1 518 276 8326; fax: +1 518 276 4033.

0020-0255/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2008.12.024

in the process, or if the concomitant increase in the bias is small with respect to the decrease in the variance. It is therefore essential that only those models that contribute to accuracy are added and the poorly performing ones are weeded out.

1.2. Subset selection in classifier combination

Additional to its effect on statistical accuracy, each additional model increases space and computational complexity. A new model may also be sensing/extracting a costly representation which can be saved if the model is considered redundant. Methods therefore have been proposed to choose a small subset from a large set of candidate models. Since there are 2^L possible subsets of L models one cannot try for all possible subsets unless L is small, and various methods have been proposed to get a reasonable subset of size $m < L$ in reasonable time.

Ensemble construction methods also differ in the criterion they optimize. Additional to the methods that directly optimize ensemble accuracy, heuristics have also been proposed as measures of "diversity" in pinpointing models that best complement each other, to allow diverse ones to be added and similar ones to be deemed redundant and pruned.

Ensemble construction can be viewed as an optimization problem [64,50] and methods proposed in the literature correspond to different search strategies in optimization: there are greedy "forward" algorithms that are incremental, and add one model at a time if the addition improves the criterion that is to be optimized. There are "backward" search methods that prune from a large set if the removal is not harmful. There are also "floating" methods that do both, as well as ones that use genetic algorithms whose operators allow both addition and deletion. A chronological review of major ensemble construction methods in more detail is deferred to Section 6.

1.3. Proposed methods for ensemble construction

In this paper, we discuss and evaluate two ensemble construction approaches:

(1) We incrementally construct an *ensemble of classifiers* as in the methods discussed above. On 38 data sets using 14 different base classifiers, we test the effect of: (i) the criterion to be optimized (accuracy, statistically significant improvement and two diversity measures, correlation and Q statistics), (ii) the search direction (forward, backward, floating), and (iii) the combiner (fixed voting, trained linear combiner).
(2) We incrementally construct an *ensemble of discriminants* where a classifier may be used for some of the classes but not for others. For example, in a three-class problem (C_1, C_2, C_3), C_1 may be linearly separable from C_2 and C_3 and therefore the first discriminant of the linear classifier is chosen. If C_2 is not linearly separable from C_1 and C_3, the second discriminant of the linear classifier cannot be used and a more complicated classifier (e.g., k-nearest neighbor) needs to be incorporated in the ensemble.

This work has two goals:

(1) First, we investigate the effect of various factors in ensemble construction using a wide variety of learning algorithms, data sets and evaluation criteria. We compare this algorithm with bagging, boosting and random subspace method.
(2) Second, we generalize the idea of subset selection to the level of discriminants, to check if it is applicable not only at the level of classifiers but also at the level of discriminants that make up a classifier.

The paper is organized as follows: in Sections 2 and 3, we introduce our algorithms. We give the details of our experiments in Section 4 and results in Section 5. We discuss related work in Section 6 and conclude in Section 7.

2. Constructing an ensemble of classifiers

2.1. The Icon algorithm

Our proposed algorithm Icon to choose m out of L base classifiers is greedy in that it starts with the empty set and Incrementally CONstructs an ensemble where at each iteration, it chooses among all possible classifiers the one that best improves the performance when added to the current ensemble. The performance is measured using an ensemble evaluation criterion, as we will discuss next. The algorithm stops when there is no further improvement. Of course, this does not guarantee the finding of the best subset, but this algorithm has polynomial complexity, $O(L^2)$, whereas exhaustive searching of all possible subsets, $O(2^L)$, is of exponential complexity. As we see shortly, despite its simplicity, this algorithm works very well in practice.

The pseudocode of the algorithm is given in Fig. 1. We start with $E^{(0)} = \emptyset$. At iteration t of the algorithm, we have ensemble $E^{(t)}$ containing t models. Given the set of remaining $L - t$ candidate models, $M_k \notin E^{(t)}$, we have new candidate ensembles for iteration $t + 1$ as $S_k^{(t+1)} \equiv E^{(t)} \cup M_k,\ k = 1, \ldots, L - t$. Among these, we choose the one that is preferred to all other candidates and is also preferred to the current ensemble (Line 5 of Fig. 1).