Abstract

Key Words: Bagging; CART; Classification Trees; Ensembles; Shrinking.

Bagging, or bootstrap aggregating, has been shown to improve the accuracy over that from a single CART tree. However, it has been noticed that bagging pruned trees does not necessarily result in better performance than bagging unpruned trees. This issue can be addressed by using the shrinking parameter for each base tree instead of pruning trees. Due to many duplicated observations in a bootstrap sample, the usual shrinking determined by cross-validation in bagging is so conservative that the resulting shrunk tree is not much different from the unshrunk tree, leading to their close performance. We propose to choose the shrinking parameter for each base tree by using only extra-bootstrap observations as test cases. For the digit data taken from Breiman et al. (1984), we find that our proposal leads to improved accuracy over that from bagging unshrunk trees.
Shrinking Classification Trees

1. Introduction

We consider the classification problem: suppose we have some observations $D = \{ (X_i, Y_i) \mid i = 1, \ldots, n \}$, where $X_i$ is a vector of some features or attributes and $Y_i$ is the corresponding class label. The goal is to learn a classifier from $f(X, Y)$, which can then be used to predict the class label $Y$ of any future observed attributes $X$. One popular tool is the classification or decision tree, such as C4.5 and CHAID (Breiman et al. 1984). Since a fully grown classification tree may over-fit the data and hence have a poor performance in predicting future observations, pruning and shrinking are adopted to avoid over-fitting. Bagging (Breiman 1996a) introduced bootstrapping aggregation (bagging) as a method to overcome this problem. However, some recent work (e.g. Bauer and Kohavi 1999) shows that bagging pruned trees does not necessarily yield a better performance than bagging unpruned trees does. In this paper we explore a similar issue in bagging shrunk/unshrunk trees. When does not necessarily yield a better performance than bagging unpruned trees does. In this paper we discuss this issue in bagging shrunk/unshrunk trees. When does not necessarily yield a better performance than bagging unpruned trees does. In this paper we discuss this issue in bagging shrunk/unshrunk trees.
In constructing a bagging ensemble, shrinking the base trees of any base tree, if needed, is determined through cross-validation based on the corresponding bootstrap sample. In Section 2, shrinking and bagging will be used to illustrate the resulting number of effective terminal nodes available from the output of the S-Tree function (Clark and Pregibon, 1992). Clark and Pregibon claim that heuristically one can also map k into the number of effective terminal nodes. We have implemented in S (environment) a cross-validation (CV) is used to choose a proper k. In this paper, as by default in S, k is fixed. The resulting number of effective terminal nodes is defined as classifying to class c for the node where \( \hat{p}(c) = \frac{1}{k} p(c|\text{parent}) + \frac{k-1}{k} p(c|\text{node}) \).

Shrinking works as follows:

1. In Clark and Pregibon (1992), which was implemented in S environment, we used, a node to be a convex combination of those of the parent and the frequencies in the child node (Clark and Pregibon, 1992). When k = 0, then there is no shrinking, i.e., results in a shrinking parameter of 1. 0 ≤ k ≤ 1 is the given size parameter, which is set to 0.1 by default in S. Clark and Pregibon claim that heuristically one can also map k into the number of effective terminal nodes. We will use the resulting number of effective terminal nodes available from the output of the S-Tree function (Clark and Pregibon, 1992) to represent the size of a (shrunk) tree throughout.

In constructing a bagging ensemble, shrinking the base trees for any base tree, if needed, is determined through cross-validation based on the corresponding bootstrap sample. In Section 2, shrinking and bagging will be used to illustrate the resulting number of effective terminal nodes available from the output of the S-Tree function (Clark and Pregibon, 1992). Clark and Pregibon claim that heuristically one can also map k into the number of effective terminal nodes. We have implemented in S (environment) a cross-validation (CV) is used to choose a proper k. In this paper, as by default in S, k is fixed. The resulting number of effective terminal nodes is defined as classifying to class c for the node where \( \hat{p}(c) = \frac{1}{k} p(c|\text{parent}) + \frac{k-1}{k} p(c|\text{node}) \).

Shrinking works as follows:

1. In Clark and Pregibon (1992), which was implemented in S environment, we used, a node to be a convex combination of those of the parent and the frequencies in the child node (Clark and Pregibon, 1992). When k = 0, then there is no shrinking, i.e., results in a shrinking parameter of 1. 0 ≤ k ≤ 1 is the given size parameter, which is set to 0.1 by default in S. Clark and Pregibon claim that heuristically one can also map k into the number of effective terminal nodes. We will use the resulting number of effective terminal nodes available from the output of the S-Tree function (Clark and Pregibon, 1992) to represent the size of a (shrunk) tree throughout.

In constructing a bagging ensemble, shrinking the base trees for any base tree, if needed, is determined through cross-validation based on the corresponding bootstrap sample. In Section 2, shrinking and bagging will be used to illustrate the resulting number of effective terminal nodes available from the output of the S-Tree function (Clark and Pregibon, 1992). Clark and Pregibon claim that heuristically one can also map k into the number of effective terminal nodes. We have implemented in S (environment) a cross-validation (CV) is used to choose a proper k. In this paper, as by default in S, k is fixed. The resulting number of effective terminal nodes is defined as classifying to class c for the node where \( \hat{p}(c) = \frac{1}{k} p(c|\text{parent}) + \frac{k-1}{k} p(c|\text{node}) \).

Shrinking works as follows:

1. In Clark and Pregibon (1992), which was implemented in S environment, we used, a node to be a convex combination of those of the parent and the frequencies in the child node (Clark and Pregibon, 1992). When k = 0, then there is no shrinking, i.e., results in a shrinking parameter of 1. 0 ≤ k ≤ 1 is the given size parameter, which is set to 0.1 by default in S. Clark and Pregibon claim that heuristically one can also map k into the number of effective terminal nodes. We will use the resulting number of effective terminal nodes available from the output of the S-Tree function (Clark and Pregibon, 1992) to represent the size of a (shrunk) tree throughout.

2. Shrinking and Bagging

Bagging ensembles in Section 2.
The effects of shrinking are different in a single tree and in bagging. For a single tree, shrinking greatly reduces the tree size and also improves the accuracy over the original tree. In other words, each base tree is shrunk independently. By default the shrinking procedure in bagging is this kind unless specified otherwise. At the nominal level (0.05), all the pairwise differences are statistically significant. It has been noticed that bagging pruned trees is not necessarily superior to bagging unpruned trees (Bauer and Kohavi 1998). We explored bagging shrunk/unshrunk trees using the digit data from Breiman et al. (1984, page 444). In the digit data there are 7 useful binary features and 17 noise binary variables. The response is one of ten digits from 0 to 9. As shown in Breiman et al. (1984, page 444), the accuracy of any classification tree is increased by using a bagging ensemble. The performance of any classification tree is increased by using a bagging ensemble. Hence, a classification tree can be generalized as many as possible by simulation. Hence, a significant improvement of accuracy and a reduced tree size is achieved. However, bagging shrunk trees seems to improve the accuracy over the original tree.
Shrinking Classification Trees

makes no obvious difference from bagging unshrunk trees, neither in performance of

To properly shrink the base trees when constructing a bagging ensemble

resultant bagging ensemble may have an improved performance. To answer this question in the next section.

Comparing the ratios of the shrunk and unshrunk tree sizes in a bagging ensemble and in a single shrunk tree (with p-value=0.0049), it

has a lower error rate than the single shrunk tree does (with p-value=0.001 from the paired t-test). If even

than bagging unshrunk trees (with p-value>0.05) from the paired t-test). If even

has an important implication: the usual shrinkage determined by CV in bagging is too

less shrinkage effect in the bagging ensemble is much more evident in Figure 2. Now the shrinkage effect in the bagging ensemble is much more evident

To investigate this, we did a simulation study using the digit data. Still we used

a training set of size 200 to construct a classifier. However, instead of using CV, we

used an independent tuning set of size 100 to select the shrinkage parameter for each

base tree. The resulting classifiers were again evaluated by a separate test set of size

1000. We independently repeated this process for ten times and the results are shown

in Figure 2. (In selecting the shrinkage parameter with bootstrap samples)

Thus leading to too

in section the shrinkage parameter with bootstraps samples), thus leading to too

that these duplicated observations result in too similar test and training sets in CV. It is suspected

words there are many replicated observations in a bootstraps sample. In other

access only 6.9% of the original observations appear in a bootstraps sample. In other

pruned trees. From the bootstraps loops (Efron and Tibshirani 1993), we know that on

and in a single tree (Figure 1), we will not observe that indeed there is less shrinkage effect in

Computing the ratios of the shrunk and unshrunk tree sizes in a bagging ensemble.

3. A Proposal for Shrinking in Bagging

in tree size. How could this happen? We will tackle this question in the next section.

In practice there is no luxury of having an extra tuning data set we are not willing to

process for ten times, and the results are shown

consequently shrinking the base trees.

in section the shrinkage parameter with bootstraps samples), thus leading to too

that these duplicated observations result in too similar test and training sets in CV. It is suspected

words there are many replicated observations in a bootstraps sample. In other

access only 6.9% of the original observations appear in a bootstraps sample. In other

pruned trees. From the bootstraps loops (Efron and Tibshirani 1993), we know that on

and in a single tree (Figure 1), we will not observe that indeed there is less shrinkage effect in

Computing the ratios of the shrunk and unshrunk tree sizes in a bagging ensemble.

3. A Proposal for Shrinking in Bagging

in tree size. How could this happen? We will tackle this question in the next section.

In practice there is no luxury of having an extra tuning data set we are not willing to

process for ten times, and the results are shown

consequently shrinking the base trees.

in section the shrinkage parameter with bootstraps samples), thus leading to too

that these duplicated observations result in too similar test and training sets in CV. It is suspected

words there are many replicated observations in a bootstraps sample. In other

access only 6.9% of the original observations appear in a bootstraps sample. In other

pruned trees. From the bootstraps loops (Efron and Tibshirani 1993), we know that on

and in a single tree (Figure 1), we will not observe that indeed there is less shrinkage effect in

Computing the ratios of the shrunk and unshrunk tree sizes in a bagging ensemble.

3. A Proposal for Shrinking in Bagging

in tree size. How could this happen? We will tackle this question in the next section.

In practice there is no luxury of having an extra tuning data set we are not willing to
A modification to 6-5-Bag is possible along the lines of Efron’s bootstrap 632 rule.

We term the final ensemble as extra-bootstrap bagging (eB-S-Bag).
Shrinking Classification Trees

to select the shrinkage parameter for each base tree not only is larger, but may also be more representative of the original sample. It is of direct interest in future studies to see whether an analogous extra-bootstrap pruning will increase the accuracy of bagging pruned trees. In addition to bagging, another ensemble called boosting has been proposed by Freund and Schapire. Though for some domains boosting can have a better performance than bagging does, another ensemble called boosting has been proposed by Freund and Schapire. (1996) Pruning will increase the accuracy of bagging pruned trees. In addition to bagging boosting, which is also of interest in future studies to see whether an analogous extra-bootstrap pruning will increase the accuracy of bagging pruned trees.

Acknowledgement

The author is grateful to the referees and Editors for helpful suggestions.

Reference


Shrinking Classification Trees


San Francisco, C.A.


S.J.M. Chambers and T.J. Hastie (eds), Wadsworth and Brooks/Cole, Monterey, CA.

IEEE Transactions on Pattern Recognition and Machine Intelligence 13, 340.

Shrinking Classification Trees

Figure 1: Boxplots for classification error rates and average sizes of single trees w/o shrink-
ing (Tree), single shrunk en trees (S-Tree), bagging unshrunk en trees (Bag), and bagging shrunk en trees (S-Bag) for the digit data.
Figure 2: Boxplots for classification error rates and average sizes of single trees without shrinking (Tree), single shrunk trees (S-Tree), bagging unshrunk trees (Bag), and bagging shrunk trees (S-Bag) for the digit data. The shrinkage parameter in each base tree in the bagging ensemble is selected by using a separate tuning data set with size=100.
Figure 3: Boxplots for classification error rates and average sizes of single trees w/o shrinking (Tree), single shrunk trees (S-Tree), bagging unshrunk trees (Bag), and the extra-bootstrap bagging ensemble (eB-S-Bag) for the digit data.