Maximum a posteriori pruning on decision trees and its application to bootstrap BUMPing

Jinseog Kim\textsuperscript{a,}\* , Yongdai Kim\textsuperscript{b} , Jongwoo Jeon\textsuperscript{b}

\textsuperscript{a}Statistical Research Center for Complex Systems, Seoul National University, Seoul 151-742, Republic of Korea
\textsuperscript{b}Department of Statistics, Seoul National University, Seoul 151-742, Republic of Korea

Received 26 February 2004; received in revised form 4 September 2004; accepted 24 September 2004

Abstract

The cost-complexity pruning generates nested subtrees and selects the best one. However, its computational cost is large since it uses holdout sample or cross-validation. On the other hand, the pruning algorithms based on posterior calculations such as BIC (MDL) and MEP are faster, but they sometimes produce too big or small trees to yield poor generalization errors. In this paper, we propose an alternative pruning procedure which combines the ideas of the cost-complexity pruning and posterior calculation. The proposed algorithm uses only training samples, so that its computational cost is almost same as the other posterior-based algorithms, and at the same time yields similar accuracies as the cost-complexity pruning. Moreover it can be used for comparing non-nested trees, which is necessary for the BUMPing procedure. The empirical results show that the proposed algorithm performs similarly as the cost-complexity pruning in standard situations and works better for BUMPing.

Keywords: Cost-complexity pruning; Decision tree; Cross-validation; BIC; Posterior probability; BUMPing

1. Introduction

Decision trees are popular models in applied statistics and machine learning (Breiman et al., 1984; Quinlan, 1993) since they are simple and easy to be interpreted. A typical
procedure of the decision tree induction consists of two steps, called growing and pruning. The growing step constructs a fully grown tree, which is usually done by finding the best split recursively until no more splits are possible. Then, the pruning step selects the best subtree of the fully grown tree.

Various pruning algorithms have been developed such as the cost-complexity pruning (CCP; Breiman et al., 1984), the reduced-error pruning (REP; Quinlan, 1987), Bayesian information criterion (BIC) or equivalently the minimum description length (MDL) pruning (Rissanen, 1978; Quinlan and Rivest, 1989; Mehta et al., 1995), the minimum error pruning (MEP; Niblett and Bratko, 1986) and so on. These pruning algorithms can be classified into two types. The first type uses holdout samples, where CCP and REP belong, and the second type including BIC/MDL and MEP uses only training samples. For the pruning methods of the first type, a sufficiently large amount of holdout samples should be required for selecting the best tree. If holdout samples are not available, one can resort to heuristic methods such as the V-fold cross validation (CV). But it demands additional computations and the final result is unstable. In contrast, the pruning methods of the second type do not pose these problems. However, the size of the finally selected tree tends to be unnecessarily large or small, which results in lower prediction accuracy and arises difficulty in interpreting the model. See Section 5 for empirical evidences.

The aim of this article is to develop a new pruning algorithm to resolve the aforementioned problems. The main idea of the proposed pruning algorithm is to combine the ideas of CCP and the Bayesian methods. The proposed algorithm first generates the sequence of nested trees in the same manner as CCP, and compares them via calculating their posterior probabilities to choose the best one. By doing so, we can reduce computing time and avoid the instability due to CV. Another advantage of our pruning algorithm is to be able to compare the non-nested trees. One such example is BUMPing (Tibshirani and Knight, 1999), which uses CV to compare the trees. Since CV may produce unstable results, the posterior calculations can help avoiding this pitfall. So, we can apply our pruning method within the loop of BUMPing and compare the posteriors of trees, and hence improve the performance of BUMPing.

In order to calculate the posterior probabilities of given trees, we could use BIC since it is known as an approximation of the posterior probability. However, for tree models, we observed that BIC sometimes but not often yields unnecessary large or small trees (see Table 3 in Section 5). Mehta et al. (1995) reported the similar results. This is partly because BIC internally puts equal prior probabilities on trees of different sizes. In such case, the approximations for large sized trees can be worse since the number of observations in each terminal node is not sufficiently large. The proposed algorithm overcomes this deficiency in BIC by putting the size-dependent priors and calculating the exact posterior probabilities.

The paper is organized as follows. Section 2 briefly describes the three well-known pruning algorithms—CCP, BIC and MEP. Section 3 explains the proposed pruning algorithm including the way of specifying prior probabilities and calculating posterior probabilities for given decision trees. Section 4 presents the BUMPing technique based on the proposed pruning algorithm. In Section 5, we give the results of empirical studies using real data sets. Finally, discussions follow in Section 6.
2. Review of pruning methods

In this section, we briefly review the three pruning methods based on CCP, BIC/MDL and MEP. The objective of this paper is to combine the ideas of these pruning methods to develop a new pruning method.

2.1. Cost-complexity pruning

Breiman et al. (1984) developed the minimal CCP which performs as follows:

1. Construction of the set of candidate subtrees $T_0, T_1, \ldots, T_K$, according to the cost-complexity measures.
2. Selection of the best tree $T^*$ among $T_0, T_1, \ldots, T_K$ according to the estimates of the generalization errors through V-fold CV or holdout pruning samples.

Let $T_{\text{max}}$ denote the fully grown tree for a given training data set. For any subtree $T$ of $T_{\text{max}}$ and $\lambda > 0$, Breiman et al. (1984) defined the cost complexity measure as follows:

$$R_\lambda(T) = R(T) + \lambda |\tilde{T}|,$$

where $R(T)$ is a misclassification error of $T$ calculated on the training data, and $|\tilde{T}|$ is the number of terminal nodes in $T$. For a given $\lambda$, let $T_\lambda$ be the optimal subtree of $T_{\text{max}}$ which minimizes $R_\lambda(T)$. Breiman et al. (1984) showed that for the increasing values of $0 = \lambda_0 < \lambda_1 < \cdots$, the corresponding sequence of the optimal subtrees $T_{\lambda_0}, T_{\lambda_1}, \ldots$ forms a nested structure. To choose the best tree among the nested subtrees, they recommended to estimate the optimal $\hat{\lambda}$ by minimizing the CV error or the holdout sample error.

2.2. Bayesian information criterion

Let $M_1, M_2, \ldots$ be given models and let $\pi(M_1), \pi(M_2), \ldots$ be the prior probabilities. For a given model $M$, the likelihood and prior of parameters are denoted by $P(Y|\Theta, M)$ and $\pi(\Theta|M)$, respectively. Then the posterior probability of the model $M$ can be computed by integrating out the parameter $\Theta$. That is,

$$P(M|Y) \propto \pi(M) \int P(Y|\Theta, M)\pi(\Theta|M)\,d\Theta.$$  \hspace{1cm} (1)

When the integral part of the Eq. (1) is very complex, it is not easy to obtain the closed form directly. So, we need to approximate $P(M|Y)$. For sufficiently large samples, the Laplace’s method induces that

$$\log P(M|Y) \approx \log P(Y|\hat{\Theta}, M) - \frac{d}{2} \log N,$$

where $d$ is the number of free parameters and $\hat{\Theta}$ is the MLE under the model $M$. Using this approximation, Schwarz (1978) defined BIC as a model selection criterion by

$$\text{BIC} = -2 \log P(Y|\hat{\Theta}, M) + d \log N,$$

which is an approximation of the negative of the log-posterior.
Rissanen (1978) proposed the MDL principle, which seeks to minimize the number of bits needed to describe the data over the class of models. The total description length is defined by

\[ l(Y, \Theta, M) = l(M) + l(\Theta|M) + l(Y|\Theta, M). \]

The first term of the above equation indicates the encoding length of the model \( M \), the second encodes the parameter, and the third encodes the data \( Y \) given the model and parameter. MDL chooses the model \( M \) and the parameter \( \Theta \) which minimize \( l(Y, \Theta, M) \) for given data \( Y \). Applying Shannon’s theory, we can show \( l(Y|\Theta, M) \) equals the negative log likelihood—\( \log P(Y|\Theta, M) \). In addition, if we accept the fact \( p(\Theta|M) \propto e^{-l(\Theta|M)} \), the MDL principle can be interpreted as the maximization of the logarithm of the posterior probability of the model, which is essentially the same as minimizing BIC. Since Rissanen (1978) first proposed MDL, many modifications have been suggested and applied successively by Wallace and Freeman (1987), Quinlan and Rivest (1989), Mehta et al. (1995) and Takeuchi (1997).

2.3. Minimum error pruning

Niblett and Bratko (1986) proposed a bottom-up approach for seeking a single tree that minimizes “the expected error rate on an independent data set”. For a \( K \)-class problem, the expected probability that an observation in node \( t \) belongs to the \( k \)th class is defined as

\[ p_k(t) = \frac{n_k(t)}{n(t)} \frac{n(t)}{n(t) + m + \pi_k} \frac{m}{n(t) + m}, \]

where \( n(t) \) is the number of observations in node \( t \), \( n_k(t) \) is the number of observations in node \( t \) whose class level is \( k \), \( \pi_k \) is the prior probability of the \( k \)th class, and \( m \) is a tuning parameter that determines the impact of the prior probability. Cestnik and Bratko (1991) named \( p_k(t) \) as \( m \)-probability estimate. Actually, \( p_k(t) \)s become Bayes estimates when we assume that the distribution of class levels are multinomial, and the prior probabilities are distributed according to the Dirichlet distribution.

By letting \( m = K \) and \( \pi_k = 1/K, k = 1, \ldots, K \), i.e., the same prior probabilities for the all classes, Niblett and Bratko (1986) defined the expected error rate as

\[ EER(t) = \min_k \{1 - p_k(t)\} = \min_k \left\{ \frac{n(t) - n_k(t) + K - 1}{n(t) + K} \right\}. \]

By computing the above measure for each internal node, the minimum error pruning (MEP) is carried out as follows:

1. Start with the full grown tree.
2. For each internal node \( t \),
   - compute the expected error \( EER(t) \),
   - compute the error of the sub-tree given by \( EER(T_t) = \sum_{s \in T_t} p_s EER(s) \), where \( p_s = n(s)/n(t) \) is the proportion of the number of samples in node \( s \),
   - a node \( t \) is pruned if \( EER(t) \leq EER(T_t) \), or not otherwise.
3. Maximum a posteriori pruning algorithm

In this section, we propose a new pruning algorithm which combines the ideas of CCP and Bayesian approach. First, we explain how to calculate the posterior probability of a given tree, and then explain how to use the posterior probabilities to select the best tree.

The structure of a given tree $T$ is composed of the terminal nodes $\tilde{T}$ and parameters $\Theta = (\theta_1, \ldots, \theta_{|\tilde{T}|})$ at the terminal nodes. Let $y_{ti}$ denote the $i$th observation in the $t$th terminal node, $t = 1, \ldots, |\tilde{T}|$, $i = 1, \ldots, n_t$. Then, the data can be represented by $Y \equiv (Y_1, \ldots, Y_{|\tilde{T}|})'$ where $Y_t \equiv (y_{t1}, \ldots, y_{tn_t})'$.

For a classification tree, it is typically assumed that, conditionally on $(\Theta, T)$, $y$ values within the terminal node are independent and identical distributed, and $y$ values across terminal nodes are independent. In addition, since $y_{ti}$ belongs to one of $K$ categories (i.e. $K$ class classification problem), we assume $f(y_{ti}|\theta_t)$ to be a multinomial distribution. Then the likelihood becomes

$$P(Y|\Theta, T) = \prod_{t=1}^{|\tilde{T}|} \prod_{i=1}^{n_t} f(y_{ti}|\theta_t) = \prod_{t=1}^{|\tilde{T}|} p_{t1}^{n_{t1}} \cdots p_{tK}^{n_{tK}},$$

where $\theta_t = (p_{t1}, \ldots, p_{tK})$, $p_{tk} \geq 0$, $\sum_k p_{tk} = 1$ and $n_{tk}$ is the number of observations in node $t$ belonging to the class $k$.

To specify $\pi(T)$ for the structure of $T$, we mimic the prior developed by Chipman et al. (1998). They considered the node splitting probability and the selection probabilities of the split variables. However, we use only the node splitting probability. The reason for this modification is as follows. Our pruning algorithm chooses the optimal tree among the sequence of nested trees already constructed from the growing step. Hence, the random quantity in the tree structure is the size of the tree, but not the split variables. For a given node $t$, the split probability is defined by

$$\Pr(\text{node split}) = a(1 + d_t)^{-\beta},$$

where $d_t$ indicates the depth of a node $t$ and the parameters $a \in (0, 1)$ and $\beta > 0$ control the size and shape of the tree. This means that the larger the $a$ the smaller the probability of the bushed tree. On the other hand, for increasing $\beta$, it is difficult to split at deeper nodes. Using the above splitting probability, the total prior mass for a given tree $T$ is specified by

$$\pi(T) = \prod_{t \in T-\tilde{T}} a(1 + d_t)^{-\beta} \prod_{t \in \tilde{T}} (1 - a(1 + d_t)^{-\beta}).$$

For a prior of parameter $\pi(\Theta|T)$, it is natural to assume that a priori $\pi(\Theta|T) = \prod_{t=1}^{|\tilde{T}|} \pi(\theta_t|T)$ and $\pi(\theta_t|T)$ are Dirichlet distributions with parameters $(\gamma_{t1}, \ldots, \gamma_{tK})$ which are conjugate to the multinomial distribution.

Once the likelihood and priors have been specified, we can calculate the posterior probability of a tree $T$ given data $Y$ from the relation

$$P(T|Y) \propto P(Y|T)\pi(T),$$
where the marginal likelihood $P(Y|T)$ is

$$P(Y|T) = \prod_{t=1}^{|\tilde{T}|} \left( \frac{\Gamma(\sum_k \gamma_k)}{\Gamma(n_t + \sum_k \gamma_k)} \prod_{k=1}^{K} \frac{\Gamma(n_{tk} + \gamma_k)}{\Gamma(\gamma_k)} \right),$$

(4)

provided $\gamma_{tk} = \gamma_k$ for all $t$.

Now, we present the proposed pruning algorithm using the posterior probabilities (3) through the prior (2) and the marginal likelihood (4). For all subtrees of the initial fully grown tree, we could evaluate the posterior probabilities and then select the maximum a posteriori (MAP) tree. However, the number of all subtrees evaluated is very large, which requires too much computational resource. So, we need a reduced set of subtrees instead of all possible subtrees, and CCP is an elegant algorithm for it.

The following pruning algorithm, called CCP–MAP, combines CCP and the MAP strategy. That is, it first generates a sequence of nested trees using CCP, and then calculates the posteriori probabilities of the nested trees to select the optimal one.

Algorithm 1. MAP-pruning via cost-complexity pruned trees (CCP–MAP)

1. Grow the tree up to its maximum size.
2. Get the sequence of nested pruned trees $\{T_1, \ldots, T_M\}$ using CCP.
3. for $j = 1$ to $M$
   - Compute the posterior probability $P(T_j|data)$ from Eq. (3).
end for
4. Select $T_{MAP} = \arg\max\{P(T_j|data), j = 1, \ldots, M\}$.

4. BUMPing based on MAP-pruning

An important application of the proposed CCP–MAP pruning algorithm is the BUMPing procedure suggested by Tibshirani and Knight (1999). BUMPing is a method to select the best one among the set of trees constructed on bootstrap samples. In the problems where there are many local minima (or local maxima) such as decision trees, BUMPing can help avoiding getting stuck in a poor solution.

Let $L$ be the training sample and $L^b$ be a bootstrap sample obtained from $L$. The standard algorithm of BUMPing is as follows:

1. Let $R_\lambda(T) = R(T) + \lambda|\tilde{T}|$ be the cost-complexity measure for a given tree $T$ where $R$ is the misclassification error.
2. Estimate the complexity parameter $\lambda = \arg\min_\lambda \lambda R_\lambda(T)$ in the usual way from the training sample $L$ using CV.
3. For each bootstrap sample $L^b$, find the best trees $T^b = T_\lambda(L^b)$ for $b = 1, \ldots, B$.
4. Choose the tree $T^* = \arg\min_\lambda R(T^b)$, where $R$ is evaluated from the original training sample $L$.

One problem of the standard BUMPing algorithm is that the optimal complexity parameter $\lambda$ obtained from the original sample $L$ may not be optimal for bootstrap samples. For this reason, BUMPing may yield sub-optimal results. To improve the performance of
BUMPing, we can apply CCP–MAP to the BUMPing algorithm instead of CV. This strategy makes it possible to compare larger number of trees than the usual BUMPing, so that gives more chances of finding the optimal one. The proposed BUMPing algorithm based on CCP–MAP, called BUMP–MAP, is given as follows.

**Algorithm 2. BUMPing based on CCP–MAP (BUMP–MAP)**

1. \( \text{for } b = 1 \text{ to } B \text{ do} \)
   - Make a bootstrap sample \( \mathcal{L}^b \) from \( \mathcal{L} \).
   - Grow tree \( T^{(b)} \) from \( \mathcal{L}^b \).
   - Construct a sequence of pruned trees via CCP: \( \{ T^{(b)}_j, j = 1, \ldots, M_b \} \).
   - Compute \( P(T^{(b)}_j | \mathcal{L}) \) from the training data \( \mathcal{L} \).
2. Choose the tree having the maximum a posteriori probability.

**5. Empirical results**

**5.1. Experimental setup**

In this section, we compare the proposed method with other pruning algorithms by analyzing 10 real data sets listed in Table 1, which are available in UCI machine learning repository (http://www.ics.uci.edu/~mlearn/MLRepository.html). For evaluation of the performance, we first divide the data set randomly into training set (70%) and test set (30%). This random division of the data set is repeated 20 times. All the following results are the summary over the 20 iterations.

For CCP–MAP pruning, we should determine the hyperparameters \( \alpha, \beta \) and \( \gamma \) appropriately. We use a pair \((0.9, 1.0)\) for \((\alpha, \beta)\) that produces the best results empirically. For \( \gamma \), we choose \((1, 1)\) on two-class problems because the corresponding Dirichlet distribution becomes the uniform distribution which is thought to be a noninformative prior. However, in the case of multi-class problems, this choice results in poor accuracies since the total

<table>
<thead>
<tr>
<th>#</th>
<th>Data sets</th>
<th>No. of obs.</th>
<th>No. of classes</th>
<th>No. of input variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Australian</td>
<td>690</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>Breast cancer</td>
<td>683</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>German</td>
<td>1,000</td>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>Glass</td>
<td>214</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>Ionosphere</td>
<td>351</td>
<td>2</td>
<td>34</td>
</tr>
<tr>
<td>6</td>
<td>Iris</td>
<td>150</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>Diabetes</td>
<td>768</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>Sonar</td>
<td>210</td>
<td>2</td>
<td>60</td>
</tr>
<tr>
<td>9</td>
<td>Vehicle</td>
<td>846</td>
<td>4</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>Vowel</td>
<td>999</td>
<td>11</td>
<td>10</td>
</tr>
</tbody>
</table>
The averaged misclassification error rates (%) and standard errors for the four pruning methods

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Pruning methods</th>
<th>CCP–CV</th>
<th>CCP–MAP</th>
<th>CCP–BIC</th>
<th>MEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australian</td>
<td></td>
<td>15.12 (0.51)</td>
<td>15.05 (0.51)</td>
<td>14.98 (0.53)</td>
<td>15.82 (0.56)</td>
</tr>
<tr>
<td>Breast cancer</td>
<td></td>
<td>5.07 (0.28)</td>
<td>5.14 (0.32)</td>
<td>5.02 (0.31)</td>
<td>5.17 (0.32)</td>
</tr>
<tr>
<td>German</td>
<td></td>
<td>27.67 (0.64)</td>
<td>28.70 (0.66)</td>
<td>29.13 (0.61)</td>
<td>30.90 (0.51)</td>
</tr>
<tr>
<td>Glass</td>
<td></td>
<td>34.69 (1.19)</td>
<td>35.23 (1.16)</td>
<td>35.86 (1.12)</td>
<td>34.92 (1.16)</td>
</tr>
<tr>
<td>Ionosphere</td>
<td></td>
<td>10.76 (0.72)</td>
<td>12.67 (0.61)</td>
<td>12.24 (0.65)</td>
<td>12.52 (0.55)</td>
</tr>
<tr>
<td>Iris</td>
<td></td>
<td>6.78 (0.65)</td>
<td>6.56 (0.67)</td>
<td>6.44 (0.60)</td>
<td>6.89 (0.70)</td>
</tr>
<tr>
<td>Diabetes</td>
<td></td>
<td>25.33 (0.40)</td>
<td>25.33 (0.50)</td>
<td>26.24 (0.56)</td>
<td>28.87 (0.60)</td>
</tr>
<tr>
<td>Sonar</td>
<td></td>
<td>26.05 (1.12)</td>
<td>25.08 (1.22)</td>
<td>25.65 (1.01)</td>
<td>25.40 (1.28)</td>
</tr>
<tr>
<td>Vehicle</td>
<td></td>
<td>30.33 (0.77)</td>
<td>30.31 (0.47)</td>
<td>30.04 (0.34)</td>
<td>29.72 (0.39)</td>
</tr>
<tr>
<td>Vowel</td>
<td></td>
<td>35.37 (0.77)</td>
<td>32.64 (0.75)</td>
<td>42.59 (0.92)</td>
<td>33.55 (0.66)</td>
</tr>
</tbody>
</table>

The averaged size of trees and standard errors for the four pruning methods

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Pruning methods</th>
<th>CCP–CV</th>
<th>CCP–MAP</th>
<th>CCP–BIC</th>
<th>MEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australian</td>
<td></td>
<td>12.15 (2.23)</td>
<td>14.30 (0.88)</td>
<td>17.05 (0.65)</td>
<td>34.85 (0.56)</td>
</tr>
<tr>
<td>Breast cancer</td>
<td></td>
<td>5.50 (0.35)</td>
<td>10.75 (0.43)</td>
<td>8.45 (0.25)</td>
<td>12.10 (0.27)</td>
</tr>
<tr>
<td>German</td>
<td></td>
<td>10.10 (1.35)</td>
<td>24.00 (1.82)</td>
<td>44.65 (1.33)</td>
<td>73.70 (0.88)</td>
</tr>
<tr>
<td>Glass</td>
<td></td>
<td>10.65 (0.76)</td>
<td>9.80 (0.40)</td>
<td>6.65 (0.29)</td>
<td>13.10 (0.53)</td>
</tr>
<tr>
<td>Ionosphere</td>
<td></td>
<td>4.60 (0.50)</td>
<td>9.45 (0.30)</td>
<td>8.35 (0.39)</td>
<td>12.15 (0.27)</td>
</tr>
<tr>
<td>Iris</td>
<td></td>
<td>3.35 (0.17)</td>
<td>3.65 (0.15)</td>
<td>3.25 (0.10)</td>
<td>4.90 (0.16)</td>
</tr>
<tr>
<td>Diabetes</td>
<td></td>
<td>8.75 (1.63)</td>
<td>17.45 (1.02)</td>
<td>31.65 (1.17)</td>
<td>55.45 (0.55)</td>
</tr>
<tr>
<td>Sonar</td>
<td></td>
<td>6.65 (0.56)</td>
<td>10.20 (0.33)</td>
<td>9.20 (0.21)</td>
<td>11.55 (0.14)</td>
</tr>
<tr>
<td>Vehicle</td>
<td></td>
<td>27.80 (2.43)</td>
<td>21.40 (1.12)</td>
<td>31.40 (1.18)</td>
<td>57.70 (0.49)</td>
</tr>
<tr>
<td>Vowel</td>
<td></td>
<td>53.50 (2.69)</td>
<td>58.85 (0.74)</td>
<td>36.10 (0.97)</td>
<td>55.65 (0.88)</td>
</tr>
</tbody>
</table>

contribution of the prior to the posterior measured by \( \sum_k \gamma_k \) is too large. Thus we set the total contribution to 2 as is the case for the two-class problems. That is, we set \( \gamma_k = 2/K \) for \( K \)-class problems.

5.2. Comparison of four pruning methods

We compare the four pruning methods—CCP–CV, CCP–MAP, CCP–BIC and MEP. Tables 2 and 3 summarize the averaged generalization errors and sizes of the final trees for 20 randomly partitioned data sets.

CCP–CV and CCP–MAP yield similar generalization errors. For tree sizes, the standard errors of CCP–CV are consistently larger than those of CCP–MAP, though the averaged tree sizes of CCP–CV tend to be smaller. This implies that CCP–MAP gives more stable results. We believe that the instability in CCP–CV is mainly due to the instability of CV.
Table 4
The averaged tree sizes and error rates by the two BUMPing procedures

<table>
<thead>
<tr>
<th>Data set</th>
<th>Tree sizes (s.e)</th>
<th>Average errors (s.e)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BUMP–CV</td>
<td>BUMP–MAP</td>
</tr>
<tr>
<td>Australian</td>
<td>7.55 (1.00)</td>
<td>14.95 (0.74)</td>
</tr>
<tr>
<td>Breast cancer</td>
<td>5.35 (0.41)</td>
<td>6.60 (0.21)</td>
</tr>
<tr>
<td>German</td>
<td>17.30 (3.02)</td>
<td>28.45 (0.82)</td>
</tr>
<tr>
<td>Glass</td>
<td>9.85 (0.70)</td>
<td>7.95 (0.38)</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>5.25 (0.36)</td>
<td>7.15 (0.22)</td>
</tr>
<tr>
<td>Iris</td>
<td>3.35 (0.15)</td>
<td>3.00 (0.00)</td>
</tr>
<tr>
<td>Diabetes</td>
<td>14.35 (2.05)</td>
<td>22.70 (0.83)</td>
</tr>
<tr>
<td>Sonar</td>
<td>5.60 (0.41)</td>
<td>7.25 (0.27)</td>
</tr>
<tr>
<td>Vehicle</td>
<td>26.60 (1.13)</td>
<td>18.75 (0.53)</td>
</tr>
<tr>
<td>Vowel</td>
<td>48.30 (1.70)</td>
<td>27.40 (0.48)</td>
</tr>
</tbody>
</table>

MEP tends to yield larger trees. In particular, for two data sets (German and Diabetes), the final trees of MEP are unnecessarily large. It seems that MEP could overfit the training data (i.e. lower accuracies). The results of BIC are data dependent. For German and Diabetes, the tree sizes are too large that the accuracies are worse than those of CCP–CV and CCP–MAP. On the other hand, for Vowel, the resulting tree size is too small to give the lowest accuracy.

5.3. Comparisons of the two bumping methods

We also compare the performance of the BUMPing procedures based on the CV and MAP. From Table 4, we can see that BUMP–MAP improves significantly the standard BUMPing (BUMP–CV) in most cases (8 out of 10 data sets). For the two data sets Vehicle and Vowel where the accuracies of BUMP–MAP are lower than those of BUMP–CV, the tree sizes of BUMP–MAP are too small compared to BUMP–CV. However, note that the MAP pruning algorithm can control the sizes of trees by controlling the hyperparameters (i.e. $\alpha$ and $\beta$ in Eq. (2)). By doing so, we can improve the performance of BUMP–MAP for these two data sets.

6. Discussions

BUMP–MAP can be compared with Bayesian CART proposed by Chipman et al. (1998). The common feature of the two algorithms is to search the maximum a posteriori tree. The difference is that the candidate trees are constructed by the MCMC algorithm in Bayesian CART while BUMP–MAP generates them using bootstrap samples and applying the algorithm of CCP. One disadvantage of Bayesian CART is that the MCMC algorithm requires too much computation so that it cannot be applied to large data sets easily. It is clear that the proposed method needs much less computation.
Instead of choosing the best one from many trees in BUMP–MAP, we can combine all trees into one final model similarly to bagging (Breiman, 1996). While all trees are averaged out in bagging, we can use the weighted average of trees with weights proportional to their posterior probabilities. That is,

\[ \tilde{T}(x) = \sum_{j=1}^{\tilde{\omega}} w_j T_j(x), \]

where \( w_j = (P(T_j|data)) / (\sum_l P(T_l|data)) \). This approach can be considered as an approximation of Bayesian model averaging. We will pursue this idea in future.

Acknowledgements

The authors would like to thank the co-editor and anonymous referee for their helpful comments. This research was supported in part by US Air Force Research Grant F62562-03-P-0658 and in part by KOSEF through Statistical Research Center for Complex Systems at Seoul National University, Korea.

References