A new reverse reduce-error ensemble pruning algorithm

Qun Dai*, Ting Zhang, Ningzhong Liu

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

ABSTRACT

Although greedy algorithms possess high efficiency, they often receive suboptimal solutions of the ensemble pruning problem, since their exploration areas are limited in large extent. And another marked defect of almost all the currently existing ensemble pruning algorithms, including greedy ones, consists in: they simply abandon all of the classifiers which fail in the competition of ensemble selection, causing a considerable waste of useful resources and information. Inspired by these observations, an interesting greedy Reverse Reduce-Error (RRE) pruning algorithm incorporated with the operation of subtraction is proposed in this work. The RRE algorithm makes the best of the defeated candidate networks in a way that, the Worst Single Model (WSM) is chosen, and then, its votes are subtracted from the votes made by those selected components within the pruned ensemble. The reason is because, for most cases, the WSM might make mistakes in its estimation for the test samples. And, different from the classical RE, the near-optimal solution is produced based on the pruned error of all the available sequential subensembles. Besides, the backfitting step of RE algorithm is replaced with the selection step of a WSM in RRE. Moreover, the problem of ties might be solved more naturally with RRE. Finally, soft voting approach is employed in the testing to RRE algorithm. The performances of RE and RRE algorithms, and two baseline methods, i.e., the method which selects the Best Single Model (BSM) in the initial ensemble, and the method which retains all member networks of the initial ensemble (ALL), are evaluated on seven benchmark classification tasks under different initial ensemble setups. The results of the empirical investigation show the superiority of RRE over the other three ensemble pruning algorithms.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

It has been extensively reported in the literature that pooling together complementary classifiers is a desirable strategy to construct robust classification systems with good generalization performance [1–3]. Remarkable improvement in generalization performance has been observed from ensemble learning in a broad scope of applications, for example: face recognition [4], optical character recognition [5], scientific image analysis [6,7], medical diagnosis [8,9], financial time series prediction [6], military purposes [10], intrusion detection [11] etc.

Despite its remarkable performance, a main disadvantage of ensemble learning is that, generally, it is necessary to combine a large number of classifiers to make certain that the error converges to its asymptotic value. This brings on large computational requirements, including the training costs, the storage needs and the prediction time. And moreover, a large amount of communication costs are required, when classifiers are distributed over a

network. To alleviate these drawbacks, various ensemble pruning algorithms have been proposed [12–20].

However, the problem of selecting the subset of classifiers that has the best generalization performance, i.e. ensemble pruning, has proven to be an NP-complete problem [21,22]. Assuming that the generalization capability can be estimated based on some quantity measured on the pruning set, selective ensemble is a combinatorial search problem whose complexity grows exponentially with the size of the initial ensemble, since for an ensemble of T base classifiers, the number of nonempty subsets is $2^T - 1$. Therefore, finding the exact solution using enumerative algorithm is unfeasible for typical ensemble sizes. To solve this problem, it is proposed to use approximate algorithms that, with high probability, select near-optimal subensembles. In particular, genetic algorithms (GAs) [23,24] and semidefinite programming (SDP) [25] have been employed to address the problem of ensemble pruning. Despite the fact that the computational complexities of these algorithms are not exponential in the size of the initial ensemble any more, their computational costs are still rather large [26].

Several efficient ensemble pruning algorithms, which are based on greedy selection strategy in the space of subensembles, report good classification results and desirable generalization

---

* Corresponding author. Tel.: +86 25 84593038; fax: +86 25 84498069.
  E-mail address: daiqun@nuaa.edu.cn (Q. Dai).

performances [12–14,17]. These algorithms start with an empty (or full) initial ensemble and explore the space of different subsets by iteratively broadening (or shrinking) the initial ensemble by an individual classifier. The greedy exploration is guided by an evaluation measure in terms of either the classification accuracy or the diversity of the candidate subsets, which is the main ingredient of a greedy ensemble pruning algorithm and differentiates those algorithms that fall into this category. Some evaluation measures that have been successfully employed to guide the greedy selection process include Reduce-Error (RE) pruning [14], Kappa pruning [14], Complementarity Measure (CC) [17], Margin Distance Minimization (MDSQ) [17], Orientation Ordering (OO) [18], Boosting-Based pruning (BB) [27], and Uncertainty Weighted Accuracy (UWA) [28]. A more informative and thorough literature survey about existing ensemble pruning algorithms has been given in Section 2 of this paper.

Although greedy algorithms possess relatively high efficiency, as they only consider a very small subspace within the entire solution space, they might receive suboptimal solutions of the ensemble pruning problem [12–14,17,21,27]. Besides, another notable disadvantage of almost all the existing ensemble pruning algorithms, including greedy ones, consists in that they hastily discard all of the classifiers which are not selected into the pruned ensemble, causing a waste of useful resource and information. Since all the models in the initial ensemble are generated through training phase, which requires rather heavy consumptions of computing time and storage space. And moreover, in most cases, the classifiers which are not selected into the pruned ensemble, and therefore will be abandoned, constitute the majority of the component networks in the initial ensemble. From this point of view, the waste of resource and information is impressive and considerable.

Aiming at ameliorating the above mentioned drawbacks, an interesting greedy Reverse Reduce-Error (RRE) pruning algorithm incorporated with subtraction operation is proposed in this paper. Different from all the other currently existing ensemble pruning algorithms, in RRE, the classifiers not selected into the pruned ensemble are not simply and hastily abandoned. In contrast, the RRE algorithm makes the best of the defeated candidate networks in a way that, the Worst Single Model (WSM) is selected, which gets the largest pruned error within the set of defeated networks. And then, in the testing phase to the pruned ensemble, the opinion of WSM is considered and taken. In particular, the votes made by the WSM are subtracted from the votes made by those selected components within the pruned ensemble. The rationale of this step of vote subtraction lies in that, for most cases, the WSM might make a wrong predictive decision for the test sample. Therefore, it is expected that subtracting the votes made by WSM from the ones made by those selected members will increase the classification accuracy and enhance the generalization performance of the pruned ensemble furthermore.

And in the classical Reduce-Error (RE) pruning algorithm, the final selection operation is implemented according to the desired amount of pruning, where the first υ classifiers in the reordered sequence are selected [14]. While the subensemble selection operation of the proposed RRE algorithm is implemented differently, which is achieved based upon the validated error of all the available sequential subensembles on the pruning set.

In the original RE algorithm, backfitting is applied after each step of incorporation of a new component network. Backfitting sequentially attempts to replace one of the selected networks by another network not yet included in the subensemble. A replacement is made if a network that reduces the subensemble validated error is found in the pool of unselected networks. If one or more networks are replaced, then backfitting is applied repeatedly with a limit of 100 iterations. It is reported that, in bagging ensembles, when the training set is used as the pruning set, backfitting does not

significantly reduce the generalization error of the pruned ensembles [18]. And another severe defect of backfitting consists in its significant increase in the computational cost of the original RE pruning algorithm [26]. In contrast, in RRE pruning algorithm, backfitting is replaced with the selection step of a WSM based on its performance on the pruning set.

In some works, e.g. [26], ties are resolved by discarding the votes of the last classifiers included in the ensemble, one at a time, until the tie is broken. In the proposed RRE algorithm, the votes made by the WSM are subtracted, which might solve the problem of ties more reasonably and naturally.

Besides, in the testing phase of RRE algorithm, soft voting technique is employed [29], viz., the voting process is implemented by the summation of the computing results of all the selected models and the subsequent subtraction of the result of the WSM, and the final classification decision is made according to the results of this soft voting.

Moreover, we have proposed a novel Ensemble Pruning method based on BackTracking algorithm (EnPBT) in our previous work [30]. In comparison with other pruning methods including greedy algorithms, the pruned ensemble achieved with EnPBT algorithm generally possesses significantly stronger classification and generalization performance. However, we find an obvious defect in EnPBT that, the definition of its solution space contains many redundant solution vectors. And naturally, the solution space tree of EnPBT also contains many redundant solution vectors. This causes a number of redundant explorations of EnPBT algorithm, and finally affects its entire searching efficiency.

Aiming at improving the above mentioned defect of EnPBT algorithm, we further proposed a Modified Backtracking Ensemble Pruning Algorithm (ModEnPBT) [30]. In contrast, its solution space is compact with no repeated solution vectors. And naturally, its solution space tree is also concise, having no redundant solution vectors. Therefore, it possesses relatively higher searching efficiency in comparison with EnPBT algorithm.

Compared with the RRE pruning algorithm proposed in this work, EnPBT and ModEnPBT algorithm aim at improving classification and generalization performance of some state-of-the-art pruning methods, including greedy algorithms, based on Back-Tracking algorithm [30], whereas the proposal of RRE pruning algorithm aims at ameliorating those specific drawbacks of the classical RE pruning algorithm. Consequently, their designing motivations and starting points are totally different. With regard to the comparison of their performances, the three algorithms possess similar performances. In some cases, EnPBT or ModEnPBT performs the best [30], while in other cases, RRE pruning algorithm has the optimal performances. We could make choice from these three algorithms according to the actual application problem to be addressed.

The remains of this paper are structured as follows. A thorough discussion on the existing literature on the domain of ensemble pruning is provided in Section 2. Section 3 briefly reviews the classical Reduce-Error (RE) ensemble pruning algorithm. Section 4 presents the proposed Reverse Reduce-Error (RRE) pruning algorithm. Section 5 reports the results of experimental study, and from these experimental results, the final conclusions are drawn in Section 6.

2. A literature survey

During the past decade, many effective ensemble pruning approaches have been proposed. Roughly speaking, these approaches can be classified into four categories: (1) ordering-based pruning; (2) clustering-based pruning; (3) optimization-based pruning; (4) other pruning methods [31,32].