Statistics and Its Interface Volume 1 (2008) 155–167

Partially Bayesian variable selection in classification trees[*]

Douglas A. Noe and Xuming He

Tree-structured models for classification may be split into two broad categories: those that are completely data-driven and those that allow some direct user interaction during model construction. Classifiers such as CART [3] and QUEST [11] are members of the first category. In those data-driven algorithms, all predictor variables compete equally for a particular classification task. However, in many cases a subject-area expert is likely to have some qualitative notion about their relative importance. Interactive algorithms such as RTREE [17] address this issue by allowing users to select variables at various stages of tree construction. In this paper, we introduce a more formal partially Bayesian procedure for dynamically incorporating qualitative expert opinions in the construction of classification trees.

An algorithm that dynamically incorporates expert opinion in this way has two potential advantages, each improving with the quality of the expert. First, by de-emphasizing certain subsets of variables during the estimation process, machine-based computational activity can be reduced. Second, by giving an expert's preferred variables priority, we reduce the chance that a spurious variable will appear in the model. Hence, our resulting models are potentially more interpretable and less unstable than those generated by purely data-driven algorithms.

Keywords and phrases: Feature selection, Expert opinion, Supervised learning.

1. INTRODUCTION

Given a data set for a particular application, a researcher will typically build a statistical model with one or both of the following objectives in mind: (1) to use information from this data to make useful predictions about future observations, and (2) to gain some insights into the underlying structure of the data. Tree-based classification models, including CART [3], QUEST [11], C4.5 [15, 16], and many others [4, 7–10, 12] are attractive because of their potential to blend these two objectives quite effectively.

One potential drawback of these older tree-based classification models, however, is that they are completely data-driven. In other words, the resulting tree model is completely determined from the input data with no internal mechanism to indicate the types of results that users might consider "reasonable" or "more desirable" in a particular application context. More recent programs, including RTREE [17] address this problem by allowing direct user involvement in the construction of the tree. Humans may interact with these programs by selecting particular splitting variables at each node as the tree is being constructed.

In this paper, we propose a more formal partially Bayesian framework for dynamically involving qualitative expert opinion in the construction of a classification tree, thereby allowing the user's preferences to compete with the information contained in the data. In pursuit of this goal, we are also mindful of two additional objectives. First, we seek to avoid the variable selection bias inherent in many popular algorithms (including CART). Also, we aim to achieve computational efficiency gains over CART for a large class of problems.

1.1 Framing the problem

To meet these goals, we should more clearly define "expert opinion." Given a particular classification problem, subject-area experts will likely have at least a qualitative notion about which variables are most important to the classification scheme under investigation.

To illustrate, consider Fisher's [5] famous iris classification problem. Based on a learning sample of 150 irises, we wish to predict the categorical response $Y \in \{\text{setosa, versicolor, virginica}\}$ from the four physical measurements petal width (X_1), petal length (X_2), sepal width (X_3), and sepal length (X_4). The tree in Figure 1 represents a common fit based on a CART-like algorithm.

The flowers are first separated by their petal lengths, with those smaller than 2.45 centimeters being correctly declared setosa irises. The remaining 100 flowers are then separated by their petal widths. In this case five virginicas with petal widths under 1.75 centimeters are incorrectly classified as versicolors, and one versicolor with a larger petal width is incorrectly classified as a virginica iris. However, the apparent misclassification rate of only 4% is quite reasonable.

Notice that in this problem, although all four variables were equally considered by the classification algorithm, only

[*] This material is based upon work supported by the National Science Foundation Award DMS-0604229. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the NSF.

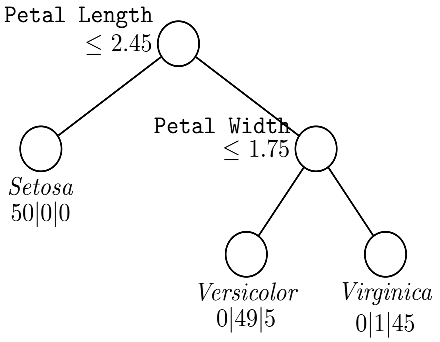

Figure 1. A CART-Based Classification Tree for Fisher's [5] Iris Data. The Three Numbers Present at Each Terminal Node Indicate the Number of Setosa, Versicolor, and Virginica Cases, Respectively, Present in the Node.

two were eventually useful: petal length and petal width. In some sense, the sepal width and sepal length information was not even needed, because the same model could have been generated without these variables.[1]

Quite possibly, a botanist with experience studying iris characteristics might have already suspected that the petal characteristics are more useful than the sepal characteristics in distinguishing between the three species. Throughout our discussion, "expert opinion" will refer to this *qualitative* notion of the relative importance of the available variables to the classification problem at hand.

Our aim is to develop a classification tree algorithm that dynamically incorporates this expert opinion. Our approach is to implement a partially Bayesian variable selection process at each split in the tree, thereby giving full consideration to variables the expert thinks are most important, and lesser consideration to variables the expert thinks are less important to the classification problem.

An algorithm that dynamically incorporates expert opinion in this way has two potential advantages over other single-tree classifiers, each improving with the quality of the expert. First, by de-emphasizing certain subsets of variables during the estimation process, we may reduce the required machine-based computational activity. Moreover, by giving an expert's preferred variables priority, we reduce the chance that a spurious variable will appear in the model. Hence, our resulting models are potentially more interpretable and less unstable than those generated by purely data-driven algorithms.

We shall develop our partially Bayesian classification tree algorithm in Sections 2 and 3. Section 4 explores the properties of our algorithm using a spam classification example. Finally Section 5 concludes with some discussion.

2. PARTIALLY BAYESIAN VARIABLE SELECTION

Our new classification tree algorithm is loosely based on the QUEST method developed by Loh and Shih [11]. QUEST, which stands for Quick, Unbiased, Efficient, Statistical Tree, is particularly well-suited for our purposes because it separates the tasks of variable selection and split value selection. However, QUEST is completely data-driven; expert opinion is not considered in the variable selection process.

The primary innovation in our new tree-growing algorithm lies in the variable selection procedure. As in the QUEST algorithm, variable selection and split value selection are handled separately. Unlike QUEST, our procedure provides expert opinion a dynamic influence on the selection of the splitting variable at each node of the tree.

We shall build the general algorithm over the next two sections. In the current section, we assume that each variable is assigned a *unique* importance rank by the expert user. We refer to this as the *unblocked* algorithm. In Section 3 we relax this assumption, allowing the expert to provide equal importance ranks within blocks of variables.

2.1 Expert variable ranks

Assume that at a particular node we have K potential splitting variables, $X_1, \ldots, X_K$. Suppose further that the expert provides unique "importance ranks" $\mathbf{r^0} = (r_1^0, \ldots, r_K^0)$ corresponding to each of the splitting variables.[2] Here we assume that ranks decrease with the importance of the variable.

This vector of expert variable ranks serves as a parameter describing a prior distribution π of candidate ranks $\mathbf{r} \in \Theta(\mathbf{r^0})$,

$$\pi(\mathbf{r}) \propto \begin{cases} 1 + \tau(\mathbf{r}, \mathbf{r^0}), & \text{if } \tau(\mathbf{r}, \mathbf{r^0}) \geq \tau^\star; \\ \varepsilon_1, & \text{otherwise,} \end{cases} \tag{1}$$

where $\Theta(\mathbf{r^0})$ represents the set of all permutations of the vector $\mathbf{r^0}$, $\tau(\mathbf{x}, \mathbf{y})$ denotes Kendall's τ_b rank correlation between the vectors $\mathbf{x}$ and $\mathbf{y}$, and $\tau^\star \in (-1 + \varepsilon_1, 1]$ is a user-defined threshold value designating the assumed quality of the expert. The parameter ε_1 is a small positive constant used to ensure that all permutations of $\mathbf{r^0}$ receive at least some positive prior probability; the support of $\pi(\cdot)$ should be the entire set $\Theta(\mathbf{r^0})$.

The prior distribution tells us that as the candidate variable ranks, $\mathbf{r}$, more closely agree with the expert-provided ranks, $\mathbf{r^0}$, we consider them more likely to be descriptive of the true relative importance of the predictor variables. We use rank correlation to measure this level of agreement between $\mathbf{r}$ and $\mathbf{r^0}$. However, because probability distributions are non-negatively-valued, we shift the correlation scale by one unit from $[-1, 1]$ to $[0, 2]$ to arrive at equation (1).

[1] We do note that ensemble methods such as random forests [2], bagging [1], and boosting [6], might capture additional information from these otherwise unused variables. However, we focus our attention in this paper on single-tree models for their simplicity and ease of interpretation.

[2] Note that since each variable has a unique rank, the vector $\mathbf{r^0}$ is a permutation of the vector $(1, 2, \ldots, K)$.

The threshold value $\tau^\star$ can be interpreted as a designation of the assumed "certainty" of our expert. A candidate rank vector $\mathbf{r}$ only attains a significant positive prior distribution probability if it meets this pre-defined threshold of agreement with the expert rank vector $\mathbf{r^0}$. If the expert is very certain about the relative importance of the predictor variables, then only those $\mathbf{r}$ that are very similar to $\mathbf{r^0}$ should be seriously considered as potential representations of the "true" variable importance ranks. In this case, $\tau^\star$ should take a value near 1. On the other hand, if the expert is less certain, as might be the case in a purely exploratory study, then very little restriction should be placed on $\mathbf{r}$. In this case, $\tau^\star$ should take a value close to -1.

Running example: Fisher's iris classification problem

Suppose an expert approaches us to consult on Fisher's iris classification problem introduced in Section 1.1. Our expert declares that petal characteristics are preferred to sepal characteristics for proper iris classification, and furthermore, that lengths are preferred to widths. Thus, this expert would rank the variables from most important to least important as follows: $\{X_2, X_1, X_4, X_3\}$, resulting in $\mathbf{r^0} = (2, 1, 4, 3)$.

To form the prior distribution π of variable importance ranks, we need to select our threshold value $\tau^\star$. Here we shall select $\tau^\star = .25$, indicating that we have a moderately certain expert. In this example, ε_1 shall remain unspecified; it suffices to consider ε_1 a negligible positive constant. In this case, our sample space $\Theta(\mathbf{r^0})$ consists of the set of all 24 permutations of $\mathbf{r^0}$.

Nine of the 24 permutations are similar enough to the expert's variable importance rank vector $\mathbf{r^0}$ to receive non-negligible consideration in the prior distribution π. That is, these nine $\mathbf{r} \in \Theta(\mathbf{r^0})$ satisfy the condition $\tau(\mathbf{r}, \mathbf{r^0}) \geq .25$. So, our prior distribution of variable ranks $\mathbf{r} \in \Theta(\mathbf{r^0})$ is:

$$\pi(\mathbf{r}) \propto \begin{cases} 2, & \text{if } \mathbf{r} = (2,1,4,3); \\ 5/3, & \text{if } \mathbf{r} \in \{(1,2,4,3),(3,1,4,2),(2,1,3,4)\}; \\ 4/3, & \text{if } \mathbf{r} \in \{(3,2,4,1),(1,3,4,2),(3,1,2,4), \\ & \qquad (1,2,3,4),(4,1,3,2)\}; \\ \varepsilon_1, & \text{otherwise.} \end{cases}$$

2.2 Posterior variable rank distribution

Following the Bayesian methodology, once our prior distribution of variable importance ranks is defined, we need to update the expert-provided ranks based on observed data. For this purpose, we borrow the QUEST method of using p-values derived from appropriate statistical tests as a means of ranking the predictor variables [11].

For each ordered or continuous variable, we conduct an ANOVA F-test for equality of group means. Intuitively, variables with greater departures from the null hypothesis (mean equality across classes) should be more important in classifying cases at a given node. Similarly, for each unordered categorical variable, we conduct a chi-square test of independence across categories. Again, those variables with greater departures from the null hypothesis (identical categorical distributions across classes) should be more important in classifying cases. Since the p-value measures the departure from the null hypothesis in each case, we use these p-values as a proxy for the importance of each variable. Note that an inverse relationship exists between the qualitative importance of a predictor and the p-value associated with its statistical test; as a variable becomes more important, its p-value tends to decrease.

2.2.1. Working distribution of the p-value ranks

To formulate the posterior distribution of the variable importance ranks (given the p-values), we need to consider the distribution of the data (p-values) given the true variable importance ranks. Because the real distribution would depend upon the unknown model in a way that is infeasible to quantify, we instead consider a working distribution based on the *ranks* of the p-values.

Intuitively, those vectors of p-values most closely resembling the true variable importance ranks should be those most likely generated by the data. Using logic similar to the specification of the prior importance rank distribution, we specify the working distribution of the vector of p-values, $\mathbf{p}$, given the true importance ranks, $\mathbf{r}$, to be

$$f(\mathbf{p}|\mathbf{r}) \propto \begin{cases} 1 + \psi(\mathbf{p}, \mathbf{r}), & \text{if } \psi(\mathbf{p}, \mathbf{r}) \geq \psi^\star; \\ \varepsilon_2, & \text{otherwise,} \end{cases} \tag{2}$$

where $\psi(\mathbf{x}, \mathbf{y})$ is some measure of the correlation between the vectors $\mathbf{x}$ and $\mathbf{y}$, and $\psi^\star \in (-1 + \varepsilon_2, 1]$ is a user-defined threshold value regulating the variance of this p-value distribution. The parameter ε_2 is a small positive constant that ensures that all configurations of p-value ranks are assigned a positive probability.

Again, the intuition behind this conditional distribution is quite straightforward. Given repeated sampling from a population with "true" variable importance ranks, $\mathbf{r}$, we would expect most of our samples to be in relatively close agreement with $\mathbf{r}$. Potential sample ranks, $\mathbf{p}$, that are in stark disagreement with the true variable importance ranks, $\mathbf{r}$, should be quite rare.

The threshold parameter $\psi^\star$ allows the user to regulate the spread of the distribution of observable vectors of p-value ranks. If the set of observable vectors is tightly restricted near the true importance rank vector $\mathbf{r}$, then $\psi^\star$ should take a value near 1. On the other hand, if this set is less restricted, $\psi^\star$ should take a value near -1.

Because the working model considers p-value *ranks*, a natural choice for the correlation measure $\psi(\cdot)$ is again Kendall's τ_b. Unfortunately, this quantity may be undefined if either (1) all of the given variable importance ranks are equal, or (2) the p-values are all so near zero that they are indistinguishable. In either case, we shall define $\psi(\mathbf{p}, \mathbf{r}) = -1$. In this way, no variable importance rank vector $\mathbf{r}$ would meet the $\psi^\star$ threshold for agreement with the data, so our working distribution for the p-values would have a discrete uniform distribution.

Table 1. Summary of Predictor Variable Means (in Centimeters) by Species for Fisher's [5] Iris Data. Also Included Are the ANOVA-Based p-Values for Testing Mean Equality Across Species

Species	Petal Width (X_1)	Petal Length (X_2)	Sepal Width (X_3)	Sepal Length (X_4)
Setosa	0.246	1.462	3.428	5.006
Versicolor	1.326	4.260	2.770	5.936
Virginica	2.026	5.552	2.974	6.588
p-value	4×10^{-85}	3×10^{-91}	5×10^{-17}	2×10^{-31}

2.2.2. Deriving the posterior rank distribution

Now that we have specified the prior distribution, $\pi(\mathbf{r})$, of variable importance ranks and the conditional distribution, $f(\mathbf{p}|\mathbf{r})$, of the data given the "true" variable importance ranks, we have enough information to formulate the posterior distribution of the variable importance ranks given observed data. Using Bayes' Theorem, we derive:

$$\pi(\mathbf{r}|\mathbf{p}) \propto f(\mathbf{p}|\mathbf{r})\pi(\mathbf{r}). \tag{3}$$

Substituting from equations (1) and (2), we obtain:

$$\pi(\mathbf{r}|\mathbf{p}) \propto \begin{cases} (1+\tau(\mathbf{r},\mathbf{r^0}))(1+\psi(\mathbf{p},\mathbf{r})), \\ \qquad \text{if } \tau(\mathbf{r},\mathbf{r^0}) \geq \tau^\star \text{ and } \psi(\mathbf{p},\mathbf{r}) \geq \psi^\star; \\ \varepsilon_1(1+\psi(\mathbf{p},\mathbf{r})), \\ \qquad \text{if } \tau(\mathbf{r},\mathbf{r^0}) < \tau^\star \text{ and } \psi(\mathbf{p},\mathbf{r}) \geq \psi^\star; \\ \varepsilon_2(1+\tau(\mathbf{r},\mathbf{r^0})), \\ \qquad \text{if } \tau(\mathbf{r},\mathbf{r^0}) \geq \tau^\star \text{ and } \psi(\mathbf{p},\mathbf{r}) < \psi^\star; \\ \varepsilon_1\varepsilon_2, \quad \text{if } \tau(\mathbf{r},\mathbf{r^0}) < \tau^\star \text{ and } \psi(\mathbf{p},\mathbf{r}) < \psi^\star. \end{cases} \tag{4}$$

For computational convenience, since the last three cases are all of negligible magnitude, we may approximate this expression by:

$$\pi(\mathbf{r}|\mathbf{p}) \propto \begin{cases} (1+\tau(\mathbf{r},\mathbf{r^0}))(1+\psi(\mathbf{p},\mathbf{r})), \\ \qquad \text{if } \tau(\mathbf{r},\mathbf{r^0}) \geq \tau^\star \text{ and } \psi(\mathbf{p},\mathbf{r}) \geq \psi^\star; \\ \varepsilon, \quad \text{otherwise,} \end{cases} \tag{5}$$

where $\varepsilon > 0$ is set to a negligible positive constant. In our implementation, the default value of ε is 10^{-6}.

The posterior distribution of the variable importance ranks indicates that a candidate vector, $\mathbf{r}$, of variable importance ranks is deemed most likely to represent the "true" variable importance ranks if it is simultaneously in agreement with both the expert-defined ranks, $\mathbf{r^0}$, and the observed p-value ranks, $\mathbf{p}$. If the expert and the data agree, then the posterior distribution is tightly packed around the original prior variable importance ranks. On the other hand, if the expert and the data are in stark disagreement, then the posterior distribution approaches discrete uniformity.

Running example: Fisher's iris classification problem

To determine the approximated posterior variable importance rank distribution $\pi(\mathbf{r}|\mathbf{p})$ (as expressed in equation (5)) we first must select values for the parameters $\psi^\star$ and ε. In this case, we shall select $\psi^\star = 0$, indicating that we require importance ranks, $\mathbf{r}$, to have non-negative rank association with the observed p-value ranks, $\mathbf{p}$, to receive non-negligible consideration in the posterior distribution. Again, we shall set the parameter ε to its default value, 10^{-6}.

Next, we need to obtain p-values from the appropriate statistical test for each variable. In this case, each of the four predictor variables are continuous, so an ANOVA F-test for mean equality across iris species is used. Table 1 summarizes the average values of each predictor variable by species. (Recall that there are 50 members of each species in the learning sample.) Here we see that each of the variables exhibit significantly different class means by any reasonable standard; our observed p-values are (2×10^{-31}, 5×10^{-17}, 3×10^{-91}, 4×10^{-85}). Since the posterior distribution depends only on the *ranks* of the observed p-values, these data may be represented as $\mathbf{p} = (2, 1, 4, 3)$.

To obtain our final expression for the posterior rank distribution, $\pi(\mathbf{r}|\mathbf{p})$, we need only check the nine vectors $\mathbf{r}$ that met the τ threshold to determine whether they also meet the ψ threshold. In this case, $\mathbf{p}$ and $\mathbf{r^0}$ are equal, i.e., the observed data completely agree with the expert-assigned variable importance ranks. Therefore, since the τ threshold is less than the ψ threshold, all of these nine vectors shall receive non-negligible weight in the posterior rank distribution. Our final expression for $\pi\left[\mathbf{r}|\mathbf{p} = (2, 1, 4, 3)\right]$ is as follows:

$$\begin{aligned} &\pi[\mathbf{r}|\mathbf{p} = (2,1,4,3)] \\ &\quad \propto \begin{cases} 4, & \text{if } \mathbf{r} = (2,1,4,3); \\ 25/9, & \text{if } \mathbf{r} \in \{(1,2,4,3),(3,1,4,2),(2,1,3,4)\}; \\ 16/9, & \text{if } \mathbf{r} \in \{(3,2,4,1),(1,3,4,2),(3,1,2,4), \\ & \qquad (1,2,3,4),(4,1,3,2)\}; \\ 10^{-6}, & \text{otherwise.} \end{cases} \end{aligned} \tag{6}$$

2.3 Splitting a node

Following our partially Bayesian framework, we shall use the posterior mean, $E[\mathbf{R}|\mathbf{p}]$, to prioritize the predictor variables. In our first major departure from QUEST, we will *not* automatically split on the top-priority variable. Instead, we sequentially audition each variable in order of posterior importance (from lowest posterior mean rank to highest posterior mean rank). If a split on an "auditioning" variable meets or exceeds a pre-designated improvement threshold, then the variable is selected, and the split is enforced. However, if the auditioning variable fails to meet the minimum improvement threshold, the next-priority variable (the variable with the next-lowest posterior mean rank) is auditioned.

The minimum improvement threshold at each node is regulated by a global *minimum relative improvement* parameter. The user may specify a relative improvement threshold $r \in (0, 1)$. A split on node t meets this threshold if it represents a $100r\%$ improvement on the original impurity level[3] of the node. Letting i_t represent the impurity level at node t prior to a split, we can convert r to a maximum post-split impurity threshold $i_t^\star$ as follows:

$$i_t^\star = (1 - r)i_t.$$

In our current program, the default relative improvement threshold is set to 10%.

Implicit in this procedure is a potential stopping rule. If no variable produces a sufficient improvement at node t, then node t could be declared a terminal node. On the other hand, if we wish to force a split, we would select the highest-priority variable that achieves the greatest reduction in impurity.

Algorithm 2.1 details the split variable (and split point) selection procedure in the new algorithm.

Algorithm 2.1 (Variable Selection under the New Algorithm)**.** *Let node t be given. Suppose $E[\mathbf{R}|\mathbf{p}]$ is the posterior mean vector of variable importance ranks, $\mathbf{R}$, given the observed ranks, $\mathbf{p}$. Let $X_{(k)}$ denote the variable having the k^{th}-*lowest *posterior mean.*

For $k = 1, \ldots, K$,

1. *Attempt to split node t using variable $X_{(k)}$.*
2. *If this split satisfies the minimum improvement requirement, then split on $X_{(k)}$, and break the loop.*

If the loop is completed and no variable satisfies the minimum improvement requirement, then:

- *if the user has elected to implement stopping rules, declare node t a terminal node;*
- *otherwise, split on the variable $X_{(i)}$ that achieves the greatest improvement in impurity. If two or more variables fit this criterion, select the variable of highest priority among them.*

Running example: Fisher's iris classification problem

Having derived (6), the posterior distribution of the variable importance ranks given the sample data, we now wish to select a splitting variable. To make our selection, we need to calculate the posterior mean of the variable importance ranks, $E\left[\mathbf{R}|\mathbf{p} = (2, 1, 4, 3)\right]$. Since the distribution is discrete, this is a relatively straightforward exercise; our conditional mean is approximately proportional to $(46.0, 31.1, 75.0, 60.1)$, or $(2, 1, 4, 3)$ in ranks.

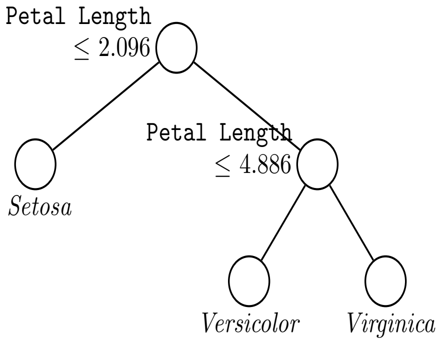

Figure 2. Classification Tree for Fisher's [5] Iris Data Using Our Proposed Algorithm and a Minimum Relative Improvement Threshold of 10%. Labels at Each Terminal Node Indicated the Classification Species.

We prioritize these variables for splitting in order of their posterior mean rank. Therefore, X_2 (petal length) is our first choice, followed by X_1 (petal width), X_4 (sepal length), and X_3 (sepal width).

Following Algorithm 2.1 using the Gini index as our measure of impurity, we first attempt to split the data on X_1 (petal width), producing a splitting value of $X_1 = 2.095$ centimeters. Because the data consist of 50 setosa, 50 versicolor, and 50 virginica irises, the starting Gini index is $1 - (50/150)(50/150)(50/150) = 2/3$. After the split, the 50 setosa irises are isolated in their own node (with a Gini index of 0), and the 50 versicolor and 50 virginica irises are grouped into a second node (having a Gini index of 1/2). The weighted Gini value resulting from the split is therefore $(50*0+100*.5)/150 = 1/3$. Hence, this split results in a 50% improvement in impurity. Since this meets any reasonable threshold, we accept the split. Had an acceptable split *not* been achieved, we would have tried to split on the remaining variables in order of priority until either an acceptable split was achieved or all the predictors were exhausted without reaching the minimum improvement threshold.

In this example, we select the tree size based on the 1-SE method of overfitting and pruning described in [3]. Figure 2 depicts the 1-SE tree generated by our partially Bayesian algorithm with unblocked variable importance ranks and a minimum relative improvement parameter of 10%.

The results from this new method are very similar to those of CART and QUEST, with one exception. Our trees split *twice* on petal lengths before other variables are considered; CART and QUEST split on petal length only once before splitting on petal width. While this may seem a bit strange at first, the partitioned sample space (see Figure 3) illustrates that our split is quite reasonable; virginica irises tend to have longer petals than versicolor irises, though there is some overlap. The reason our method selected petal length instead of petal width for the second split lies in the use of expert variable importance weights. Our expert

[3] Common impurity measures include the deviance (cross-entropy) and the Gini index. In this paper, we use the Gini index.

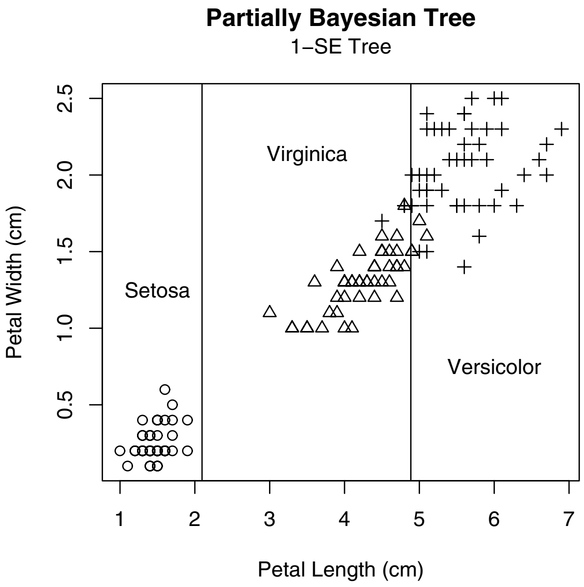

Figure 3. A Partitioned Variable Space for Fisher's [5] Iris Data, Based on Our Proposed Classification Algorithm. Setosa Irises Are Depicted by Circles, Virginica Irises by Pluses, and Versicolor Irises by Triangles. Labels in Each Region Indicate the Model-Predicted Species.

indicated that petal length is more important to the classification of irises than petal width. Therefore, if petal length provides a reasonable split, it *should* be selected in favor of a competing variable that performs slightly better in the learning sample. If our expert's views represent the normative views in his field, then this tree should be more intuitive (and therefore more widely accepted) than either the CART or QUEST trees. Moreover, an expert in this field might consider our tree more generalizable to the problem of classifying out-of-sample irises.

3. RANKING BLOCKS OF VARIABLES

The procedure outlined in Section 2 meets one of our main objectives: expert opinion is dynamically incorporated into the construction of the tree classifier. However, two key problems arise:

1. For problems with large numbers of variables, producing the set of admissible variable importance ranks becomes intractable.
2. Experts cannot be expected to produce unique importance ranks for all variables, especially as the number of variables, K, becomes large. Even for relatively small problems, such a task may be unreasonable.

To address these issues, we propose that the expert provide importance ranks for *blocks* of predictor variables.

This solution makes practical sense; faced with potentially hundreds or more variables, a subject expert could reasonably sort the variables into groups in order of importance. The following sections incorporate such blocking into the methodology developed in Section 2.

3.1 Blocked expert variable ranks

Suppose that at node t we have K potential splitting variables, $X_1, \ldots, X_K$. Suppose that the expert provides a vector of variable importance ranks, $\mathbf{r^0}$, which take on $B \leq K$ unique values. Then, we may view $\mathbf{r^0}$ as the result of a transformation $\eta : \mathbb{N}^B \to \mathbb{N}^K$ defined by $\eta(1, \ldots, B) = \mathbf{r^0}$.

Let $\Omega(\mathbf{r^0})$ denote the set of all *blocked* permutations of $\mathbf{r^0}$. More precisely, if $\Theta(\mathbf{r})$ is the set of all permutations of a vector $\mathbf{r}$, then

$$\Omega(\mathbf{r^0}) = \{\eta(\theta) | \theta \in \Theta(1, \ldots, B)\}.$$

For example, suppose an expert has blocked five predictor variables into three importance groups: X_1 and X_2 are "most important," X_5 is "least important," and the remaining two variables form a group in between. In this case, we have $\mathbf{r^0} = (1, 1, 2, 2, 3)$, so $\eta : \mathbb{N}^3 \to \mathbb{N}^5$ is defined by $\eta(x, y, z) = (x, x, y, y, z)$. The set of all permutations of unique weights is

$$\begin{aligned}\Theta(1,2,3) = \{&(1,2,3),(1,3,2),(2,1,3),(2,3,1),(3,1,2),\\ &(3,2,1)\}.\end{aligned}$$

Therefore, the set $\Omega(\mathbf{r^0})$ of blocked permutations of $\mathbf{r^0}$ is

$$\begin{aligned}\Omega(\mathbf{r^0}) &= \{\eta(1,2,3),\eta(1,3,2),\eta(2,1,3),\eta(2,3,1),\eta(3,1,2),\\ &\qquad \eta(3,2,1)\}\\ &= \{(1,1,2,2,3),(1,1,3,3,2),(2,2,1,1,3),\\ &\qquad (2,2,3,3,1),(3,3,1,1,2),(3,3,2,2,1)\}.\end{aligned}$$

As in the unblocked case, these expert ranks define a prior distribution π of candidate ranks $\mathbf{r} \in \Omega(\mathbf{r^0})$,

$$\pi(\mathbf{r}) \propto \begin{cases} 1 + \tau(\mathbf{r}, \mathbf{r^0}), & \text{if } \tau(\mathbf{r}, \mathbf{r^0}) \geq \tau^\star; \\ \varepsilon_1, & \text{otherwise,} \end{cases} \tag{7}$$

where $\tau(\mathbf{x}, \mathbf{y})$ denotes Kendall's τ_b rank correlation between the vectors $\mathbf{x}$ and $\mathbf{y}$, and $\tau^\star \in (-1 + \varepsilon_1, 1]$ is a user-defined threshold value designating the assumed quality of the expert. As noted in Section 2.2.2, we must extend the definition of Kendall's τ_b to include the case in which all variables are assigned equal importance by the expert. In this case, we shall define $\tau(\mathbf{r}, \mathbf{r^0}) = -1$. As a result, an expert who assigns equal importance to all variables generates an uninformative (discrete uniform) prior distribution.

Running example: Fisher's iris classification problem

Let us revisit the iris classification problem discussed in the prior section. Suppose a second expert approaches us with a different view of the problem. This expert believes that petal characteristics (widths, X_1 and lengths, X_2) are more important than sepal characteristics (widths, X_3, and lengths, X_4) for classifying irises, but opines that because the lengths and widths of each are positively correlated, the particular dimension measured is unimportant. In this way, the expert has provided us two *blocks* of variables with different levels of importance in classification: $\{X_1, X_2\}$ is a more important set than $\{X_3, X_4\}$.

In the notation above, $\mathbf{r^0} = \eta(1,2) = (1,1,2,2)$. The set $\Omega(\mathbf{r^0})$ of all blocked permutations of $\mathbf{r^0}$ is given by

$$\begin{aligned}\Omega(\mathbf{r^0}) &= \{\eta(\theta) : \theta \in \Theta(1,2)\}\\ &= \{\eta(\theta) : \theta \in \{(1,2),(2,1)\}\}\\ &= \{(1,1,2,2),(2,2,1,1)\}\end{aligned}$$

To obtain our prior distribution π of variable importance ranks $\mathbf{r} \in \Omega(\mathbf{r^0})$, we must first declare a τ threshold value designating the certainty of our expert. Taking the same value, $\tau^\star = .25$, as in the unblocked case, we obtain:

$$\pi(\mathbf{r}) \propto \begin{cases} 2, & \text{if } \mathbf{r} = (1,1,2,2);\\ \varepsilon_1, & \text{if } \mathbf{r} = (2,2,1,1).\end{cases}$$

Again, ε_1 is a negligible positive constant, which we shall leave undeclared in this illustration.

Note that in this blocked example, there are only two elements in the sample space (the unblocked case had 24). Clearly, in addition to making life easier on the expert, blocking offers the potential of significant computational savings. This particular issue will be explored in greater detail in later sections.

3.2 Posterior variable rank distribution

The form of the posterior variable importance rank distribution is identical to that from Section 2.2. The only critical change is in the set of candidate variable rank vectors, $\Omega(\mathbf{r^0})$, which consists of the set of *blocked* permutations of the expert-provided importance ranks.

The working conditional distribution of the observed variable ranks, $\mathbf{p}$, given the true (blocked) variable importance ranks, $\mathbf{r} \in \Omega(\mathbf{r^0})$ is

$$f(\mathbf{p}|\mathbf{r}) \propto \begin{cases} 1+\psi(\mathbf{p},\mathbf{r}), & \text{if } \psi(\mathbf{p},\mathbf{r}) \geq \psi^\star;\\ \varepsilon_2, & \text{otherwise,}\end{cases} \tag{8}$$

where $\psi(\mathbf{x},\mathbf{y})$ is some measure of the correlation between the vectors $\mathbf{x}$ and $\mathbf{y}$, and $\psi^\star \in (-1+\varepsilon_2, 1]$ is a user-defined threshold value regulating the variance of the conditional p-value distribution.

Combining equations (7) and (8) as in the prior section, we arrive at our approximate posterior distribution $\pi(\mathbf{r}|\mathbf{p})$ of candidate variable weights $\mathbf{r} \in \Omega(\mathbf{r^0})$:

$$\pi(\mathbf{r}|\mathbf{p}) \propto \begin{cases} (1+\tau(\mathbf{r},\mathbf{r^0}))(1+\psi(\mathbf{p},\mathbf{r})), & \\ \quad \text{if } \tau(\mathbf{r},\mathbf{r^0}) \geq \tau^\star \text{ and } \psi(\mathbf{p},\mathbf{r}) \geq \psi^\star; & \\ \varepsilon, \quad \text{otherwise.} & \end{cases} \tag{9}$$

Running example: Fisher's iris classification problem

Now we seek the blocked posterior variable weight distribution $\pi(\mathbf{w}|\mathbf{p})$ for the iris data. Again, we shall reuse the parameters from the unblocked case, setting $\psi^\star = 0$ and $\varepsilon = 10^{-6}$. Recall from Table 1 that the observed p-values for the iris data are $(4\times10^{-85}, 3\times10^{-91}, 5\times10^{-17}, 2\times10^{-31})$. Again, since we only need the ranks of these p-values, we may represent the data as $\mathbf{p} = (2,1,4,3)$.

To form the posterior variable importance rank distribution, we must first check that the vector satisfying the τ threshold also satisfies the ψ threshold. Calculating, we find that $\psi((2,1,4,3),(1,1,2,2)) = \tau_b((2,1,4,3),(1,1,2,2)) \approx .8165 \geq \tau^\star = 0$. We thus obtain:

$$\pi[\mathbf{r}|\mathbf{p} = (2,1,4,3)] \propto \begin{cases} 3.633, & \text{if } \mathbf{r} = (1,1,2,2);\\ 10^{-6}, & \text{if } \mathbf{r} = (2,2,1,1).\end{cases} \tag{10}$$

3.3 Splitting a node

As in the unblocked case, we shall use the posterior mean, $E[\mathbf{R}|\mathbf{p}]$, to prioritize the predictor variables for splitting. However, by construction, all variables in a particular block, b, will have equivalent posterior mean importance ranks. Therefore, an additional criterion is required to distinguish among variables belonging to the same block. We assign higher within-block priority to variables having smaller p-values.

In addition, we wish to take advantage of the block structure to avoid the computational cost of potentially auditioning *every* variable in a large list. To this end, we only designate one variable per block as a potential splitting variable. In this way, at node t we only have the potential cost of auditioning B variables instead of all K of them. For very large data sets, this will constitute a significant savings.

The detailed procedure is described in Algorithm 3.1.

Algorithm 3.1 (Variable Selection with Blocked Importance Ranks)**.** *Let node t be given. Suppose $E[\mathbf{R}|\mathbf{p}]$ is the blocked posterior mean vector of variable ranks, $\mathbf{R}$, given the observed variable ranks, $\mathbf{p}$. Let (b) represent the variable block having the b^{th}*-lowest *posterior mean rank, and let $X_{(b)_{(1)}}$ represent the variable in block (b) having the smallest associated p-value.*

For $b = 1, \ldots, B$,

1. *Attempt to split node t using variable $X_{(b)_{(1)}}$ as in QUEST.*
2. *If this split satisfies the minimum improvement requirement, then split on $X_{(b)_{(1)}}$, and break the loop.*

If the loop is completed and no variable satisfies the minimum improvement requirement, then:

- *if the user has elected to implement stopping rules, declare node t a terminal node;*
- *otherwise, split on the variable $X_{(i)_{(1)}}$ that achieves the greatest improvement in impurity. If two or more variables fit this criterion, select the variable of highest priority among them.*

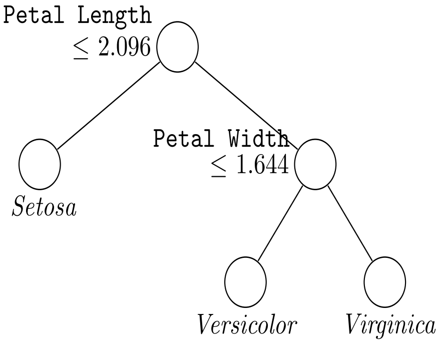

Figure 4. Classification Tree for Fisher's [5] Iris Data Using Our Proposed Blocked Algorithm. Labels at Each Terminal Node Indicate the Classification Species.

Running example: Fisher's iris classification problem

Now we shall conclude the iris classification example. Based on the blocked posterior variable importance rank distribution (10), the posterior mean is easily determined: $E[\mathbf{R}|\mathbf{p} = (2,1,4,3)] \approx (3.633, 3.633, 7.266, 7.266) \equiv (1,1,2,2)$. Therefore, we prioritize groups of variables for splitting as follows: $\{X_1, X_2\}$ (petal characteristics) is our first choice, followed by $\{X_3, X_4\}$ (sepal characteristics).

Following Algorithm 3.1, we use the observed p-values from our ANOVA F-tests of mean equality across species to select from among the petal characteristic variables. Recall from Table 1 that $p_1 = 4 \times 10^{-85}$ while $p_2 = 3 \times 10^{-91}$, so X_2—the petal length variable—is auditioned for splitting. As in the unblocked case, the splitting value chosen is $X_2 = 2.096$ cm, which achieves a relative improvement of 50%.

Again, we overfit and prune, accepting the 1-SE tree as our solution.

Figure 4 depicts the classification tree generated by the blocked procedure. In this model, we note that the data are only split *once* on petal length before petal width enters the model. This result is consistent with our new expert's prior opinion: petal characteristics are more important than sepal characteristics, but petal lengths and petal widths are equally valuable. In this model, since the second split performs better using petal width than petal length, the width variable is selected.

4. EXAMPLE AND PROPERTIES

Having presented the development of the blocked partially Bayesian classification tree algorithm, we now evaluate its performance and explore its properties. In this section, we use a more complex data example—spam filtering—to compare the predictive performance and computational properties of our partially Bayesian tree algorithm and the standard benchmarks: CART and QUEST.[4]

4.1 Spam filtering trees

To examine each model's performance, we consider the task of designing an automatic spam (junk e-mail) filter. The data for this task are publicly available from the UCI Machine Learning Repository[5] [13], and were donated by George Foreman from Hewlett-Packard Laboratories in Palo Alto, California.

The data consist of 58 variables describing 4601 electronic mail messages. The dependent variable is categorical, indicating whether or not a particular message is spam (1 = spam, 0 = not spam). The 57 predictor variables are all ordered, and may be split into three groups:

- *Word frequency variables* indicate the percentage of words in a message that match the specified word. For example, the variable `WFaddress` indicates the frequency of the word "address" as a percentage of all words in an e-mail. There are 48 word frequency variables in the data set.
- *Character frequency variables* indicate the percentage of characters in a message that match the specified character. For example, the variable `CFsemicolon` indicates the frequency of the ";" character as a percentage of all characters in an e-mail. There are nine character frequency variables in the data set.
- *Capital run length variables* provide information about strings of consecutive capital letters in a message. The three capital run length variables are defined as follows:
  - `CRLaverage` is the average size of all strings of consecutive capital letters found in the message.
  - `CRLlongest` is the longest string of consecutive capital letters found in the message.
  - `CRLtotal` is the total number of capital letters in the message.

Of the 4601 messages described in the data set, 1813 (39.4%) are spam.

Table 2 provides summary statistics for each of the predictor variables. The "%=Min" column indicates the percentage of observations that match each variable's minimal value. For each of the word frequency and character frequency variables, the minimum value is 0; for each of the capital run length variables, the minimum value is 1. For example, the word "address" failed to appear in 80.5% of the e-mail messages. On average, "address" comprised 0.213% of the words in each message, and in at least one instance, "address" accounted for 14% of the words in a message.

The contributors note that by the nature of this particular collection of e-mails, certain attributes should be important indicators of non-spam. Most of the non-spam messages

[4] We remain focused on single-tree methods for the basis of comparison. However, we do note that richer ensemble methods tend to outperform single-tree methods in terms of classification accuracy. A simple random forest generated by the authors achieves a 5.43% out-of-bag misclassification rate.

[5] http://www.ics.uci.edu/~mlearn/MLRepository.html

Table 2. Summary Statistics for Predictor Variables in the Spam Classification Data

Variable	%=Min	Mean	Max	Variable	%=Min	Mean	Max
			Word Frequency Variables				
WFaddress	80.5	0.213	14	WForiginal	91.8	0.046	4
WFaddresses	92.7	0.049	4	WFour	62.0	0.312	10
WFall	59.0	0.281	5	WFover	78.3	0.096	6
WFbusiness	79.1	0.143	7	WFparts	98.2	0.013	8
WFconference	95.6	0.032	10	WFpeople	81.5	0.094	6
WFcredit	90.8	0.086	18	WFpm	91.7	0.079	11
WFcs	96.8	0.044	7	WFproject	92.9	0.079	20
WFdata	91.2	0.097	18	WFre	71.5	0.301	21
WFdirect	90.2	0.065	5	WFreceive	84.6	0.060	3
WFedu	88.8	0.180	22	WFremove	82.5	0.114	7
WFemail	77.4	0.185	9	WFreport	92.2	0.059	10
WFfont	97.5	0.121	17	WFtable	98.6	0.005	2
WFfree	73.0	0.249	20	WFtechnology	87.0	0.097	8
WFgeorge	83.0	0.767	33	WFtelnet	93.6	0.065	13
WFhp	76.3	0.550	21	WFwill	49.5	0.542	10
WFhpl	82.4	0.265	17	WFyou	29.9	1.662	19
WFinternet	82.1	0.105	11	WFyour	47.3	0.810	11
WFlab	91.9	0.099	14	WF000	85.2	0.102	5
WFlabs	89.8	0.103	6	WF1999	82.0	0.137	7
WFmail	71.7	0.239	18	WF3d	99.0	0.065	43
WFmake	77.1	0.105	5	WF415	95.3	0.048	5
WFmeeting	92.6	0.132	14	WF650	89.9	0.125	9
WFmoney	84.0	0.094	13	WF85	89.5	0.105	20
WForder	83.2	0.090	5	WF857	95.5	0.047	5
			Character Frequency Variables				
CFbracket	88.5	0.017	4	CFparen	41.0	0.139	10
CFdollar	69.6	0.076	6	CFpound	83.7	0.044	20
CFexclam	50.9	0.269	32	CFsemicolon	82.8	0.039	4
			Capital Run Length Variables				
CRLaverage	7.6	5.192	1103	CRLtotal	0.2	283.289	15841
CRLlongest	7.6	52.173	9989				

were gathered from filed business correspondence. Therefore, the presence of the word "George" (the contributor's name) and the area code "650" suggest legitimacy.

In the rest of this subsection, we shall explore various tree-based solutions to the spam filtering problem. In Section 4.1.1, we present the CART-based exhaustive search solution as a benchmark. Section 4.1.2 follows with the solution from the QUEST algorithm. Finally, Section 4.1.3 explores a solution produced by the partially Bayesian algorithm derived from a knowledgeable expert's prior.

4.1.1. CART tree

In this section we examine the CART 1-SE solution to the spam filtering problem. Figure 5 provides the CART tree, which contains 23 terminal nodes. The CART algorithm selects the frequency of the "$" character as the first splitting variable; the split value is .0555%, which is near the third quartile of its distribution. If the dollar character appears frequently in a particular message, this case is sent down the right branch of the tree, where the frequency of the word "hp" is checked. If the word "hp" comprises at least 0.4% of the message, then the e-mail is classified based on the frequency of the word "remove." Otherwise, the classification is further refined based on the relative frequencies of the words "edu" and "george," as well as the length of the longest string of capital letters.

The above split is common of the structure of this tree: typical indicators of spam are selected, and if one is found, the classification is refined by examining indicators of legitimate messages specific to this user at Hewlett-Packard. Consider the left branch of the tree, where the dollar character is infrequent. The frequency of the word "remove," which is commonly found at the end of bulk e-mail, is checked. If this appears frequently, one might suspect the message is spam. However, the frequency of the word "george" is checked to see if the message might legitimately refer to company business.

In general, the CART 1-SE tree makes sense for this spam classification problem and presents a reasonable estimated misclassification rate (8.63%).

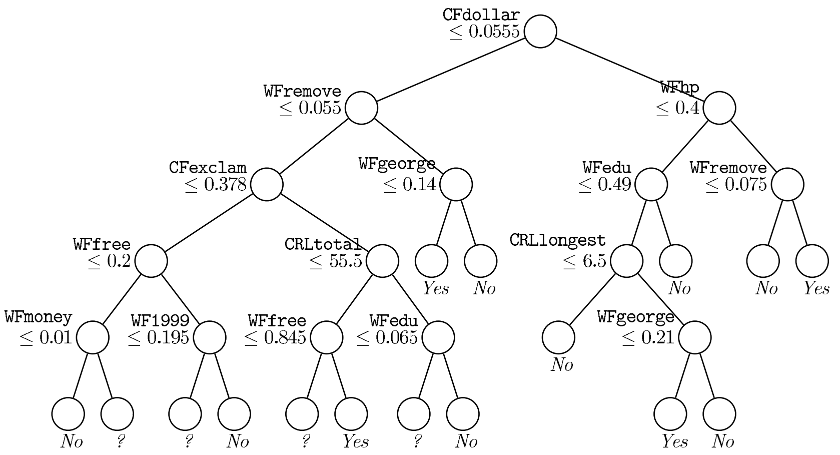

Figure 5. A CART-Based 1-SE Classification Tree for the Spam Classification Data. The CV-Estimated Misclassification Rate Is 8.63%. Nodes Labelled "?" Are Non-Terminal, but Were Snipped for the Sake of Display. The Full Tree Has 23 Terminal Nodes.

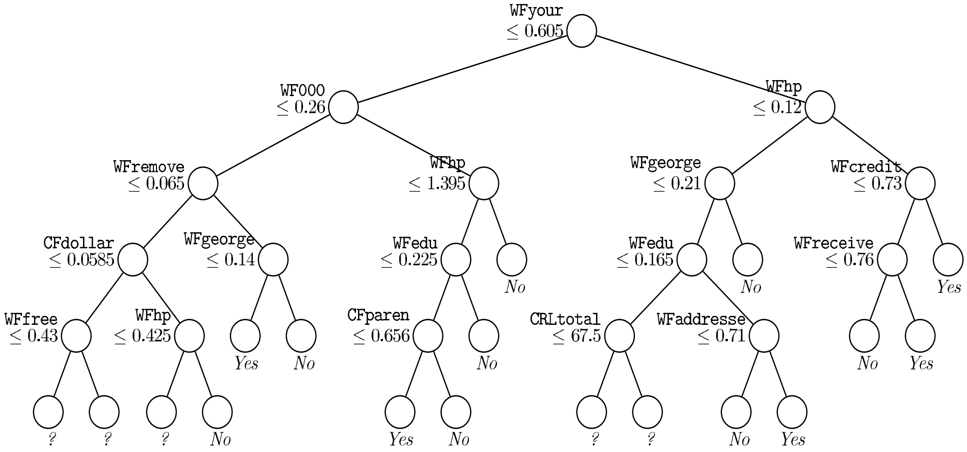

Figure 6. A QUEST-Based 1-SE Classification Tree for the Spam Classification Data. The CV-Estimated Misclassification Rate Is 8.32%. Nodes Labelled "?" Are Non-Terminal, but Were Snipped for the Sake of Display. The Full Tree Has 46 Terminal Nodes.

4.1.2. QUEST tree

The 1-SE classification tree generated by the QUEST algorithm is much larger than that produced by CART, containing twice as many terminal nodes. This much larger tree is somewhat more accurate than the CART example, with an estimated misclassification rate of only 8.32%, compared to 8.63% for the CART tree.

The initial splits in the QUEST tree (see Figure 6) are much different than those in the CART example. The QUEST algorithm splits the root node based on the frequency of the word "your." Messages in which "your" comprises more than 0.605% of all words are sent down the right branch of the tree, where the frequencies of legitimacy indicators (such as "hp" and "george") and spam indicators (such as "credit") are checked. In the left branch of the tree, where the word "your" seldom appears, the frequency of other potential spam indicators (such as "000," "remove" and the dollar) are checked before legitimacy indicators are again examined to refine the classification.

4.1.3. Partially Bayesian tree

We now examine the effect that applying informative qualitative expert knowledge has on the performance of a tree-structured classifier.

In this data set, we have two potentially useful groups of variables: spam indicators and legitimacy indicators. The spam indicators tend to be common to all e-mail users. These include the frequencies of the words "remove" and "free," as well as all of the character frequency variables and character run length variables. The legitimacy indicators are

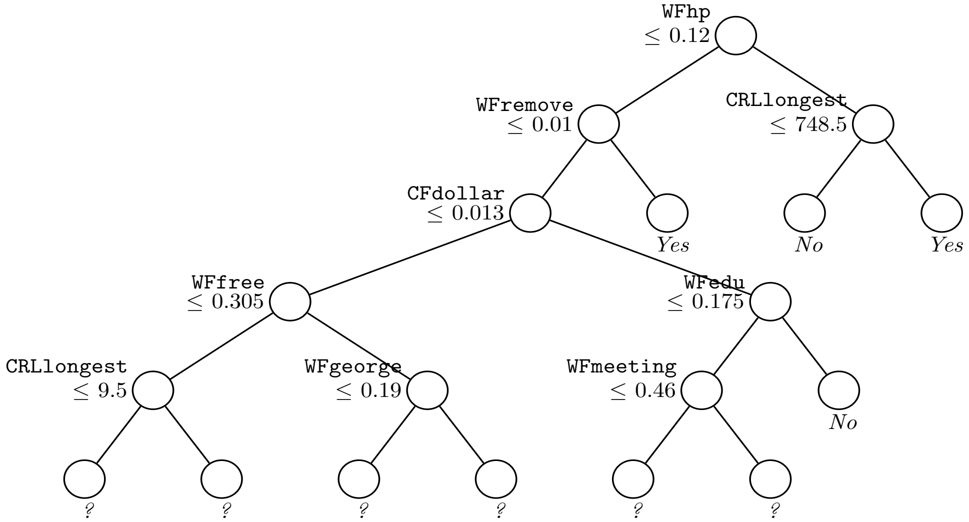

Figure 7. A Partially Bayesian Classification Tree for the Spam Classification Data. In this Case, the Expert Incorporates Situation-Specific Knowledge of Discriminatory Variables. The CV-Estimated Misclassification Rate Is 7.35%. Nodes Labelled "?" Are Non-Terminal, but Were Snipped for the Sake of Display. The Full Tree Has 16 Terminal Nodes.

specific to this individual user's e-mail account. Frequent use of words related to his place of business, his name, or his phone number would suggest (though not prove) that an e-mail is legitimate. Hence, variables indicating the frequencies of "hp," "hpl," "george," and "650" shall be classified as legitimacy indicators. None of the rest of the variables are obviously spam or legitimacy indicators, so these will be grouped into a third category. Hence, we have constructed the following three "blocks" of variables:

- *Typical Spam Variables*—`WFremove`, `WFfree`, all of the character variables and capital run length variables
- *Situation-Specific Legitimacy Indicators*—`WFhp`, `WFhpl`, `WFgeorge`, `WF650`
- *Others*—all other variables

We shall construct our expert opinion to give top priority to the user-specific legitimacy indicators, then next priority to the common spam indicators. The remaining variables will receive the lowest priority. In this way, we assign expert rank 2 to the typical spam variables and expert rank 1 to the situation-specific legitimacy indicator variables. The remaining 42 variables are assigned expert rank 3.

Figure 7 depicts the top levels of the latest tree, which takes advantage of this special knowledge. Here we see a drastic change in the early structure of the tree. Rather than splitting on the frequency of "your," more common spam indicators such as "remove" and "$" are used. Also, the situation-specific legitimacy indicators ("george" and "hp") are more quickly introduced.

We also note that the special knowledge 1-SE tree is significantly less complex yet more accurate than its less well-informed counterparts. The tree contains only 16 terminal nodes and achieves an estimated misclassification rate of 7.35%.

4.2 Computational comparisons

In this section, we assess the differences in machine-based computational requirements for the CART, QUEST, and partially Bayesian tree algorithms using the spam filtering classification example. To compare computational complexity, we shall focus on the number of splitting value calculations required under each model. Using terminology from our discussion of the blocked partially Bayesian procedure of Section 3, we examine the number of variables that are "auditioned" for splitting in each tree.

Recall that to select a splitting variable at a particular node, CART examines every potential variable-split point combination. In essence, this means that *every* variable is auditioned at each node in the tree. On the other end of the spectrum, the QUEST algorithm auditions exactly *one* variable per split. The splitting variable is selected based on p-values from statistical tests prior to computing the split point.

Our partially Bayesian procedure lies between these two extremes. Like QUEST, the partially Bayesian algorithm uses statistical tests; however, these tests merely *prioritize* the splitting variables for auditioning. The variable having the lowest p-value is not always selected. Our procedure will audition up to B variables per split, where B is the number of variable blocks designated by the expert. Therefore, at a particular node the QUEST algorithm requires no more computation than our procedure. An uninformative expert (who places all variables in the same block) produces a tree essentially the same as QUEST in terms of structure and computational load. On the other hand, a partially Bayesian tree directed by unblocked rankings (i.e., a distinct importance rank for each variable) has the *potential* to match the number of auditions required by CART at a particular node.

One should note that although the partially Bayesian algorithm is more computationally expensive at a given node than the QUEST algorithm, in certain cases the process of constructing an entire partially Bayesian *tree* will be more efficient than constructing a QUEST tree. As the quality of the expert increases, splits near the root of a partially Bayesian tree should be more reliable. This condition can result in a smaller final tree than QUEST—potentially requiring less overall computation.

Table 3 provides summary information for each of the spam filtering trees we have considered. To facilitate meaningful comparison, the basic tree construction parameters have been standardized across algorithms. In each case, we employ the overfit-then-prune strategy of estimation. The Gini measure of impurity is used for growing each tree, while the misclassification rate is used to guide pruning. The partially Bayesian tree uses a minimum relative improvement threshold of 10% for blocked variable selection. Each tree is grown until all nodes contain fewer than 1% of the data or splits on larger nodes are no longer possible.

For each overfit tree, the numbers of terminal nodes and required variable auditions are listed. We use 10-fold cross-validation for pruning and estimating misclassification error. The size and estimated error rate of the 1-SE trees are provided for each method. In practice the 1-SE tree is commonly selected as an appropriately "generalizable" solution.

We note that the overfit CART, QUEST, and partially Bayesian trees were of comparable size, ranging from 115 to 132 terminal nodes. However, because the CART algorithm effectively auditions every variable at every node, it required 6,498 variable auditions, which was roughly 20 times as many as the partially Bayesian algorithm, and 50 times as many as QUEST. Moreover, for this particular application, the 1-SE CART tree had the worst estimated misclassification rate of the three, at 8.63%. Clearly, in this case, the computational expense of the CART algorithm did not pay dividends relative to the other two options.

The partially Bayesian algorithm required about 2.5 times as many variable auditions as the QUEST algorithm; however, it produced a 1-SE tree that was about one-third the size of QUEST's. Moreover, the partially Bayesian tree was more accurate than the QUEST tree, with a cross-validation estimated misclassification rate of 7.35% compared with 8.32% for QUEST.

Table 3. CART, QUEST, and Partially Bayesian Computational Comparisons

	Overfit Tree		1-SE Tree	
Method	# TNodes	# Auditions	# TNodes	%Misclass[a]
CART	115	6498	23	8.63
QUEST	132	131	46	8.32
PB Tree	127	325	16	7.35

[a] Misclassification rates for the 0-SE and 1-SE trees are estimated using 10-fold cross-validation.

We see that although there is a small machine-based computational loss relative to QUEST, the partially Bayesian algorithm in the hands of a knowledgeable expert has the potential to produce smaller pruned trees that are more accurate than those of the purely data-driven methods.

We do note, however, that the comparisons here only involve machine-based computations. A pure comparison would also need to include the extent of the unmeasured human resources required to evaluate the relative importance of the predictor variables for blocking. Though the evaluation would be a daunting task in most problems for a very inexperienced user, a knowledgeable expert could perform the task quite quickly based on prior experience, especially with a two-block strategy: well-known important variables versus all others.

4.3 Stability comparisons

In addition to improved predictive performance, one motivating factor for dynamically incorporating qualitative expert opinion in the construction of a tree-based classifier is potential stability. By stability, we mean that the basic structure of the model is unchanged by variations in the sample data. That is, the early variables selected in a tree should not be affected by small changes to the sample data. A model lacking in stability is likely not generalizable to new observations, and it often loses credibility with users.

To compare the stability of our partially Bayesian model to that of CART, we generated 1000 random samples of size 4000 (each taken without replacement from the original data set of size 4601). Hence, each data set was comprised of a random 87% of the original data set. As a crude measure of model stability, we examined the percentage of samples for which each variable appeared as the top splitter in a tree. The results are provided in Table 4.

The results are striking. We see that while the first split in the CART tree is almost equally likely to be based on either the frequency of the dollar sign or the frequency of the exclamation point, the partially Bayesian tree is much more consistent in its first splitting variable. Over 94% of the time, the "hp" word frequency variable is selected, which is consistent with the expert opinion used with the model. While the CART initial split is quite sensitive to the particular set of data, the partially Bayesian split is more resilient. As a result, users in an established application area may be apt to place more trust in an expert-aided partially Bayesian tree.

Table 4. Percentage of Samples for Each Variable to Be Selected as the Top Splitter

	CFdollar	CFexclam	WFhp	WFremove	
CART	54.9	45.1	0.0	0.0	100.0
PB Tree	1.5	0.7	94.1	3.7	100.0

5. DISCUSSION

This paper has identified a potential area of improvement for a large class of classification algorithms. By dynamically incorporating potentially valuable qualitative expert opinion in their feature selection mechanisms, we are led to models that appear more sensible to users without losing fidelity to data. We hypothesize that classification algorithms with this characteristic might produce more interpretable and readily-acceptable models in a manner computationally comparable to or more efficient than related purely data-driven procedures. We tested this hypothesis using the special case of classification trees, and our partially Bayesian algorithm provides a "proof-of-concept" that such a modification is possible.

Computationally, our partially Bayesian classification tree algorithm borrows heavily from the QUEST algorithm. The reasons are simple: since the QUEST algorithm separates the split variable selection process from the task of selecting a splitting value, it avoids the variable selection bias inherent to CART and other simultaneous search algorithms, and it is computationally more efficient.

To retain as much of this computational saving as possible while still benefitting from expert opinion, we ask our expert to arrange the predictor variables into broad importance *blocks*. With only a few blocks, the number of variable auditions is kept at a reasonably low level, and the task is made much easier for the expert.

In spite of our conceptual success, more refinement is needed with the current algorithm. For example, the current algorithm makes use of global variable importance ranks. Of course, one can reasonably argue that after the first few splits in a tree, the usefulness of the expert's original opinion wanes. Constructing hybrid algorithms that make heavy use of the expert's opinion in the early stages of the tree but become more data-driven in the later splits appears promising. In addition, the use of variable importance *weights* in place of variable importance *ranks* could be useful, especially as the partially Bayesian concept is applied to other classification and prediction algorithms. Incorporating a sense of scaled importance could add more richness to these models. Implementation of our concept to more general trees such as that of [17] has not been made, and additional work is clearly called for in making the partially Bayesian approach generally applicable.

Received 16 January 2008

REFERENCES

[1] Breiman, L. (1996). Bagging Predictors. *Machine Learning* **24** 123–140.

[2] Breiman, L. (2001). Random Forests. *Machine Learning* **45** 5–32.

[3] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Chapman & Hall, New York. MR0726392

[4] Chan, K.-Y. and Loh, W.-Y. (2004). LOTUS: An Algorithm for Building Accurate and Comprehensible Logistic Regression Trees. *Journal of Computational and Graphical Statistics*. In press. MR2109054

[5] Fisher, R. A. (1936). The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics* **7** 179–188.

[6] Freund, Y. and Schapire, R. E. (1996). Experiments with a New Boosting Algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, L. Saitta, ed., 148–156.

[7] Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York. MR1851606

[8] Kass, G. V. (1980). An Exploratory Technique for Investigating Large Quantities of Categorical Data. *Applied Statistics* **29** 119–127.

[9] Kim, H. and Loh, W.-Y. (2001). Classification Trees with Unbiased Multiway Splits. *Journal of the American Statistical Association* **96** 589–604. MR1946427

[10] Lim, T.-S., Loh, W.-Y., and Shih, Y.-S. (2000). A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms. *Machine Learning Journal* **40** 203–228.

[11] Loh, W.-Y. and Shih, Y.-S. (1997). Split Selection Methods for Classification Trees. *Statistica Sinica* **7** 815–840. MR1488644

[12] Loh, W.-Y. and Vanichsetakul, N. (1988). Tree-Structured Classification via Generalized Discriminant Analysis (with discussion). *Journal of the American Statistical Association* **83** 715–728. MR0963799

[13] Newman, D. J., Hettich, S., Blake, C. L. and Merz, C. J. (1998). UCI Repository of Machine Learning Databases (`http://www.ics.uci.edu/~mlearn/MLRepository.html`). University of California, Department of Information and Computer Science, Irvine, CA.

[14] Quinlan, J. R. (1979). Discovering Rules by Induction from Large Collections of Examples. In *Expert Systems in the Microelectronic Age*, D. Michie, ed., 168–201. Edinburgh University Press.

[15] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.

[16] Quinlan, J. R. (1996). Improved Use of Continuous Attributes in C4.5. *Journal of Artificial Intelligence Research* **4** 77–90.

[17] Zhang, H. P. (1998). Classification Trees for Multiple Binary Responses. *Journal of the American Statistical Association* **93** 180–193.

Douglas A. Noe
Department of Mathematics and Statistics
Miami University
301 S. Patterson Ave.
Oxford, OH 45056, USA
E-mail address: noeda@muohio.edu

Xuming He
Department of Statistics
University of Illinois at Urbana-Champaign
725 S. Wright St.
Champaign, IL 61820, USA
E-mail address: x-he@uiuc.edu