Model Switching for Bayesian Classification Trees with Soft Splits

Jörg Kindermann and Gerhard Paass

RWCP - Theoretical Foundation Lab
GMD – German National Research Center for Information Technology
D-52754 Sankt Augustin, Germany
kindermann@gmd.de paass@gmd.de *

Abstract. Due to the high number of insolvencies in the credit business, automatic procedures for testing the credit-worthiness of enterprises become increasingly important. For this task we use classification trees with soft splits which assign the observations near the split boundary to both branches. Tree models involve an extra complication as the number of parameters varies as the tree grows and shrinks. Hence we adapt the reversible jump Markov Chain Monte Carlo procedure to this model which produces an ensemble of trees representing the posterior distribution. For a real-world credit-scoring application our algorithm yields lower classification errors than bootstrapped versions of regression trees (CART), neural networks, and adaptive splines (MARS). The predictive distribution allows to assess the certainty of credit decisions for new cases and guides the collection of additional information.

1 Introduction

During the last years *local* approaches are increasingly used for classification and regression tasks. They cover the input space $\mathcal{X}$ with a finite set of local regions $\mathcal{X}_\tau$ such that $\mathcal{X} = \bigcup_\tau \mathcal{X}_\tau$. Within each region the function $y = f(\boldsymbol{x})$ of interest is approximated by simple local functions $\hat{f}_\tau(\boldsymbol{x})$, e.g. a constant, a linear or quadratic polynomial. This is motivated by Taylor's theorem which states that if a local region is small enough any continuous function can be well approximated by a low order polynomial within it.

It is well known, that the mean square error can be decomposed into a systematic component (bias) and a random component (variance) [8]. For local learning the variance can be reduced by increasing the number of observations in a local region. On the other hand the bias is inflated if the size of the region grows, as then the data – especially near the border of the region – does no longer reflect the local characteristics of the underlying function. In addition the variance usually grows if the number of free parameters of the local function is increased. Hence the selection of local regions always involves a compromise.

Obviously the shape of regions should reflect the approximation properties of the local functions. A region can have a long extension in a direction, where the local function can approximate the real relation with little error. A new region should begin where the

* We thank Prof. Dr. Jörg Baetge, University of Münster for granting access to the dataset.

error gets larger than a certain threshold. There are two basic types of local procedures [7]:

- *Radial basis function* networks and *k-nearest neighbour* [13] models create local regions whose shape is globally uniform or is defined by the distribution of input values in the sample. Hence the shape is not determined according to the approximation error. For high input dimensions k the diameter of "local" regions containing a fraction of γ of the sample elements is approximately $\gamma^{1/k}$ of the regions containing all elements. Hence local regions in general have a large diameter even if they cover only a small fraction of sample values (curse of dimensionality).
- A *classification tree* [2, 3, 12] recursively partitions the input space into a number of disjunct regions $\mathcal{X}_\tau$ and separately trains a model $f_\tau(\boldsymbol{x})$ in each region. The regions are formed in such a way that the approximation error is minimised. However, as each region requires a number of observations to estimate the local function relatively few regions can be formed and the approximation error gets large near the boundaries.

In this paper we propose a tree based classification procedure which tries to combine the advantages of both types:

- Instead of recursively splitting each regions in two separate subregions we form an intermediate region between the new regions (soft split). Observations located in this intermediate region are assigned to both regions for training as well as for prediction. This increases the average number of observations in each final region. The bias is potentially reduced as a point near the boundary is predicted as the average of the predictions of adjacent regions.
- Not a single 'optimal' tree is determined in a greedy fashion, but using *Bayesian* statistics a large number of plausible trees is constructed. Each tree has a different set of local regions with different boundaries. As the prediction is determined as the average prediction of the single trees the biases near boundaries are potentially reduced.

The gradual change between the regions of a tree has been considered by several authors. Carter and Catlett [4] define upper and lower split points and use linear interpolation to smooth the "membership". Quinlan [12, p.75] discusses this setup and suggests a method for choosing cut-points based on the distribution of misclassified training cases. Ripley [13, p.239] suggests the use of a logistic function for "membership" and Friedman utilises splines [6] or a step function [7]. Jacobs and collaborators [10] proposed soft splits which are not perpendicular to the input axes but are induced by specific "gating models". For a review of these mixture-of-expert models see [14].

The procedure described in this paper for the first time applies the Bayesian paradigm to trees with soft splits along the coordinate axes. Bayesian statistics derives a probability measure on the model parameters, which characterises the plausibility that a model with this parameter has generated the data [1]. Tree models involve an extra complication as the number of parameters varies as the tree grows or shrinks. Hence we require a procedure which is able to switch between different model structures. We describe a Bayesian method for classification trees of varying size and implement the corresponding Markov Chain Monte Carlo procedure as a variant of [5].

We apply the procedure to a real world *credit-scoring* problem. In this setup it is vital to take into account the misclassification costs by a loss function. We compare the Bayesian approach with alternatives like neural networks, MARS [6], CART [2] and linear discriminant analysis as well as their bootstrapped versions [11]. Bayesian methods derive a predictive distribution for a new input. This may be used to assess the reliability of a classification in a real world situation.

In the next section we introduce the basic concepts of Bayesian classification and the Metropolis-Hastings algorithm which is used to obtain a representative set of models. Then an extension of the algorithm to switching between models of different dimensionality is described. Section 3 describes Bayesian classification trees with overlapping leaves and the generation of a Markov chain of trees. Section 4 describes a credit-scoring application and present the results for a set of over 6000 data records of balance sheet figures. The last section summarises the paper.

2 Bayesian Classification

2.1 Basic Concepts

Assume for each object we have features $\boldsymbol{x} = (x_1, \ldots, x_k) \in \mathcal{X}$ and want to determine the class $y \in \{0, 1\}$ of the object. We assume a non-deterministic relation between $\boldsymbol{x}$ and y, given by a conditional distribution $p(y|\boldsymbol{x}, \boldsymbol{w})$ with unknown parameters $\boldsymbol{w} \in \mathcal{W}$. Suppose we have an independent random sample $(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)})$, where $y^{(i)}$ is distributed according to the 'true' $\dot{p}(y|\boldsymbol{x}^{(i)})$ conditional distribution. We denote the input data by $\boldsymbol{X} := (\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)})$ and the output data by $\boldsymbol{y} := (y^{(1)}, \ldots, y^{(n)})$.

Let $p(\boldsymbol{w})$ be our *prior* distribution of the model parameters describing the relative plausibility of parameter values *before* any data is available. For fixed data $\boldsymbol{X}$ and $\boldsymbol{y}$ $p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) := \prod_{i=1}^{n} p(y^{(i)}|\boldsymbol{x}^{(i)}, \boldsymbol{w})$ is the *likelihood* of $\boldsymbol{w}$. Then the *Bayesian formula* yields the *posterior distribution*

$$p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}) = \frac{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w})\, p(\boldsymbol{w})}{\int p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w})\, p(\boldsymbol{w})\, d\boldsymbol{w}} \tag{1}$$

It describes the relative plausibility of different parameters $\boldsymbol{w}$ after the data $\boldsymbol{X}$ and $\boldsymbol{y}$ has been observed. Let q_1 be the probability that y has the class-value 1. For a fixed input $\boldsymbol{x}$ the 'parameter' q_1 is uncertain and has a probability distribution determined by the distribution of $\boldsymbol{w}$. We can compute the probability $P(q_1 \leq \eta|\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{X})$ that q_1 is smaller than some η as $\int_{\{\boldsymbol{w}|p(y=1|\boldsymbol{x},\boldsymbol{w})\leq\eta\}} p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X})\, d\boldsymbol{w}$. This means, that the class probability itself is uncertain, and the data supports different probability values to a varying extent. The expected 'average' probability that a new object with features $\boldsymbol{x}$ belongs to class 1 is

$$E(q_1|\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{X}) = \int p(y{=}1|\boldsymbol{x}, \boldsymbol{w})\, p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X})\, d\boldsymbol{w} \tag{2}$$

For classification we have to decide which class should be selected. More generally we have to take an action $a \in \mathcal{A}$, whose consequences depend on the class c of an object. In the case of credit-scoring a may be 'grant credit' or 'deny credit'. Let $L(a; c) \in \Re$ be

the loss we incur, if action a is taken and c is the actual class of the object. The *expected loss* $E(L(a)|\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{X})$ of a is

$$\int \sum_{y=0}^{1} L(a;y)\, p(y|\boldsymbol{x}, \boldsymbol{w})\, p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X})\, d\boldsymbol{w} = \sum_{y=0}^{1} L(a;y)\, E(q_y|\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{X}) \tag{3}$$

According to the Bayesian decision theory [1] it is optimal to select the action a with *minimal expected loss*. Note that only the mean value of q_c enters the decision, not its variance.

2.2 Markov Chain Monte Carlo Analysis

As (2) and (3) in general cannot be evaluated analytically, we have to perform a Markov chain Monte-Carlo (MCMC) analysis. This involves the construction of a Markov chain $\boldsymbol{w}(0), \boldsymbol{w}(1), \ldots$ designed to be distributed according to the posterior density $p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X})$. If the chain is currently at $\boldsymbol{w} = \boldsymbol{w}(t)$, the *Metropolis-Hastings algorithm* [15] requires a *proposal density* $\boldsymbol{q}(\boldsymbol{w}, \tilde{\boldsymbol{w}})$, which is the conditional distribution of proposing a move from $\boldsymbol{w}$ to $\tilde{\boldsymbol{w}}$. The *acceptance probability* is defined as

$$p^{\text{acc}}(\boldsymbol{w}, \tilde{\boldsymbol{w}}) = \min\left\{1, \frac{p(\tilde{\boldsymbol{w}}|\boldsymbol{y}, \boldsymbol{X})\, \boldsymbol{q}(\tilde{\boldsymbol{w}}, \boldsymbol{w})}{p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X})\, \boldsymbol{q}(\boldsymbol{w}, \tilde{\boldsymbol{w}})}\right\} \tag{4}$$

With probability $p^{\text{acc}}(\boldsymbol{w}, \tilde{\boldsymbol{w}})$ the candidate $\tilde{\boldsymbol{w}}$ is accepted and the chain moves to $\boldsymbol{w}(t+1) = \tilde{\boldsymbol{w}}$. Otherwise the candidate is rejected and $\boldsymbol{w}(t+1)$ takes the old value $\boldsymbol{w}$. For the actual transition probability $\boldsymbol{p}(\boldsymbol{w}, \tilde{\boldsymbol{w}}) := \boldsymbol{q}(\boldsymbol{w}, \tilde{\boldsymbol{w}}) p^{\text{acc}}(\boldsymbol{w}, \tilde{\boldsymbol{w}})$ the *detailed balance* condition holds for all $\boldsymbol{w}, \tilde{\boldsymbol{w}}$

$$p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X})\, \boldsymbol{p}(\boldsymbol{w}, \tilde{\boldsymbol{w}}) = p(\tilde{\boldsymbol{w}}|\boldsymbol{y}, \boldsymbol{X})\, \boldsymbol{p}(\tilde{\boldsymbol{w}}, \boldsymbol{w}) \tag{5}$$

If the resulting Markov chain is aperiodic and irreducible (i.e. reaches all states with positive probability) then its distribution converges to an invariant stationary limit distribution, which is just the posterior distribution $p(\boldsymbol{w}|\boldsymbol{y}, X)$ [15].

For decision trees, where the number and interpretation of parameters varies, the approach cannot be used. Recently Green [9] has proposed an MCMC-scheme for varying dimension problems, termed *reversible jump MCMC*. When the current state is $\boldsymbol{w}$ and $p(\boldsymbol{w}|\boldsymbol{y}\boldsymbol{X})$ is the target probability measure (the posterior density) we consider a countable number of different moves m. Depending on the state $\boldsymbol{w}$ a move m and a destination $\tilde{\boldsymbol{w}}$ is proposed with a joint distribution $q_m(\boldsymbol{w}, \tilde{\boldsymbol{w}})$. $q_m(\boldsymbol{w}, \tilde{\boldsymbol{w}})$ may be a sub-probability measure, with probability $1 - \sum_m \int_{\tilde{\boldsymbol{w}}} q_m(\boldsymbol{w}, \tilde{\boldsymbol{w}}) d\tilde{\boldsymbol{w}}$ no move is attempted.

For the case that $\boldsymbol{w}$ and $\tilde{\boldsymbol{w}}$ have the same dimension, the procedure reduces to the Metropolis-Hastings algorithm (4). Now suppose that starting from $\boldsymbol{w}$ a move of type m is proposed that yields a higher-dimensional $\tilde{\boldsymbol{w}}$. This can be implemented by drawing a vector $\boldsymbol{u}$ of continuous variables distributed according to a known density $p_m(\boldsymbol{u})$ independent of $\boldsymbol{w}$. It is required that the sum of the dimensions of $\boldsymbol{w}$ and $\boldsymbol{u}$ is equal to the dimension of $\tilde{\boldsymbol{w}}$. Then the new state $\tilde{\boldsymbol{w}}$ is defined by an invertible deterministic function $\tilde{\boldsymbol{w}} = h_m(\boldsymbol{w}, \boldsymbol{u})$. The reverse of the move can be accomplished by using the

inverse transformation, so that the proposal is deterministic. Then we get the acceptance probability

$$p_m^{\text{acc}}(\boldsymbol{w}, \tilde{\boldsymbol{w}}) = \min\left(1, \left|\frac{\partial h_m(\boldsymbol{u}, \boldsymbol{w})}{\partial(\boldsymbol{u}, \boldsymbol{w})}\right| * \frac{p(\tilde{\boldsymbol{w}}|\boldsymbol{y}, \boldsymbol{X})\, j_m(\tilde{\boldsymbol{w}})}{p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X})\, j_m(\boldsymbol{w})\, p_m(\boldsymbol{u})}\right) \tag{6}$$

Here $j_m(\boldsymbol{w})$ and $j_m(\tilde{\boldsymbol{w}})$ are the probabilities of selecting move m or its inverse in states $\boldsymbol{w}$ and $\tilde{\boldsymbol{w}}$ respectively. Green [9] shows that the detailed balance condition (5) holds and consequently the equilibrium distribution of the resulting Markov chain is the posterior distribution $p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X})$. Similar to the usual Metropolis-Hastings formula (4) the densities have to be known only up to a factor, which cancels out in (6).

3 Bayesian Classification Trees

3.1 Overlapping Regions

Tree models divide the predictor space $\mathcal{X}$ into rectangular regions $\mathcal{X}_\tau$ by recursive splits along the coordinate axes and assume that the conditional distribution of y within the terminal regions $\mathcal{X}_\tau$ is identical for all $\boldsymbol{x}$. In this paper we recursively divide an existing region $\mathcal{X}_\tau$ into two new regions $\mathcal{X}_{\tau_1}$ and $\mathcal{X}_{\tau_2}$ which are not disjoint, but may have some overlap. We select a variable x_{s_τ} and define

$$\begin{array}{l} \mathcal{X}_{\tau_1} = \{\boldsymbol{x} \in \mathcal{X}_\tau | x_{s_\tau} \leq \xi_\tau^+\} \\ \mathcal{X}_{\tau_2} = \{\boldsymbol{x} \in \mathcal{X}_\tau | x_{s_\tau} > \xi_\tau^-\} \end{array} \qquad \text{where } \xi_\tau^- \leq \xi_\tau^+ \tag{7}$$

with *upper split point* ξ_τ^+ and *lower split point* ξ_τ^-. A recursive application of this procedure yields a binary tree structure. A Bayesian model has to specify, how the data is generated from the underlying model. If the tree consists of a single leaf $\mathcal{X}_\tau$ the dependent variable y is assumed to be binomially distributed with parameter $\theta_{\tau,1} := p(y{=}1|\boldsymbol{x} \in \mathcal{X}_\tau)$. If the tree consists of two overlapping leaves $\mathcal{X}_{\tau_1}$ and $\mathcal{X}_{\tau_2}$, then in $\mathcal{X}_{\tau_1} \setminus \mathcal{X}_{\tau_2}$ the class variable y again is assumed to be binomially distributed with parameter θ_{τ_1} and in $\mathcal{X}_{\tau_2} \setminus \mathcal{X}_{\tau_1}$ y is assumed to be binomially distributed with parameter θ_{τ_2}. In $\mathcal{X}_{\tau_2} \cap \mathcal{X}_{\tau_1}$ it follows a binomial 'mixture' distribution with parameter $(\theta_{\tau_1} + \theta_{\tau_2})/2$. This scheme is recursively applied if a region $\mathcal{X}_\tau$ is covered by two subregions $\mathcal{X}_{\tau_1}$ and $\mathcal{X}_{\tau_2}$: The distribution in the overlap is just the mixture with weights 1/2 between the distributions within $\mathcal{X}_{\tau_1}$ and $\mathcal{X}_{\tau_2}$.

For prediction this has the interesting consequence, that a new observation $\boldsymbol{x}$ 'belongs' to more than one leaf and the predicted class probability is a mixture of the separate predictions of each leaf. If the two split points ξ_τ^- and ξ_τ^+ are identical for all τ, we get the usual binary trees as a special case. As an alternative, more complex distributions within the leaves can be considered, e.g. multinomial distributions or generalised linear models.

3.2 Prior and Posterior Distributions

Recall that each region $\mathcal{X}_\tau$ has various intersections with other regions $\mathcal{X}_\eta$, and that within these intersections the distribution is a weighted mixture of the binomial distributions with parameters $\boldsymbol{\theta}_\tau$ and $\boldsymbol{\theta}_\eta$ and known weights. Consequently an observation

$(\boldsymbol{x}^{(i)}, y^{(i)})$ is attributed to the different regions with varying weights. For a region $\mathcal{X}_\tau$ this uniquely determines the vector of counts $\boldsymbol{m}_\tau = (m_{\tau,0}, m_{\tau,1})$, i.e. the number of y-values in each class.

As prior density $p(\boldsymbol{\theta})$ for the binomial parameter $\boldsymbol{\theta} = ((1-\theta), \theta)$ we use the Dirichlet density $\mathrm{Di}(\alpha)$ $p(\boldsymbol{\theta}|\alpha) \propto \theta_0^{\alpha_0 - 1}\theta_1^{\alpha_1 - 1}$. If $\boldsymbol{y}$ has the counts $\boldsymbol{m}$ we get the likelihood $p(\boldsymbol{y}|\boldsymbol{\theta}) \propto \theta_0^{m_0}\,\theta_1^{m_1}$. Hence in each leaf we get posteriors of the form $p(\boldsymbol{\theta}|\boldsymbol{m}) \propto \theta_0^{m_0+\alpha_0-1}\theta_1^{m_1+\alpha_1-1}$ which again are Dirichlet densities $\mathrm{Di}(\alpha + \boldsymbol{m})$.

3.3 Comparing Splits with the Un-split Leaf

We want to assess the effect of splitting some leaf $\mathcal{X}_\tau$ containing n_τ observations into two subsets $\mathcal{X}_{\tau_1}$ and $\mathcal{X}_{\tau_2}$. We assume that within $\mathcal{X}_{\tau_i}$ the dependent variable is binomially distributed as $\mathrm{Bi}(\boldsymbol{\theta}_{\tau_i}, n_{\tau_i})$. Hence we have to compare two hypotheses:

H_1: within each $\mathcal{X}_{\tau_i}$ the class variable $\boldsymbol{y}$ has been generated according to a binomial distribution $\mathrm{Bi}(\boldsymbol{\theta}_\tau, n_{\tau_i})$ with a single but unknown parameter $\boldsymbol{\theta}_\tau$.

H_2: within each $\mathcal{X}_{\tau_i}$ the class variable $\boldsymbol{y}$ has been generated according to a binomial distribution $\mathrm{Bi}(\boldsymbol{\theta}_{\tau_i}, n_{\tau_i})$ with different unknown parameters $\boldsymbol{\theta}_{\tau_i}$.

We assume that on $\mathcal{X}_\tau$ and $\mathcal{X}_{\tau_i}$ the prior is Dirichlet $p(\boldsymbol{\theta}|\alpha_{\tau_i}) = \mathrm{Di}(\alpha_{\tau_i})$. As the posterior again is Dirichlet we get after some tedious algebra

$$\frac{p(H_1|\boldsymbol{y}, \boldsymbol{X})}{p(H_2|\boldsymbol{y}, \boldsymbol{X})} = \frac{p(\boldsymbol{y}|\boldsymbol{X}, H_1)p(H_1)}{p(\boldsymbol{y}|\boldsymbol{X}, H_2)p(H_2)} = \frac{p(H_1)\, R_{Di}(\boldsymbol{m}_\tau, \alpha_\tau)}{p(H_2)\, R_{Di}(\boldsymbol{m}_{\tau_1}, \alpha_{\tau_1})\, R_{Di}(\boldsymbol{m}_{\tau_2}, \alpha_{\tau_2})}$$

where $R_{Di}(\boldsymbol{m}, \alpha) := \eta_{Di}(\boldsymbol{m} + \alpha)/\eta_{Di}(\alpha)$, $\eta_{Di}(\alpha) = \Gamma(\alpha_0)\Gamma(\alpha_1)/\Gamma(\sum_c \alpha_c)$, and $p(H_i)$ is the prior probability of a hypothesis. This is the *Bayes factor* for comparing the models.

3.4 Possible Moves

To reduce the computational effort we make changes only at the bottom of the tree, as otherwise the calculation effort is too large.

GROW/PRUNE: Select a variable x_{s_τ} for splitting and split a leaf $\mathcal{X}_\tau$ into two new leaves $\mathcal{X}_{\tau_1}$ and $\mathcal{X}_{\tau_2}$, or collapse two leaves $\mathcal{X}_{\tau_1}$ and $\mathcal{X}_{\tau_2}$ into their common parent $\mathcal{X}_\tau$.

SHIFT: Move the split points ξ^+ and ξ^- of the parent of two leaves.

CHANGE: Split the parent of two leaves by another variable.

SHIFT does not change the number of parameters and is covered by the usual Metropolis-Hastings procedure. CHANGE is the combination of a GROW and a subsequent PRUNE. We demonstrate the algorithm for a GROW/PRUNE step. As the leaf probabilities $\boldsymbol{\theta}_\tau$ do not enter the model selection, the only parameters we have are the index i of the variables to be split and the split points ξ^+ and ξ^-.

We now have the task to define the different quantities in (6). The state $\boldsymbol{w}$ corresponds to a given tree, where $\boldsymbol{w}$ contains the continuous upper and lower split points

for each nonterminal node. The m-th move is applicable only to two specific tree structures and involves the split of a specific leaf $\mathcal{X}_\tau$ of a tree or the pruning of children of $\mathcal{X}_\tau$ in the resulting tree. Note that for each different variable to be split we need a new move type. $j_m(\boldsymbol{w})$ is the probability of selecting move m if the current states is $\boldsymbol{w}$. We define it according to the following lines:

- The split variable is randomly selected with equal probability for each variable.
- In the initial phase GROW/PRUNE, SHIFT and CHANGE are selected with probabilities $1/(3k)$, $1/3$ and $1/3$ respectively. This avoids that the tree grows too fast and a large number of split variables are not explored.
- Each eligible node (leaf to be split, or parent of two leaves to be pruned) is selected with identical probability.

Our prior distribution for the different tree shapes only takes into account the number N of nodes in the tree. It is proportional to $1/(1 + \beta \exp(\gamma N))$ with $\beta, \gamma > 0$ and penalises large trees. The variables used for splitting have equal prior probability to be selected.

As discussed by [3] it is advantageous to let the prior depend on the data $\boldsymbol{X}$ in some aspects. Assume $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(l)}$ are the sorted observed values for some variable x_i in region $\mathcal{X}_\tau$. Then we first assume that both split points ξ^- and ξ^+ are located in the interval $(x_{(1)}, x_{(l)})$. In addition we currently suppose that the overlap covers a fixed proportion ρ of the interval $(x_{(1)}, x_{(l)})$. Therefore ξ^- is has a uniform prior over $(x_{(1)}, x_{max})$ with $x_{max} = x_{(1)} + (1 - \rho)(x_{(l)} - x_{(1)}))$. ξ^+ is defined as $\xi^- + \rho(x_{(l)} - x_{(1)})$.

This allows a simple definition of the quantities in (8). The auxiliary variable $\boldsymbol{u}$ is defined as a univariate random variable with a uniform distribution $p_m(\boldsymbol{u})$ in $(x_{(1)}, x_{max})$. The map $h_m(\boldsymbol{u}, \boldsymbol{w})$ is just the identity, which leaves $\boldsymbol{w}$ unchanged. Hence the Jacobian determinant $J_m(\boldsymbol{u}, \boldsymbol{w}) = 1$. Using these terms we may calculate the ratio of posteriors (8) defining the acceptance probability of a split.

3.5 Generating the Markov Chain

The procedure of generating the Markov chain involves the following steps

1. Randomly select a move m according to probability $j_m(\boldsymbol{w})$. In the case of GROW/PRUNE this involves the random decision whether two leaves are pruned or a split takes place, the random selection which variable is to be split, and the random selection of the leaf or parent node. In addition a lower and an upper split point has to be selected if SPLIT was chosen.
2. Calculate the ratio of posteriors (8).
3. Determine the acceptance probability by (6) and accept the new state or keep the old state.

The algorithm is iterated for some time until the number of nodes in the tree stabilises. Then the resulting tree is saved and a new tree is grown. This yields a set $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_B$ of parameters distributed according to the posterior distribution $p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X})$ and defines

Table 1: β-error for different ensemble methods: Percentage of rejected solvent enterprises, if 8.75% of the insolvent enterprises were accepted as solvent.

Algorithm	Single Model	Sampling Method	Mean
LDA	69.54	bootstrap	49.23
MARS	48.08	bootstrap	39.90
CART	43.82	bootstrap	36.62
BayesTree	42.18	Bayes MCMC	33.72

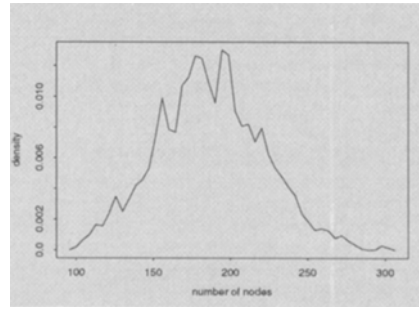

Fig.1 Distribution of tree sizes in an ensemble of Bayes trees.

a representative ensemble of models. We then may estimate, for instance, the expected probability that a new object with features $\boldsymbol{x}$ belongs to class c according to (2) by

$$E(q_c|\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{X}) \approx \frac{1}{B}\sum_{i=1}^{B} p(y{=}c|\boldsymbol{x}, \boldsymbol{w}_b) \tag{8}$$

which converges to (2) by the Law of Large Numbers.

4 Results for Real World Data

We applied our procedure to a dataset containing 6667 records with 73 predictors. The predictors were indicators from balance sheet figures of enterprises. A fraction of 14% of the businesses were classified as 'not solvent' and the rest as 'solvent'. We divided the data randomly into a training set and a validation set. To compare the tree methods with the procedures analysed by [16] we determined a threshold on the validation set such that exactly 8.75% of the insolvent enterprises were wrongly accepted as solvent (α-error) and subsequently measured the percentage of solvent enterprises which were rejected (β-error). In [16] a β-error of 38% on the validation set is reported for a pruned MLP as the best result.

For bootstrapped[11] versions of the algorithms CART [2], LDA, MARS [6], and our Bayesian MCMC algorithm BayesTree, we computed ensembles of $B = 500$ models $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_B$, using the data of the training set. The error percentages were determined from the validation set. The results of our experiments are shown in table 1. In the column 'single model' there is the β-error of a single model estimated by maximum likelihood without resampling. The column 'mean' reports the β-error when the mean of all 500 models is used for classification. The distribution of tree sizes for BayesTree is shown in figure 1.

For BayesTree the β-error on the validation set had a mean of about 33.7%. This shows that the tree procedures are able to beat MLPs in this classification task by about **5%**. In addition the Bayesian trees have a small advantage over the bootstrap trees.

According to (3) the expected loss should be the decision criterion: only if the expected loss is negative (a profit) the loan should be granted. This defines a threshold

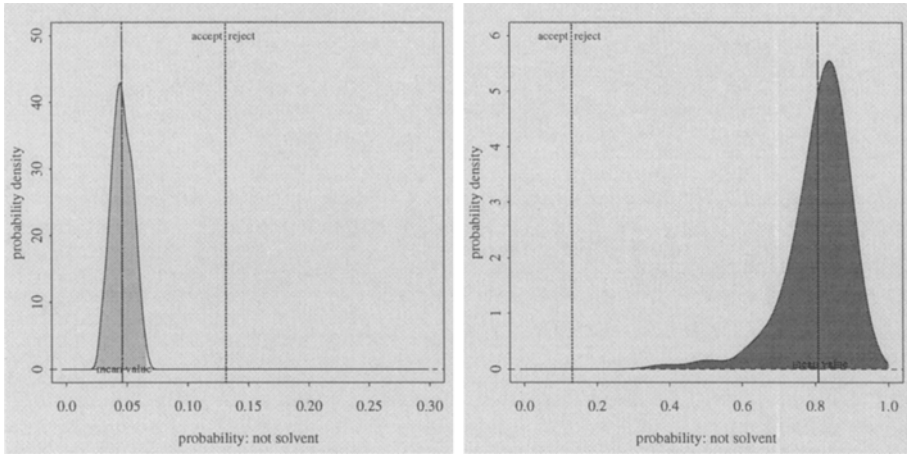

Fig.2 Posterior density of the probability q_{ns} "not solvent" for two previously unseen enterprises. In these cases the decision is clear cut.

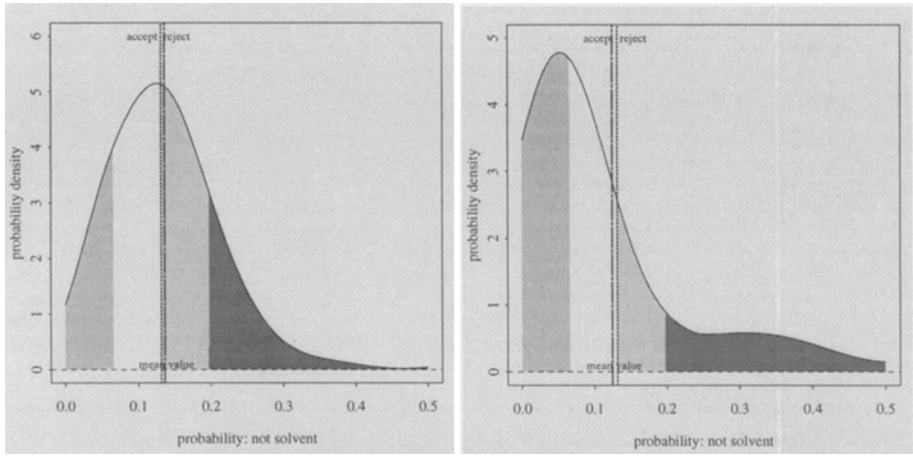

Fig.3 Posterior density of the probability q_{ns} "not solvent" for two previously unseen enterprises.

for the mean value of the predicted risk (probability of insolvency). For a plausible loss function in the figures 2-3 this threshold is depicted as a straight vertical line: If the expected probability of insolvency (vertical dotted line) is above the threshold, we can expect a net loss on the average and the credit application should be rejected.

Figure 2 shows predictive densities for other inputs x, which were determined by (8). In both cases the mean value of the posterior distribution is far away from the decision boundary, and both distributions show low variance. Therefore the left case readily can be accepted and the right case can be rejected. By (3) we may also calculate a distribution of plausible losses. Figure 3 shows two different predictive densities with expectations near the decision boundary. The distribution on the right side has a a larger

uncertainty and hence a much higher chance that the true probability is further away from the decision boundary. This means that the collection of new information (e.g. an audit of the enterprise) has a high chance of revealing that the credit risk is much lower (or much higher) than the borderline. Hence the uncertainty of the credit risk can be used to select cases for a further investigation.

5 Summary

In this paper we developed a Bayesian classification procedure using an ensemble of models which is representative for the distribution of model parameters. The models are allowed to switch between different levels of complexity. This is controlled by a special Metropolis-Hastings algorithm which approximates the desired posterior distribution by the stationary distribution of a Markov chain.

We adapted this procedure to classification trees with overlapping regions. In a real-world credit-scoring task significantly improved results were obtained as compared to both single-model classification and bootstrap methods. The work on Bayes trees is still in progress. So far we use fixed percentages of leaf overlap. This can be generalised to adaptively determine the optimal percentage of overlap for each pair of split points in each leaf of the tree.

References

1. J.O. Berger. *Stat. Dec. Theory, Foundations, Concepts and Methods.* Springer, NY, 1980.
2. L. Breiman, J.H. Friedman, R. Olshen, and C.J. Stone. *Classif. and Regr. Trees.* Wadsworth Int. Group, Belmont, CA, 1984.
3. W. Buntine. Learning classification trees. *Statistics and Computing*, 2:63–73, 1992.
4. C. Carter and J. Catlett. Assessing credit card appl. using machine learning. *IEEE Expert*, 2(3):71–79, 1987.
5. H. Chipman, E. George, and R. McCulloch. Bayesian CART. TR, Dept. of Stat., Univ. of Texas,Austin, 1995.
6. J. H. Friedman. Multivariate adaptive regression splines. *Ann. of Stat.*, 19(1):1–67, 1991.
7. J.H. Friedman. Local learning based on recursive covering. TR, Stanford Uni, August 1996.
8. S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation 4, p.-58*, 1992.
9. P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. TR, Bristol Univ., 1995.
10. R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.
11. Gerhard Paaß. Assessing and improving neural network predictions by the bootstrap algorithm. In S. Hanson, J. Cowan, and C. Giles, editors, *NIPS-5*, pages 196–203. Morgan Kaufmann, San Mateo, CA, San Mateo, CA., 1993.
12. J.R. Quinlan. *C4.5: Prog. f. Machine Learning.* Morgan Kaufmann, San Mateo, CA, 1993.
13. B.D. Ripley. *Pattern Recog. and Neural Networks.* Cambridge Univ. Press, 1996.
14. Waterhouse S.R. *Classification and Regression using Mixtures of Experts.* PhD thesis, Cambridge Univ. Engineering Dept., October 1997.
15. L. Tierney. Markov chains for expl. post. distr. TR 560, School of Stat., UMinnesota, 1994.
16. J. Wallrafen. Kreditwürdigkeitsprüfung von Unternehmen mit neuronalen Klassifikationsverfahren. Master's thesis, University of Erlangen-Nürnberg, 1995.