A Bayesian approach to estimate probabilities in classification trees

Andrés Cano and Andrés R. Masegosa and Serafín Moral
Department of Computer Science and Artificial Intelligence
University of Granada
18071, Granada, Spain

Abstract

Classification or decision trees are one of the most effective methods for supervised classification. In this work, we present a Bayesian approach to induce classification trees based on a Bayesian score splitting criterion and a new Bayesian method to estimate the probability of class membership based on Bayesian model averaging over the rules of the previously induced tree. In an experimental evaluation, we show as our approach reaches the performance of Quinlan's C4.5, one of the most known decision tree inducers, in terms of predictive accuracy and clearly outperforms it in terms of better probability class estimates.

1 Introduction

Decision trees or classification trees (decision trees where is predicted a probability of class membership instead of the class label simply) have been one the most used and better studied predictive models. Several reasons appear examining their wide popularity such as their simplicity and their easy interpretability. At the same time, they are fast and effective as classifiers (Lim et al., 2000) even at very large data sets (Provost and Kolluri, 1999) and there have been available several software packages for learning classification trees (CART (Breiman et al., 1984) and C4.5 (Quinlan, 1993)).

One of the main problems of classification trees is the poor estimates of class probabilities they usually produce (Breiman, 1996a; Pazzani et al., 1994). In many situations, a good estimate of the probability class is a strong requirement, specially, when it is needed a ranking of the samples by the class they belong to. E.g., most of the web search engines ranks the web pages based on their probability of being relevant for a given query.

Let us see an example of the problem to assign class probabilities in a classification tree. Suppose we have induced a classification tree for a two-class classification problem and we find that a leaf node defines a subset of 3 samples, all of them belonging to the positive class. Maximum likelihood estimate shall assign a probability of 1.0. The question that arises now if there is enough evidence with 3 samples to assess such a strong statement.

In (Provost and Domingos, 2003) a survey study of different methods for better probability of class estimates was carried out based on C4.5. They compare three different methods: Laplace estimate, C4.5 pruning (Quinlan, 1993) and, specially, bagging (Breiman, 1996b). They conclude with a positive evidence in favor of Laplace and Bagging, but they do not find a definitive conclusion for pruning.

In this work, we present a Bayesian approach to induce classification trees with the aim to maintain the predictive accuracy of one of the state-of-the-art classification tree inducers $J48$ (an advanced version of Quinlan's C4.5) and produce significative improvements in the estimates of probability class beyond the use of Laplace correction or a post-pruning process. We conduct an experimental study over 27 UCI databases to evaluate our Bayesian approach.

The rest of the paper is organized as follows. In Section 2 we introduce the necessary notations and we briefly explain classification trees and C4.5 tree inducers (Quinlan, 1993). After that, in Section 3 we describe our Bayesian approach to induce classification trees. The experimental evaluation and the results are shown in Section 4. Finally, Section 5 is devoted to the final conclusions and future works.

2 Previous Knowledge

In a classification problem, we have a target or dependent variable C with k cases $(c_1, ..., c_k)$, $|C| = k$, and a set of predictive or independent variables $\mathbf{X} = (X_1, ..., X_n)$. The goal is to induce a probability function for every unseen sample $\mathbf{x}$ to a probability distribution of C: $(\mathcal{L}(\mathbf{X}) \to \mathcal{P}(C))$. This function is represented as a posterior probability of C given a sample $\mathbf{x}$: $P(C|\mathbf{x}) = P(C|X_1 = x_1, ..., X_n = x_n)$.

This posterior probability has to be inferred from a limited set of samples of the joint distribution $(\mathbf{X}, C)$, the learning data $\mathcal{D}$[1].

2.1 Classification Trees

A classification tree, $\mathcal{T}$ is an efficient representation of this posterior probability of C as a tree recursive structure, such as the one in Figure 1.

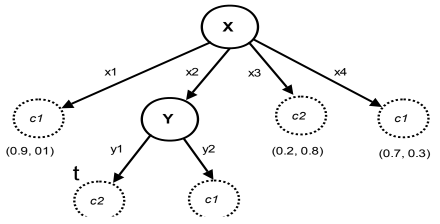

Figure 1: Classification Tree

Each path in the tree from the root node to another descendant node t (not necessarily to a leaf node) defines a configuration $\mathbf{x}^t$ for the variables in $\mathbf{X}^t$, $\mathbf{X}^t \subseteq \mathbf{X}$, where $\mathbf{X}^t$ is the set of variables that labels the internal nodes contained in the path from the root node to the descendant node t without considering the variable at node t. We also denote as $\bar{\mathbf{X}}^t = \mathbf{X} - \mathbf{X}^t$ the set of attributes not included in $\mathbf{X}^t$. E.g., supposing for the tree of Figure 1 that $\mathbf{X} = \{X, Y, Z\}$, we have at the node t: $\mathbf{x}^t = \{X = x_2, Y = y_1\}$, $\mathbf{X}^t = \{X, Y\}$ and $\bar{\mathbf{X}}^t = \{Z\}$.

We also have for each $\mathbf{x}^t$ in the tree $\mathcal{T}$ an estimate of the a posteriori conditioned probability of C given $\mathbf{x}^t$, $\hat{P}_{\mathcal{T}}(C|\mathbf{x}^t)$.

Let us $\mathcal{X}_{\mathcal{T}}$ be the set of configurations $\mathbf{x}^t$ associated to set of nodes (internal or leaf) in a tree $\mathcal{T}$ and $\hat{\mathcal{P}}_{\mathcal{X}_{\mathcal{T}}}$ their associated set of estimated probabilities. We say that a configuration $\mathbf{x}^t \in \mathcal{X}_{\mathcal{T}}$ is *consistent* with a sample $\mathbf{x}$, $\mathbf{x}^t \sim \mathbf{x}$, if for each $X_i \in \mathbf{X}^t$, $\mathbf{x}^t$ and $\mathbf{x}$ contains the same value x_i for X_i. In the previous example, if $\mathbf{x} = \{X = x_2, Y = y_1, Z = z_2\}$ then $\mathbf{x}$ is consistent with $\mathbf{x}^t = \{X = x_2, Y = y_1\}$.

Let us denote as $\mathcal{X}_{\mathcal{T}}^{\mathbf{x}} = \{\mathbf{x}^t \in \mathcal{X}_{\mathcal{T}} : \mathbf{x}^t \sim \mathbf{x}\}$ to the set of configurations $\mathbf{x}^t$ *consistent* with the sample $\mathbf{x}$. It can be seen that this set contains the configurations $\mathbf{x}^t$ of the nodes t in the path from the root node until one of the leaf nodes. This set shall contain p elements, $\mathcal{X}_{\mathcal{T}}^{\mathbf{x}} = \{\mathbf{x}^{t_0}, \mathbf{x}^{t_1}, ..., \mathbf{x}^{t_p}\}$, where t_0 is the root node ($\mathbf{x}^{t_0}$ is an empty configuration), t_1 is the children of t_0 in the previous path ($\mathbf{x}^{t_1}$ is configuration for the variable at the root node) and so on until t_p, which is the leaf node in the previous defined path. For the previous example, $\mathcal{X}_{\mathcal{T}}^{\mathbf{x}} = \{\emptyset, \{X = x_2\}, \{X = x_2, Y = Y_1\}\}$.

The largest configuration, $\mathbf{x}^{t_p}$, is taken to make the classification through the use of the estimated conditional probability $\hat{P}_{\mathcal{T}}(C|\mathbf{x}^{t_p})$.

In order to build or induce a classification tree there are two main elements that need to be defined: a *splitting criterion* and a *stop criterion*. The *splitting criterion* is based on a measure or score, $\mathbf{SC}_t$, to decide which attribute is placed in the node t' of the tree as descendent of the node t. Usually it is taken the attribute $X^t = arg\max_{X_i \in \bar{\mathbf{X}}^t} \mathbf{SC_t}(X_i)$.

The *stop criterion* at the branch of the node t is normally associated with $\mathbf{SC}_t$ and it decides when no more attributes are added to the actual configuration $\mathbf{x}^t$.

We shall use $\mathbf{x}$ instead of $\mathbf{x}^t$ for simplicity reasons when there is no possibility of confusion.

2.2 C4.5: A classification tree inducer

As we have already commented, C4.5 (Quinlan, 1993) is one of the most known and widely used

[1] Absence of missing values and continuous attributes in $\mathcal{D}$ are assumed.

classification tree inducers.

*J*48 is an advanced version of C4.5 implemented in Weka (Witten and Frank, 2000) uses the following score as splitting criterion known as *Info-Gain Ratio* (Quinlan, 1993) based on the *Info-Gain* measure, *IG*, and the *entropy* measure, *H*:

$$\mathbf{SC}_t(X) = \frac{IG(X,C)}{H(X|\mathbf{x}^t)}$$

The attributes X that does not verify that its *Info-Gain* measure is greater than the mean value of the *Info-Gain* measure of the rest of candidate variables are discarded. $\mathbf{SC}_t(X) = 0$ if $IG_{\mathbf{x}^t}(X,C) < Average\{IG_{\mathbf{x}^t}(X_i,C) : X_i \in \bar{\mathbf{X}}^t\}$. This C4.5 version stops at $\mathbf{x}^t$ when $\forall X_i \in \bar{\mathbf{X}}^t : SC_{\mathbf{x}^t}(X_i) \leq 0$.

Once the tree is built, C4.5 applies a post-pruning process (Quinlan, 1993) in order to reduce the size of the tree and avoid an overfitting to the data. This pruning process is a single bottom-up pass where the estimated error rate for each subtree is computed as an upper bound of the misclassification rate with a confidence level of 0.25 through the use of a binomial trial process. A subtree is removed if there is not reduction in the estimated error rate. We shall call $C4.5_\rho$ the version with pruning and $C4.5_{\neg\rho}$ the version without pruning.

In both cases, a Laplace estimate is used in order to smooth the probabilities of class membership.

3 A Bayesian Approach for classification tree induction

In this section, our approach is presented. Firstly, we describe the splitting criterion problem as a Bayesian model selection problem. Secondly, we express the problem of probability class estimates as a Bayesian model averaging problem. And finally, we present a method to introduce non-uniform priors in the two previous Bayesian approaches.

3.1 Bayesian Model Selection to Classification Tree induction

Bayesian model selection (BMS) (Wasserman, 2000) is an approach to choose among a set of alternative models in a model space $\mathcal{M} = \{M_1, ..., M_k\}$ where M_i is a set of probability densities with unknown parameters θ_i: $M_i = \{P_{\theta_i}(\mathbf{x}) : \theta_i \in \Omega_i\}$. In this approach the quality of a model is estimated by the posterior probability of the model given the learning data $\mathcal{D}$. Where $P(M|\mathcal{D})$ can be computed through Bayes' rule as:

$$P(M|\mathcal{D}) = \frac{P(\mathcal{D}|M)P(M)}{P(\mathcal{D})} = \frac{P(M)\int \mathcal{L}(\theta)P(\theta)d\theta}{P(\mathcal{D})}$$

where $\mathcal{L}(\theta)$ is the likelihood of the parameters given the data ($P(\mathcal{D}|M,\theta)$) and $P(\theta)$ the prior distributions over the parameters of the model.

The use of Bayesian metrics (Heckerman et al., 1994) are suitable to compute model quality because of the inherent penalty they give to the more complex models in order to prevent against over-fitting and they provide closed-form equations to compute the posterior probabilities of the model given the learning data.

The metric, denoted as $\beta(M)$, computes these probabilities assuming an uniform prior probability over the possible models, $P(M)$, and a prior Dirichlet distribution over the parameters, θ, with independence for the parameters at the different conditional distributions. Usually a global or equivalent sample size, S, is considered and then it is assumed that for each variable X the prior probability about the vector $\sigma_X = (\sigma_1, ..., \sigma_{|X|})_X$ is Dirichlet with the same parameters $\alpha_k = S/|X|$ for all values, where $|X|$ is the number of possible cases of X. $P(\mathcal{D})$ can be also discarded because is the same for all models. So that $P(M|\mathcal{D}) \propto P(\mathcal{D}|M)$. The score will be the value $\beta(M) = P(\mathcal{D}|M)$ (usually the log of this value for computational reasons).

In the classification tree induction process, the model selection problem appears when we have to choose between to stop the branching or to branch by one of the available attributes.

Formally, we are at a given leaf node t. Let us denote by $\mathcal{D}_t$ the learning data $\mathcal{D}$ restricted to $\mathbf{x}^t$ and projected over the variables $(\bar{\mathbf{X}}^t, C)$, where $\bar{\mathbf{X}}^t$ are the variables not included at the node t (or available variables for branching at node t, Section 2.1).

In the model selection problem we face here, our model space problem $\mathcal{M}$ is composed by the following alternatives:

- Model M^t: stop branching at node t.
- For each variable $X \in \bar{\mathbf{X}}^t$, model M^t_X: branch node t by variable X and then stop.

Each one of these models M in $\mathcal{M}$ is scored by $P(\mathcal{D}_t|M)$. In this computation, only the class and available variables $\bar{\mathbf{X}}^t$ are relevant. If we denote by c_i and $\bar{\mathbf{x}}^t_i$ the actual values of class variable and available attributes in case r_i of $\mathcal{D}_t$ and if we assume that each r_i is independent and identically distributed, the scores can be expressed as:

$$P(\mathcal{D}_t|M) = \prod_{r_i\in\mathcal{D}_t} P(c_i, \bar{\mathbf{x}}^t_i|M) = P(\bar{\mathbf{x}}^t_i|M).P(c_i|M, \bar{\mathbf{x}}^t_i)$$

Models M^t and M^t_X only differs in the values $P(c_i|M, \bar{\mathbf{x}}^t_i)$ as they not make any hypothesis about the way in which the rest of the variables are distributed, so that we could assume equally distributed in both cases. Then the score of model M will be proportional to $\prod_{r_i\in\mathcal{D}_t} P(c_i|M, \bar{\mathbf{x}}^t_i)$. In the concrete models we have, we obtain:

- In model M^t, variable C is independent of the rest of variables (no branching). So, $\beta(M^t) = P(\mathcal{D}_t|M^t) \propto \prod_{r_i\in\mathcal{D}_t} P(c_i|M_t)$.
- In models M^t_X, C is only dependent of the branching variable X, so that $\beta(M^t_X) = P(\mathcal{D}_t|M^t_X) \propto \prod_{r_i\in\mathcal{D}_t} P(c_i|M_t, x_i)$, where x_i is the value of the branching variable X in case r_i.

Assuming Dirichlet prior probabilities for the class C with uniform parameters $\alpha_k = S/|C|$, where S is a fixed equivalent sample size, then these values can be obtained with the standard expressions:

$$\beta(M^t) \propto \frac{\Gamma(S)}{\Gamma(S+n^t)} \frac{\prod_k \Gamma(n_{c_k} + \alpha_k)}{\prod_k \Gamma(\alpha_k)}$$

$$\beta(M^t_X) \propto \prod_j^{|X|} \frac{\Gamma(S)}{\Gamma(S+n_{x_j})} \frac{\prod_k \Gamma(n_{c_k,x_j} + \alpha_k)}{\prod_k \Gamma(\alpha_k)}$$

where $\Gamma(.)$ is the gamma function ($\Gamma(\alpha+1) = \alpha\cdot\Gamma(\alpha)$), n_{c_k,x_j} is the number of occurrences of $\{(C = c_k, X = x_j)\}$ in the learning sample at node t, $\mathcal{D}_t$ (analogously for n_{c_k}, n_{x_j}) and n^t is the number of cases in $\mathcal{D}_t$.

In that way, our approach branches at the node leaf t by the variable $X^* = arg\max_{X\in\bar{\mathbf{X}}^t}\{\beta(M^t_X)\}$. It stops branching when $\beta(M^t) > \beta(M^t_{X^*})$, in other words, when the model without branching has a higher probability.

Another important factor, which will be used when estimating the probabilities at the leaves, is that when variable X^* is selected for branching, as the score is proportional to the posterior probability of the data given the model, we have that the value

$$\mathcal{B}_t = \frac{\beta(M^t_{X^*})}{\beta(M^t)} = \frac{P(\mathcal{D}_t|M^t_{X^*})}{P(\mathcal{D}_t|M^t)}$$

As only data in $\mathcal{D}_t$ are relevant at node t, we can assume that the rest of data in $\mathcal{D}$ is equally distributed in this node and that

$$\mathcal{B}_t = \frac{P(\mathcal{D}|M^t_{X^*})}{P(\mathcal{D}|M^t)} = \frac{P(M^t_{X^*}|\mathcal{D})}{P(M^t|\mathcal{D})}$$

So we also have a value equal to the ratio between the probability of the model branching at t with X^* and not branching at all given the data.

3.2 Bayesian Model Averaging to estimate class probabilities

Most of approaches to predictive learning are based on the assumption of the whole database had been completely generated by one model, the "right" one, ignoring the underlying model decision uncertainty involved in the classifier inducing process. This problem specially happens to model selection processes in classification tree inducers, because as smaller the learning sample size is as more uncertain model selection becomes. In a classification tree, the sample size decreases exponentially with the depth of the branch, so decision of branching at final leaves accumulates a big uncertainty.

Bayesian Model Averaging (BMA) (Wasserman, 2000; Hoeting et al., 1999) provides a foundation to deal with this uncertainty through the use of a weighting scheme for each candidate hypothesis of the hypothesis space computing its posterior probability given the learning data.

Formally, we start with a hypothesis space $\mathcal{H}$ and a set of learning data $\mathcal{D}$. So, the posterior probability of C given a new sample $\mathbf{x}$ is computed by:

$$P(C|\mathbf{x}, \mathcal{D}, \mathcal{H}) = \sum_{h \in \mathcal{H}} P(C|\mathbf{x}, h)P(h|\mathcal{D})$$

Our application of BMA is an alternative to pruning the final tree: as for each inner nodes t we can compute the value $\mathcal{B}_t$ which is proportional to the ratio of the probability of the model branching by variable X^* and the probability of the model without branching, instead of deciding by one of the models, we can make the average of the probabilities estimated with each one of the models weighted by a value proportional to their probability. This averaging is applied in each leaf node for all the nodes in the path from this node to the root.

One advantage of this application of BMA is that only the final estimation of the probabilities at the leaves change, having only one decision tree structure.

In this way, for each inner node t_i we compute a weight proportional to the posterior probability that the induced tree stops at this point (that is denoted as the hypothesis h^{t_i}) and it is computed using the *Bayes factors* $\mathcal{B}_{t_j}$ in the following way:

$$\hat{P}(h^{t_i}|\mathcal{D}) \propto \prod_{j=1}^{i-1} \mathcal{B}_{t_j} = \prod_{j=1}^{i-1} \frac{P(M_{X^*}^{t_j}|\mathcal{D})}{P(M^{t_j}|\mathcal{D})}$$

where $\hat{P}(h^{t_1}|\mathcal{D}) = 1$.

So, the estimated probability in a leaf node t_p is computed as follows:

$$P(c_k|\mathbf{x}^{t_p}) \propto \sum_{i=1}^{p} \frac{n_{c_k \mathbf{x}^{t_i}} + \alpha_k}{n_{\mathbf{x}^{t_i}} + S} \hat{P}(h^{t_i}|\mathcal{D})$$

where $n_{\mathbf{x}^{t_i}}$ is the size of the learning sample at the node t_i, $\mathcal{D}_{t_i}$, and $n_{c_k \mathbf{x}^{t_i}}$ is the number of occurrences of $\{C = c_k\}$ in this set $\mathcal{D}_{t_i}$. The α_k and S values correspond to the same Dirichlet prior probability used in the induction process of the previous Section 3.1. Finally, a normalization is required.

This approach has the advantage that all these probabilities can be efficiently computed at the same time that the tree is built, with only a linear increasing in the complexity.

3.3 A non-Uniform Priors Approach

In all the previous developments we have assumed uniform values, $\alpha_k = S/|C|$, for the parameters of the prior Dirichlet distributions. In this subsection, we are giving a new approach to define non-uniform priors in the induction of decision trees, which will be incorporated in the computation of the Bayesian split criterion, Section 3.1, and in the computation of averaging probabilities, Section 3.2, trough the definition of new α_k values.

To justify this approach, we start with the following idea: if at some node t_i the frequency n_{c_k} is zero, then $\forall j > i$ at t_j descendant nodes the frequency n_{c_k} shall also be zero. So it makes sense to assume that $n_{c_k \mathbf{x}}$ will probably be zero or close to zero for most of future samples $\mathbf{x}$. So decreasing the prior probability for c_k at $\mathbf{x}^{t_{i+1}}$ is coherent.

We propose the following heuristic to modify the parameters of the Dirichlet priors distributions: Let us $\delta_i = |\{n_{c_k} = 0 : c_k \in C\}|$ in $\mathcal{D}_{t_i}$, if $\delta_i > 1$ we define $\alpha_k^{t_i+1}$ as follows:

$$\alpha_k^{t_i+1} = \frac{S}{(|C| - \delta_i + 1)} : n_{c_k \mathbf{x}^{t_i}} \neq 0$$

$$\alpha_k^{t_i+1} = \frac{S}{(|C| - \delta_i + 1)\delta_i} : n_{c_k \mathbf{x}^{t_i}} = 0$$

As we can see, those cases with non-null frequency have the same prior probability, $\frac{1}{(|C| - \delta_i + 1)}$, while all those cases with null frequency share among them the same probability mass of one non-null frequency case, i.e., $\frac{1}{(|C| - \delta_i + 1)\delta_i}$. Let us point out that for a two-class problem we get with this heuristic a uniform prior.

4 Experimental Results

In this section, we present the experimental evaluation of our approach. Firstly, the evaluation methodology is described and, after that, the experimental results of the different approaches are presented.

4.1 Evaluation Methodology

We used the following 27 databases from the UCI repository: *anneal, audiology, autos, breast-cancer, colic, credit-german, diabetes-pima, glass-2, hepatitis, hypothyroid, ionosphere, kr-vs-kp, labor, letter, lymph, mfeat-pixel, mushrooms, optdigits, segment, sick, solar-flare, sonar, soybean, sponge, vote, vowel and zoo*. The features of the databases are very different among them: from 2 to 24 class cases, from 57 to 20000 samples and from 9 to 240 attributes.

The classification tree inducers were implemented in Elvira platform (Consortium, 2002) and evaluated in Weka (Witten and Frank, 2000). And then we preprocessed the data replacing the missing values (with the mode value for nominal attributes and with the mean value for continuous attributes) and discretized with an equal-frequency method with 5 bins using Weka's own filters.

Two evaluation or performance measures are employed in this experimental evaluation: the classical prediction accuracy (noted as *Accuracy*); and the logarithm of the likelihood of the true class, computed as: *log-likelihood*= $\ln(\hat{P}(c_{r_i}|\mathbf{x}))$, where c_{r_i} is the true class value of the example of the test data set. This last score is introduced with the aim to evaluate the precision of probability class estimates. The usefulness of this score for this task is justified in many ways, as for example in (Roulston and Smith, 2002; Gneiting and Raftery, 2005).

The evaluation of the classifiers was achieved by a 10-fold-cross validation repeated 10 times scheme for each database. So, 100 train and test evaluations are carried out. With these estimates, the comparison among classifiers was achieved using a corrected paired t-test (Nadeau and Bengio, 2003) implemented in Weka with a 1% of statistically significant level. In this way, a classifier is fixed as reference (marked with $\star$) and then each proposed classifier is compared against it. The comparison is made summing up the times that the proposed classifier gets a statistically significant difference respect to the reference classifier in accordance with the corrected paired t-test in a given database. The test result can show an statistically significant improvement or Win (marked with W), a not statistically significant difference or Tie (marked with T) and a statistically significant deterioration (marked with D) in the evaluation measures. These results are shown in the rows starting with *W/T/D*. E.g., in Table 1, $\beta_{S=1}$ gets a statistically improvement or win in the accuracy respect to $C4.5_\rho$ in 3 databases, there is no differences in 23 databases and it looses or gets a significant deterioration in the accuracy respect to $C4.5_\rho$ in 1 databases.

As it is commented, our approach is three fold: a Bayesian model selection (BMS) approach as splitting criterion, Section 3.1; a Bayesian model averaging approach (BMA) to estimate the probabilities class membership, Section 3.2; and a non-uniform prior (NUP) definition approach, Section 3.3. In all cases, we use the same prior Dirichlet distributions with the same global sample size, S.

Let us define the four combinations we evaluate: β_S: only BMS; $\widehat{\beta}_S$: BMS + BMA; β_S^θ: BMS + NUP; $\widehat{\beta}_S^\theta$: BMS + BMA + NUP.

In all cases, three different global sample sizes are evaluated: $S=1$, $S=2$ and $S=|C|$.

4.2 Bayesian Metric as splitting criterion for inducing CT

We test the use of a Bayesian metric as a splitting criterion for inducing classification trees (CT), previously describe in Section 3.1. In order to compare its efficiency as a CT inducer, we compare, in Table 1, the performance of their induced trees respect to the trees induced by $C4.5$ with pruning ($C4.5_\rho$) and without pruning process ($C4.5_{\neg\rho}$).

The results of Table 1 show as the use of a Bayesian metric with $S=1$ and $S=2$ as splitting criterion is competitive to $C4.5_\rho$ and $C4.5_{\neg\rho}$ in terms of accuracy: it wins in three to five databases (always databases with high number

Table 1: Bayesian metric as Splitting Criterion

Classifier	$\star C4.5_{\rho}$	$\beta_{S=1}$	$\beta_{S=2}$	$\beta_{S=\|C\|}$
Accuracy	85.50	85.30	85.56	84.03
W/T/D		3/23/1	3/24/0	1/23/3
log-likelih.	-0.79	-0.79	-0.78	-0.78
W/T/D		4/20/3	6/20/1	5/21/1

Classifier	$\star C4.5_{\neg\rho}$	$\beta_{S=1}$	$\beta_{S=2}$	$\beta_{S=\|C\|}$
Accuracy	84.08	5/22/0	5/22/0	1/24/2
log-likelih.	-0.84	7/19/1	9/17/1	9/17/1
Tree Size	482.5	482.1	459.9	319.6

of classes) and it looses once in *sick* database (a very imbalanced two class data set). In terms of log-likelihood the behavior of β_S is competitive and slightly better (significance differences in 6 databases of $S = 2$), which is important considering the absence of a complex and costly pruning process of our approach.

It is interesting to see as the tree size is quite similar to $C4.5_{\neg\rho}$ for $S = 1$ and $S = 2$. But, obviously, it is greater than $C4.5_{\rho}$ average tree size: 306.6 nodes.

4.3 Bayesian Model Averaging for probability class estimate

Here it is evaluated the introduction of our BMA approach, $\widehat{\beta}_S$. Firstly, we compare this approach versus its respective version without probability averaging, β_S, using the same S in each case. It is also evaluated respect to $C4.5_{\rho}$.

Table 2: Bayesian Model Averaging

Classifier	$\star\beta_S$	$\widehat{\beta}_{S=1}$	$\widehat{\beta}_{S=2}$	$\widehat{\beta}_{S=\|C\|}$
Accuracy		85.50	85.82	83.98
W/T/D		0/27/0	0/27/0	0/23/1
log-likelih.		-0.63	-0.63	-0.77
W/T/D		12/14/1	14/11/2	4/20/3

Classifier	$\star C4.5_{\rho}$	$\widehat{\beta}_{S=1}$	$\widehat{\beta}_{S=2}$	$\widehat{\beta}_{S=\|C\|}$
Accuracy		3/24/0	3/24/0	1/23/3
log-likelih.		12/15/0	11/15/1	4/22/1

Results are presented in Table 2. As we can see, the introduction of the BMA approach do not produce an improvement in terms of accuracy (although it avoids the defeat versus $C4.5_{\rho}$ for $S = 1$). But, in terms of log-likelihood, there is a clear outperforming for $S = 1$ and $S = 2$ respect to the basic version, β_S, and respect to $C4.5_{\rho}$, excepting $S = |C|$.

4.4 Non-Uniform Dirichlet Priors Definition

Finally, we test the introduction of non-uniform priors in both Bayesian approaches for model selection and model averaging. In Table 3 the comparative results are divided in 4 folds.

Firstly, we evaluate the introduction of non-uniform priors in the Bayesian metric as splitting criterion, β_S^{θ}, where it is compared respect to the basic version, β_S, with the same S value. In the second part, it is evaluated the non-uniform priors definition at the BMA approach, $\widehat{\beta}_S^{\theta}$, also comparing against the basic version, $\widehat{\beta}_S$. And in the last two parts, we compare the full approach $\widehat{\beta}_S^{\theta}$ respect to the two versions of $C4.5$.

Table 3: Non-Uniform Priors Definition

Classifier	$\star\beta_S$	$\beta^{\theta}_{S=1}$	$\beta^{\theta}_{S=2}$	$\beta^{\theta}_{S=\|C\|}$
Accuracy		85.64	85.82	85.20
W/T/D		2/25/0	2/25/0	3/24/0
log-likelih.		-0.82	-0.81	-0.78
W/T/D		0/21/6	0/21/6	0/25/2

Classifier	$\star\widehat{\beta}_S$	$\widehat{\beta}^{\theta}_{S=1}$	$\widehat{\beta}^{\theta}_{S=2}$	$\widehat{\beta}^{\theta}_{S=\|C\|}$
Accuracy		85.85	86.04	85.37
W/T/D		2/25/0	2/25/0	4/23/0
log-likelih.		-0.61	-0.60	-0.69
W/T/D		3/23/0	5/21/0	10/17/0

Classifier	$\star C4.5_{\rho}$	$\widehat{\beta}^{\theta}_{S=1}$	$\widehat{\beta}^{\theta}_{S=2}$	$\widehat{\beta}^{\theta}_{S=\|C\|}$
Accuracy		3/24/0	4/23/0	0/27/0
log-likelih.		13/14/0	11/15/1	10/16/1

Classifier	$\star C4.5_{\neg\rho}$	$\widehat{\beta}^{\theta}_{S=1}$	$\widehat{\beta}^{\theta}_{S=2}$	$\widehat{\beta}^{\theta}_{S=\|C\|}$
Accuracy		7/20/0	6/21/0	3/24/0
log-likelih.		12/15/0	12/14/1	11/15/1
Tree Size		596.2	591.2	491.2

Summarizing, we find that the effect of the introduction of non-uniform priors is more clear at $\widehat{\beta}_S^{\theta}$ comparison. It supposes a slight improvement in terms of accuracy (2 wins) but strong in terms of log-likelihood (specially for $S = |C|$). The evaluation respect to $C4.5_{\rho}$ remains quite similar for $S = 1$ and $S = 2$ but suppose an important enhancement for $S = |C|$.

A close review of the results at database level indicates that the introduction of non-uniform priors implies better estimates in those databases where the Bayesian approach with uniform priors has already achieved them. That is to say, non-uniform priors is suitable for databases with a high number of classes.

5 Conclusions and Future Works

We have presented a new method to induce classification trees with a Bayesian model selection approach as split criterion and with a Bayesian model averaging approach to estimates probability class. We also introduce a new approach to define non-uniform priors over the parameters of the models.

We have carried out an experimental evaluation over 27 different UCI data sets comparing against one of the state-of-the-art tree inducers, $J48$. We have shown as these approaches suppose an slight but robust improvement in terms of accuracy, while all of them offer an important improvement in terms of better probability class estimates by the induced decision trees.

Acknowledgments

This work has been jointly supported by the Spanish Ministry of Education and Science under project TIN2007-67418-C03-03, by European Regional Development Fund (FEDER), the FPU scholarship programme (AP2004-4678) and by the Spanish Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía under project TIC-276.

References

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. 1984. *Classification and Regression Trees.* Wadsworth International Group, Belmont.

L. Breiman. 1996a. Out-of-bag estimation. *Private communication.*

Leo Breiman. 1996b. Bagging predictors. *Mach. Learn.*, 24(2):123–140.

Elvira Consortium. 2002. Elvira: An environment for probabilistic graphical models. In J.A. Gámez and A. Salmerón, editors, *Proceedings of the 1st European Workshop on Probabilistic Graphical Models*, pages 222–230.

Tilmann Gneiting and Adrian E. Raftery. 2005. Strictly proper scoring rules, prediction, and estimation. *Technical Report. Department of Statistics, University of Washington*, 463R.

David Heckerman, Dan Geiger, and David Maxwell Chickering. 1994. Learning bayesian networks: The combination of knowledge and statistical data. In *KDD Workshop*, pages 85–96.

Jennifer A. Hoeting, David Madigan, Adrian E. Raftery, and Chris T. Volinsky. 1999. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–417.

Tjen-Sien Lim, Wei-Yin Loh, and Yu-Shan Shih. 2000. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Mach. Learn.*, 40(3):203–228.

Claude Nadeau and Yoshua Bengio. 2003. Inference for the generalization error. *Mach. Learn.*, 52(3):239–281.

Michael J. Pazzani, Christopher J. Merz, Patrick M. Murphy, Kamal Ali, Timothy Hume, and Clifford Brunk. 1994. Reducing misclassification costs. In *ICML*, pages 217–225.

Foster Provost and Pedro Domingos. 2003. Tree induction for probability-based ranking. *Mach. Learn.*, 52(3):199–215.

Foster Provost and Venkateswarlu Kolluri. 1999. A survey of methods for scaling up inductive algorithms. *Data Min. Knowl. Discov.*, 3(2):131–169.

J.R. Quinlan. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Francisco.

M. S. Roulston and L.A. Smith. 2002. Evaluating probabilistic forecasts using information theory. *Monthly Weather Review*, 130:1653–1660.

Larry Wasserman. 2000. Bayesian model selection and model averaging. *J. Math. Psychol.*, 44(1):92–107.

Ian H. Witten and Eibe Frank. 2000. *Data mining: practical machine learning tools and techniques with Java implementations.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.