Computational Statistics and Data Analysis 54 (2010) 1197–1205

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

The Bayesian Additive Classification Tree applied to credit risk modelling

Junni L. Zhang [a,*], Wolfgang K. Härdle [b]

[a] Department of Business Statistics and Econometrics, Guanghua School of Management, Peking University, Beijing 100871, PR China

[b] Center for Applied Statistics and Economics, Wirtschaftswissenschaftliche Fakultät, Humboldt-Universität zu Berlin, Spandauer Straβe 1, 10178, Berlin, Germany

ARTICLE INFO

Article history:
Received 2 January 2009
Received in revised form 14 October 2009
Accepted 26 November 2009
Available online 18 December 2009

Keywords:
Classification and regression tree
Financial ratio
Misclassification rate
Accuracy ratio

ABSTRACT

We propose a new nonlinear classification method based on a Bayesian "sum-of-trees" model, the Bayesian Additive Classification Tree (BACT), which extends the Bayesian Additive Regression Tree (BART) method into the classification context. Like BART, the BACT is a Bayesian nonparametric additive model specified by a prior and a likelihood in which the additive components are trees, and it is fitted by an iterative MCMC algorithm. Each of the trees learns a different part of the underlying function relating the dependent variable to the input variables, but the sum of the trees offers a flexible and robust model. Through several benchmark examples, we show that the BACT shows excellent performance. We apply the BACT technique to classify whether firms would be insolvent. This practical example is very important for banks to construct their risk profile and operate successfully. We use the German Creditreform database and classify the solvency status of German firms based on financial statement information. We show that the BACT is a serious competitor to the logit model, CART, the Support Vector Machine, random forest and gradient boosting.

1. Introduction

Classification techniques have been popularly used in many fields. Standard classification tools include linear and quadratic discriminant analysis and the logistic model. The support vector machine (SVM) (Vapnik, 1995, 1997) has recently emerged as an important nonlinear classification tool. It maps the input space nonlinearly into a high dimensional feature space, and tries to find linear separating hyperplanes for the classes in the feature space, penalizing the distances of misclassified cases to the hyperplanes. The SVM has been widely and successfully applied to classification problems in many domains and often show excellent performance compared to other classification methods.

Decision trees compose an important category of nonlinear classification methods. Ever since the introduction of the classification and regression tree (CART) by Breiman et al. (1984), it has attracted strong interest from researchers and practitioners. Fig. 1 shows an example of a classification tree, where the root node (t_1) contains all training observations, and the training data are recursively partitioned by values of the input variables (x's) until reaching the leaf (terminal) nodes (t_3, t_4, t_6 and t_7) where the classification decision (for y) is made for all observations contained therein. For regression problems in which the dependent variable is continuous, a predicted value for the dependent variable would be assigned for all observations contained in each leaf node.

Traditional search methods for CART models use locally greedy algorithms to find the partitions. The Bayesian approaches for CART models (Chipman et al., 1998; Denison et al., 1998; Wu et al., 2007) specify a formal prior distribution for trees and other parameters and use Markov Chain Monte Carlo methods to sample them from the posterior distribution.

* Corresponding author.
E-mail addresses: zjn@gsm.pku.edu.cn (J.L. Zhang), haerdle@wiwi.hu-berlin.de (W.K. Härdle).

0167-9473/$ – see front matter © 2010 Published by Elsevier B.V.
doi:10.1016/j.csda.2009.11.022

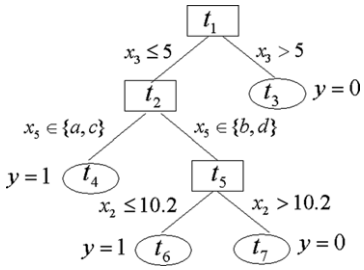

Fig. 1. Example of a classification tree.

The boosted tree model (Freund and Schapire, 1997; Friedman, 2001, 2002) and the random forest model (Breiman, 2001) combine a set of trees to improve model performance. Boosting fits a tree in each step to explain residuals not fitted by previous trees, and the final model is associated with the sum of all trees. Random forest fits a tree at each step with randomly sampled data and predictors, and then average predictions across the trees. Chipman et al. (in press) proposed the Bayesian Additive Regression Tree (BART), a Bayesian model combining a set of trees, in which the mean of a continuous dependent variable is approximated by a sum of trees. This "sum-of-trees" model is defined by a prior and a likelihood, and fitted by iterative MCMC algorithm. Each individual tree explains a different portion of the underlying mean function, but the sum of these trees turns out to be a flexible and adaptive model. Chipman et al. (in press) showed that BART is a serious competitor to LASSO (Efron et al., 2004), gradient boosting (Friedman, 2001), random forests, and neural networks with one layer of hidden units. We will extend BART into the classification context, and therefore term the resulting classification technique as the Bayesian Additive Classification Tree (BACT).

To investigate the differences among the logit model, SVM, CART and BACT, we plot in Figs. 2 and 3 the contours of these models trained to classify the solvency status of German firms using the German Creditreform database based on only two variables — the ratio of operating income to total assets ($x3$ in the figures) and the ratio of accounts payable to total sales ($x24$ in the figures). Details of this application will be discussed in Section 4. The contours for the logit model are linear, thus making it inflexible for complex applications. The SVM finds flexible smooth curves in the input space (linear hyperplanes in the feature space) that can separate the classes. The CART is based on a single tree which recursively partitions the observations by the input variables, and hence the contours are piecewise linear. The BACT is based on the sum of many trees, so the contours are not constrained to be piecewise linear as in CART; although these contours are not as smooth as in SVM, they are quite flexible in explaining complex structure.

The rest of this paper is organized as follows. Section 2 will describe the BACT in detail. Section 3 will use several benchmark examples from the UCI Machine Learning Repository to compare the performance of the BACT with the logit model, the SVM, gradient boosting and random forests. Section 4 will discuss our application to classification of solvency status of Germany firms using the German Creditreform database. Section 5 then concludes.

2. The Bayesian Additive Classification Tree (BACT)

2.1. The model

Consider a binary classification problem in which a dependent variable $Y \in \{1, 0\}$ needs to be predicted based on a set of input variables $\boldsymbol{x} = (x_1, \ldots, x_p)^\top$. The majority of classification models assume that there is a latent continuous variable Y^* that determines the value of Y as follows

$$\begin{cases} Y = 1 & \text{if } Y^* \ge 0 \\ Y = 0 & \text{if } Y^* < 0 \end{cases} \tag{1}$$

In the context of generalized linear models (GLM), the relationship of Y^* and $\boldsymbol{x}$ is

$$Y^* = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon,$$

where the distribution of ε determines the link function, e.g. logit or probit. The generalized additive models (GAM, Hastie and Tibshirani, 1990) replace each linear term in the GLM by a more generalized functional form and relate Y^* to $\boldsymbol{x}$ by

$$Y^* = \beta_0 + f_1(x_1) + \cdots + f_p(x_p) + \varepsilon,$$

where each f_j is an unspecified smooth function.

Following the idea of the BART in Chipman et al. (in press), we assume that Y^* is related to $\boldsymbol{x}$ through an additive model, where each additive component is a tree based on all input variables (rather than a flexible function based on a single input variable as in GAM). In order to formally introduce the model, we first introduce some notation. Let m denote the number of trees to be used. For $j = 1, \ldots, m$, let T_j denote the jth tree with a set of partition rules based on the input variables, and let L_j denote the number of leaf nodes in T_j; for $l = 1, \ldots, L_j$, let μ_{jl} denote the (continuous) predicted value associated with

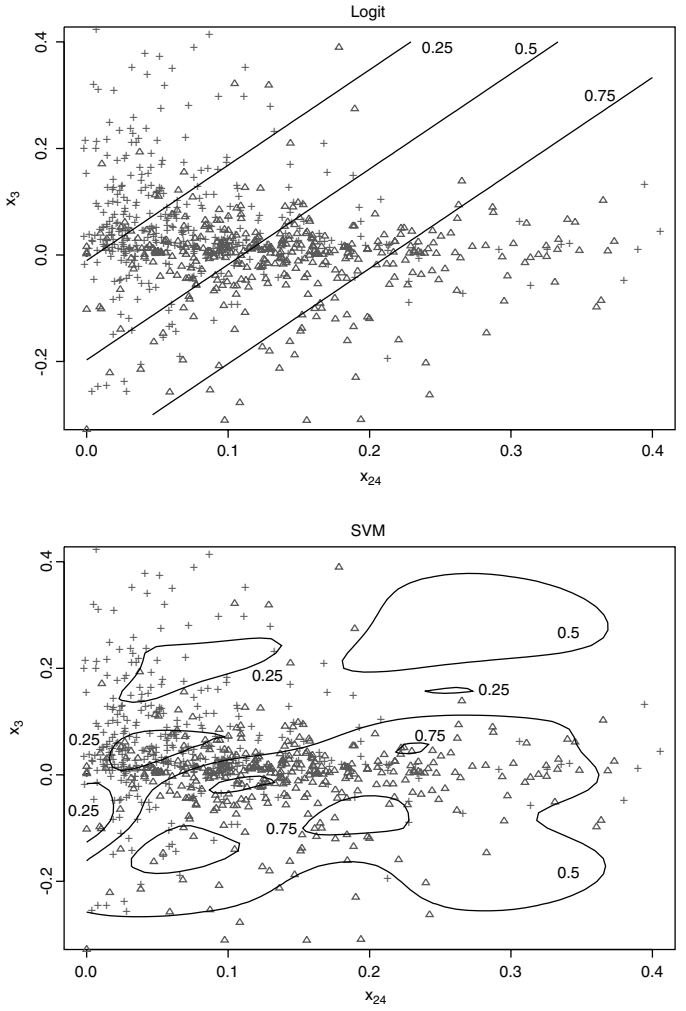

Fig. 2. The contour plots for the logit model and SVM. The triangles and pluses represent insolvent firms and solvent firms respectively. The numbers by the contours indicate the probabilities of insolvency.

the lth leaf node in T_j, and let $M_j = \{\mu_{j1}, \mu_{j2}, \ldots, \mu_{jL_j}\}$. For a given value of $\boldsymbol{x}$, let $g(\boldsymbol{x}, T_j, M_j)$ denote the predicted value associated with the leaf node for an observation with input variables $\boldsymbol{x}$ based on the partition rules for T_j. Thus Y^* is formally modelled as

$$Y^* = g(; T_1, M_1) + g(\boldsymbol{x}; T_2, M_2) + \cdots + g(\boldsymbol{x}; T_m, M_m) + \varepsilon, \tag{2}$$

and we further assume that $\varepsilon \sim N(0, 1)$, using a probit-like link.

2.2. Prior specification

In order to make inferences from the model given by (1) and (2) in a Bayesian way, we should specify a joint prior distribution for the unknown tree structures and leaf nodes parameters. As pointed out by Gelman et al. (1995), typically the prior distribution need not be realistically concentrated around the true value, because often the information about the parameters contained in the data will far outweigh any reasonable prior probability specification. Following Chipman et al. (in press), it is assumed a priori that the tree structures and the leaf node parameters have independent distributions, so the full prior distribution can be written as

$$p\{(T_1, M_1), (T_2, M_2), \ldots, (T_m, M_m)\} = \prod_{j=1}^{m} p(T_j) \prod_{j=1}^{m} \prod_{l=1}^{L_j} p(\mu_{jl}).$$

It is further assumed that every tree follows the same prior distribution, and every μ_{jl} follows the same prior distribution. So the task of prior specification is reduced to specifying the prior distribution for a single tree T and that for a single μ_{jl} parameter.

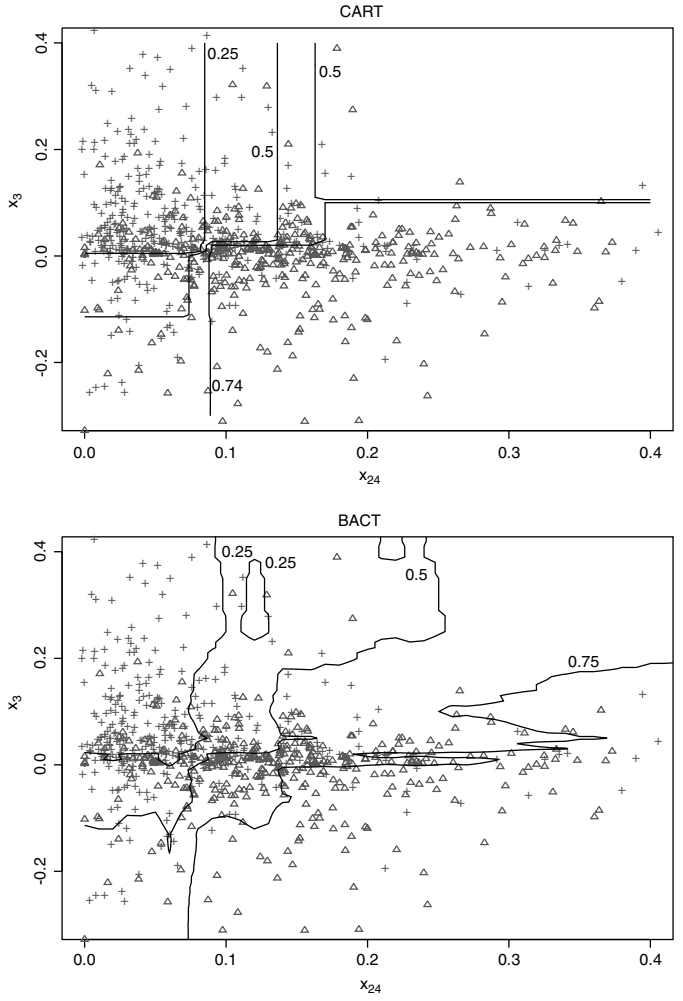

Fig. 3. The contour plots for CART and BACT. The triangles and pluses represent insolvent firms and solvent firms respectively. The numbers by the contours indicate the probabilities of insolvency.

For a single tree T, we need to specify the prior distributions for its partition rules, including whether to further split a node or leave it as a leaf node, and if a further split is needed, which input variable and what values to be used for that split. We use the prior distribution for a single tree T as in Chipman et al. (in press). The prior probability of splitting any node n in tree T is

$$p_{split}(n, T) \propto \alpha(1 + d_n)^{-\beta},$$

where d_n is the depth of node n in tree T (the depth of node n is the length of the path from the root node to node n; e.g., in Fig. 1, the node t_1 has depth 0, and the nodes t_2 and t_3 have depth 1). α and β here are positive hyperparameters, hence the deeper a node is, the smaller probability there is to further split it, or the larger probability that this node becomes a leaf node. It turns out that the performance of BACT is not very sensitive to the choice of α and β. We tried three different settings listed in Table 1 where a priori the trees range from small size to large size, and the resulting performance was quite similar. So we just pick $\alpha = .95$ and $\beta = 2$ as in Chipman et al. (in press). If a node needs to be split, the prior for the associated splitting rules assigns equal probability to each available input variable and equal probability on each available rule given the variable.

The prior distribution of μ_{jl} is taken to be a conjugate normal distribution $\mu_{jl} \sim N(0, \sigma_\mu^2)$ (conjugate because ε in (2) follows a normal distribution). From (2), we can see that the expected value of Y^* is equal to the sum of m different μ_{jl} parameters (recall that $g(\boldsymbol{x}, T_j, M_j)$ is the μ_{jl} parameter associated with the leaf node that an observation with input variables being $\boldsymbol{x}$ would land in based on the partition rules for T_j); because of the a priori independence of μ_{jl}'s, the prior distribution for the expected value of Y^* is $N(0, m\sigma_\mu^2)$. Combining this with (1), it can be inferred that a priori each observation has probability 0.5 belonging to class 1 and probability 0.5 belonging to class 0.

In a recent version of the BART paper (Chipman et al., in press), BART is also extended for classification using the probit link, but the prior distribution there differs from ours in terms of specification for σ_μ^2. Chipman et al. (in press) proposed to

Table 1
Prior distribution on number of terminal nodes based on different values of α and β.

	Setting 1	Setting 2	Setting 3
α	0.5	0.95	0.95
β	2	2	0.1
Prior probability of trees with 1 terminal node	0.5	0.05	0.05
Prior probability of trees with 2 terminal nodes	0.383	0.552	0.012
Prior probability of trees with 3 terminal nodes	0.098	0.275	0.004
Prior probability of trees with 4 terminal nodes	0.017	0.092	0.002
Prior probability of trees with ≥ 5 terminal nodes	0.003	0.031	0.932

use $\sigma_\mu = 3/k\sqrt{m}$ such that $\sum_{j=1}^m g(\boldsymbol{x}; T_j, M_j)$ will with high prior probability be in the interval $(-3, 3)$. We instead use the following procedure.

We first estimate the range of Y^* (to be explained soon), and then choose σ_μ^2 such that there is at least 95% prior probability that the expected value of Y^* is in the estimated range. Let the training data be $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, where N is the number of observations in the training data. We first randomly sample y_i^* for each observation i in the training data from truncated standard normal distributions such that the relationship in (1) holds between y_i^* and the observed y_i. Suppose that the sampled values are $\boldsymbol{y}^{*(0)} = \{y_i^{*(0)}\}_{i=1}^N$, and denote the minimum and maximum values of $y_i^{*(0)}$ as $\min(\boldsymbol{y}^{*(0)})$ and $\max(\boldsymbol{y}^{*(0)})$ respectively. Then $[\min(\boldsymbol{y}^{*(0)}), \max(\boldsymbol{y}^{*(0)})]$ is a very rough estimate of the range of Y^*. We choose an initial $\sigma_\mu^{2(0)}$ such that there is at least 95% prior probability that the expected value of Y^* is in this interval, i.e., $\left[-2\sqrt{m\sigma_\mu^{2(0)}}, 2\sqrt{m\sigma_\mu^{2(0)}}\right]$ covers $\left[\min(\boldsymbol{y}^{*(0)}), \max(\boldsymbol{y}^{*(0)})\right]$ and therefore $\sigma_\mu^{2(0)} = \left\{\max\left[-\min(\boldsymbol{y}^{*(0)}), \max(\boldsymbol{y}^{*(0)})\right]\right\}^2/4m$.

We then run the Markov Chain Monte Carlo (MCMC) algorithm to be described in Section 2.3 to generate posterior samples of y_i^*, and suppose that we obtain one posterior draw of $\boldsymbol{y}^{*(1)} = \{y_i^{*(1)}\}_{i=1}^N$ after dropping the first B_1 posterior draws used to reach convergence. We assume this set of y_i^* can be used to estimate reasonably the range of the true underlying Y^*, and choose the value of σ_μ^2 for further analysis such that there is at least 95% prior probability that the expected value of Y^* is in the interval $[\min(\boldsymbol{y}^{*(1)}), \max(\boldsymbol{y}^{*(1)})]$, i.e., $\sigma_\mu^2 = \left\{\max\left[-\min(\boldsymbol{y}^{*(1)}), \max(\boldsymbol{y}^{*(1)})\right]\right\}^2/4m$.

2.3. Generation of posterior samples and inference

We use the data augmentation method (Tanner and Wong, 1987) by treating $\boldsymbol{y}^* = \{y_i^*\}_{i=1}^N$ as missing data, and then use the Gibbs sampler to generate samples from the posterior distribution $p\{(T_1, M_1), (T_2, M_2), \ldots, (T_m, M_m), \boldsymbol{y}^*|\mathcal{D}\}$.

Let $T_{(j)}$ denote the $m-1$ trees other than T_j, and let $M_{(j)}$ denote the parameters associated with the leaf nodes in $T_{(j)}$. The Gibbs sampler composes of drawing m successive draws of (T_j, M_j) for $j = 1, \ldots, m$ from $p\{(T_j, M_j)|T_{(j)}, M_{(j)}, \boldsymbol{y}^*, \mathcal{D}\}$ followed by draws of $\boldsymbol{y}^*$ from $p\{\boldsymbol{y}^*|(T_1, M_1), (T_2, M_2), \ldots, (T_m, M_m), \mathcal{D}\}$. The draws of (T_j, M_j) can be generated similar to Chipman et al. (in press). Let $\hat{y}_i^* = \sum_{j=1}^m g(\boldsymbol{x}_i; T_j, M_j)$ denote the fitted value for observation i from the m trees. Then y_i^* $(i = 1, \ldots, N)$ can be independently generated from truncated normal distributions:

$$\begin{cases} y_i^* \sim N(\hat{y}_i^*, 1) & \text{and} \quad y_i^* \geq 0 \quad \text{if } y_i = 1 \\ y_i^* \sim N(\hat{y}_i^*, 1) & \text{and} \quad y_i^* < 0 \quad \text{if } y_i = 0. \end{cases}$$

After σ_μ^2 has been chosen according to the procedure described in Section 2.2, we can drop the first B_2 posterior draws used to reach convergence, and use subsequent S posterior draws for inference. Denote these S posterior draws as $\{(T_1^{(s)}, M_1^{(s)}), \ldots, (T_m^{(s)}, M_m^{(s)})\}_{s=1}^S$. Given the sth draw, the probability that an observation with input variables $\boldsymbol{x}$ belongs to class 1 is $\Phi\left\{\sum_{j=1}^m g(\boldsymbol{x}, T_j^{(s)}, M_j^{(s)})\right\}$, where Φ is the cumulative distribution function of standard normal distribution. Therefore, the posterior average probability that an observation with input variables $\boldsymbol{x}$ belongs to class 1 can be estimated as

$$\frac{1}{S}\sum_{s=1}^S \Phi\left\{\sum_{j=1}^m g(\boldsymbol{x}, T_j^{(s)}, M_j^{(s)})\right\}. \tag{3}$$

We can use (3) to classify observations in training data or other data: if the probability calculated from (3) is larger than 0.5, then the observation is classified into class 1; otherwise it is classified into class 0.

3. Benchmark examples

We will compare the performance of the BACT with the logit model, SVM (in which the radial basis function is used as the kernel, and the parameters are chosen by cross-validation), random forest (implemented as **randomForest** in R) and gradient boosting (implemented as **gbm** in R, where the following options are used: AdaBoost loss function, 10,000

Table 2
For five benchmark data sets from the UCI Machine Learning Repository, the number of cases, the number of variables, and the average misclassification rates for the test data using the logit model, the SVM, random forest, gradient boosting and the BACT.

Data set	# Cases	# Variables	Logit (%)	SVM (%)	Random forest (%)	Gradient boosting (%)	BACT (%)
Breast cancer	683	9	3.8	2.8	2.4	3.1	3.3
Ionosphere	351	34	12.8	4.5	6.6	6.8	7.2
Diabetes	768	8	21.8	25.2	24.2	24.3	24.8
Sonar	208	60	29.8	19.4	16.4	19.2	17.2
German credit	1000	30	23.6	27.3	24.2	23.4	23.6

trees each having 3 terminal nodes, the shrinkage parameter being 0.001, and 5-fold cross-validation for choosing the best iteration). We use five data sets for binary classification from the UCI Machine Learning Repository (Asuncion and Newman, 2007): breast cancer, ionosphere, diabetes, sonar, and German credit. Columns 2–3 in Table 2 summarize the number of cases and the number of variables for these data sets. Throughout the rest of the paper, in the BACT method, we fix $m = 200$, $B_1 = 500$, $B_2 = 1000$ and $S = 1000$.

We partition each data set randomly into 80% of training data and 20% of test data. The training data is used to fit the models, and misclassification rate on the test data is calculated. This procedure is repeated 20 times, and columns 4–6 in Table 2 report the average misclassification rates on the test data using the logit model, SVM, random forest, gradient boosting and the BACT. For these benchmark examples, the BACT has comparable performance with SVM, random forest and gradient boosting; the logit model performs way worse than the other models for the "ionosphere" and "sonar" data sets, but better than the other models for the "diabetes" data set.

4. Classification of solvency status of German firms

We use the German Creditreform database, which contains financial statement information on 20,000 solvent and 1000 insolvent firms in Germany and spans the period from 1996 to 2002. Information on the insolvent firms were collected two years prior to insolvency. Chen et al. (in press) applied SVM to the German Creditreform database to classify the solvency status of German firms. We will preprocess the data set in the same way as they do, and compare the results of the BACT with those of the logit model and CART. We also compare our results with those of SVM given by Chen et al. (in press).

Following Chen et al. (in press), we clean the data of firms whose characteristics are very different from the others. We first eliminate firms within industries with small percentage in the industry composition and are left with 949 insolvent firms and 16,583 solvent firms in four main industries—Construction, Manufacturing, Wholesale & Retail Trade and Real Estate. We then exclude those firms whose asset size is less than 10^5 EUR or greater than 10^8 EUR, because the credit quality of small firms often depends as much on the finances of a key individual as on the firm itself and largest firms rarely go bankrupt in Germany. We further exclude the solvent firms in 1996 due to lack of insolvent firms in that year. We also eliminate firms with zero value for some variables used as denominators in calculating financial ratios to be used in classification. Several apparent outliers are then deleted and we end up with a data set with 783 insolvent firms and 9575 solvent firms (due to slightly different ways of deleting outliers, our remaining solvent firms differ a little from the 9583 solvent firms in Chen et al. (in press)).

We adopt the same set of financial variables to be used for classification as in Chen et al. (in press) and list them in Table 3. The five number summary of these financial variables are listed in Table 4 for insolvent firms and solvent firms separately. In order to avoid sensitivity to outliers in applying the SVM, Chen et al. (in press) truncated each financial variable to be between its 5% quantile and 95% quantile. The BACT, however, only uses the ordering of values of the input variables in the partition rules, so there is no need to do such truncation.

We use the data from 1997 to 1999 to train the model, and use the data from 2000 to 2002 to test the resulting model. The training set contains 387 insolvent firms and 3535 solvent firms, and the test set contains 396 insolvent firms and 6040 solvent firms. Because the density of insolvent firms is rather low, we need to oversample the insolvent firms in order for the models to pick up the patterns predictive of insolvency (e.g., Berry and Linoff, 2000, chap. 5). This is done through the bootstrap technique (Efron and Tibshirani, 1993; Sobehart et al., 2001). For each bootstrap sample, a training subset is constructed as follows. We use all 387 insolvent firms in the training set and randomly sample 387 solvent firms from the training set. This subset of 774 firm with 50% being insolvent is then used to train the model. When training the CART model, the training subset is further randomly partitioned into two parts stratified by the solvency status of the firms. The first part comprises of 80% of the training subset and is used to grow the tree, and the second part comprises of the remaining 20% of the training subset and is used to prune the tree. Performance measures are then evaluated using all observations (396 insolvent firms and 6040 solvent firms) in the test set. The average performance measures over 30 bootstrap samples are then calculated. We can compare average performance measures across different models.

We consider two performance measures: Accuracy Ratio (AR) (Sobehart and Keenan, 2001; Engelmann et al., 2003) and misclassification rate. AR is calculated using the Cumulative Accuracy Profiles (CAP) (Sobehart and Keenan, 2001; Engelmann et al., 2003) curve. To obtain the CAP curve, the firms are first ordered by risk scores from riskiest to safest. For the Logit model, CART, random forest and BACT, the risk score is simply the predicted probability of insolvency; for gradient boosting, the risk score is the value for the sum of trees; for SVM, the risk score can be calculated as distance to the separating

Table 3
Definition of financial variables to be used for classification for the Creditreform data.

Var.	Definition
$x1$	Net income/total assets
$x2$	Net income/total sales
$x3$	Operating income/total assets
$x4$	Operating income/total sales
$x5$	Earnings before interest and tax/total assets
$x6$	Earnings before interest, Tax, Depreciation and amortization/total assets
$x7$	Earnings before interest and tax/total sales
$x8$	Own funds/total assets
$x9$	(Own funds − intangible assets) /(total assets − intangible assets − cash and cash equivalents − lands and buildings)
$x10$	Current liabilities/total assets
$x11$	(Current liabilities − cash and cash equivalents)/total assets
$x12$	Total liabilities/total assets
$x13$	Debt/total assets
$x14$	Earnings before interest and tax/interest expense
$x15$	Cash and cash equivalents/total assets
$x16$	Cash and cash equivalents/current liabilities
$x17$	(Cash and cash equivalents − inventories)/current liabilities
$x18$	Current assets/current liabilities
$x19$	(Current assets − current liabilities)/total assets
$x20$	Current liabilities/total liabilities
$x21$	Total assets/total sales
$x22$	Inventories/total sales
$x23$	Accounts receivable/total sales
$x24$	Accounts payable/total sales
$x25$	log(total assets)
$x26$	Increase (decrease) in inventories/inventories
$x27$	Increase (decrease) in liabilities/total Liabilities
$x28$	Increase (decrease) in cash flow/cash and cash equivalents

Table 4
Five number summary (minimum, lower quartile, median, upper quartile, maximum) of the financial variables for insolvent firms and solvent firms.

Var.	Insolvent firms					Solvent firms				
	min	Q1	mdn.	Q3	max	min	Q1	mdn.	Q3	max
$x1$	−1.51	−0.02	0.00	0.02	1.13	−4.82	0.00	0.02	0.06	5.92
$x2$	−5.41	−0.02	0.00	0.01	6.10	−17.13	0.00	0.01	0.03	15.91
$x3$	−0.97	−0.04	0.00	0.03	1.14	−4.82	0.00	0.03	0.09	5.97
$x4$	−3.38	−0.02	0.00	0.02	10.15	−44.81	0.00	0.02	0.04	20.39
$x5$	−0.99	−0.01	0.02	0.05	1.15	−1.51	0.02	0.05	0.11	5.95
$x6$	−0.91	0.03	0.07	0.11	1.17	−1.46	0.06	0.11	0.18	5.95
$x7$	−3.55	−0.01	0.01	0.04	10.27	−39.63	0.01	0.02	0.05	14.53
$x8$	0.00	0.00	0.05	0.14	0.96	0.00	0.05	0.14	0.28	0.99
$x9$	−0.86	0.00	0.05	0.17	2.31	−2.68	0.05	0.16	0.37	49.18
$x10$	0.01	0.37	0.52	0.73	1.00	0.00	0.25	0.42	0.64	4.13
$x11$	−0.35	0.33	0.49	0.69	0.99	−0.86	0.17	0.36	0.58	4.12
$x12$	0.01	0.54	0.76	0.89	1.00	0.00	0.42	0.65	0.82	4.37
$x13$	0.00	0.09	0.21	0.37	0.91	0.00	0.02	0.15	0.33	0.98
$x14$	−17658.06	−0.56	1.05	1.92	433.40	−22796.04	0.86	2.16	6.55	516896.73
$x15$	0.00	0.00	0.02	0.06	0.44	0.00	0.01	0.03	0.11	0.90
$x16$	0.00	0.01	0.03	0.12	25.01	0.00	0.01	0.08	0.30	40.61
$x17$	0.01	0.43	0.68	0.97	57.44	0.00	0.59	0.94	1.58	238.37
$x18$	0.03	1.00	1.26	1.84	62.63	0.06	1.11	1.58	2.67	989.76
$x19$	−0.69	0.00	0.15	0.36	0.92	−3.45	0.06	0.25	0.47	0.98
$x20$	0.07	0.62	0.84	0.99	1.18	0.01	0.56	0.85	1.00	1.00
$x21$	0.07	0.40	0.61	0.94	97.26	0.02	0.32	0.48	0.74	828.76
$x22$	0.00	0.08	0.16	0.34	89.96	−0.14	0.05	0.11	0.21	451.09
$x23$	0.00	0.07	0.12	0.18	0.87	0.00	0.05	0.09	0.14	21.85
$x24$	0.00	0.09	0.14	0.19	43.96	0.00	0.04	0.07	0.11	61.29
$x25$	11.72	14.07	14.87	15.76	18.25	11.51	14.25	15.41	16.62	18.42
$x26$	−46.89	−0.09	0.00	0.26	2.83	−282.51	−0.01	0.00	0.06	145.12
$x27$	−12.75	−0.04	0.00	0.11	1.00	−28.91	−0.04	0.00	0.10	1.00
$x28$	−1283.20	−0.61	0.00	0.18	1.00	−2513.39	−0.27	0.00	0.26	1.75

hyperplane. The higher the risk score is, the riskier the firm is. For a given fraction q of the total number of firms, the CAP curve is constructed by calculating the fraction $r(q)$ of the insolvent firms whose risk scores are equal to or larger than the minimum score at fraction q.

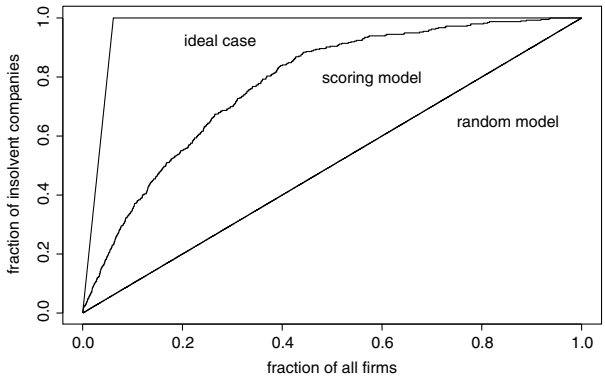

Fig. 4. The CAP curve for the test set of the Creditreform data where the scoring model is the BACT model trained using one bootstrap training subset.

Table 5
The average values of AR and the three types of misclassification rates for the Logit model, CART, random forest, gradient boosting and BACT.

Performance measure	Logit (%)	CART (%)	Random forest (%)	Gradient boosting (%)	BACT (%)
AR	52.1	58.7	58.6	61.0	60.4
Overall misclassification rate	30.2	33.8	27.4	26.7	26.6
Type I Misclassification rate	28.3	27.2	26.9	26.8	27.6
Type II Misclassification rate	30.3	34.3	27.5	26.7	26.5

Fig. 4 plots the CAP curve for the test set of the Creditreform data where the scoring model is the BACT model trained using one bootstrap training subset. In the ideal case, the insolvent firms will be assigned the highest risk scores, and therefore the CAP curve would be increasing linearly and then stay at one. For a random model without any discriminative power, the fraction q of all firms with the highest risk scores will contain fraction q of all insolvent firms, and therefore the corresponding CAP curve will be a straight line connecting the points (0, 0) and (1, 1). AR is defined as the ratio of the area between the CAP curve for a scoring model and that for the random model to the area between the CAP curve for the ideal case and that for the random model. The value of AR lies between zero and one, with zero indicating no discriminative power of the scoring model and one indicating perfect discriminative power. Mathematically, AR is defined as

$$AR \equiv \frac{\int_0^1 r_{model}(q)\mathrm{d}q - \frac{1}{2}}{\int_0^1 r_{ideal}(q)\mathrm{d}q - \frac{1}{2}}, \tag{4}$$

where $r_{model}(q)$ and $r_{ideal}(q)$ indicate $r(q)$ for the scoring model and the ideal case respectively, and the integrals can be approximated by $\frac{1}{N}\sum_{i=1}^{N} r(i/N)$ where N is the number of observations in the test set.

We also consider three types of misclassification rates: the overall misclassification rate, the type I misclassification rate and type II misclassification rate. Here type I misclassification refers to the case when the firm is in fact insolvent, but the model classifies the firm as solvent; whereas type II misclassification refers to the case when the firm is in fact solvent, but the model classifies the firm as insolvent. Financial institutions usually seek to keep either type of misclassification rate as low as possible (Sobehart et al., 2001).

Table 5 reports the average values of AR in (4) and the three types of misclassification rates for the Logit model, CART, random forest, gradient boosting and BACT. In this example, BACT performs better than the logit model, CART and random forest in almost all aspects except for average Type I misclassification rate for which BACT is worse than CART and random forest; BACT performs worse than gradient boosting in terms of AR and average type I misclassification rate, but better in terms of overall misclassification rate and average type II misclassification rate.

Rather than using all data from 2000 to 2002 as the test set, Chen et al. (in press) used a test subset for each bootstrap sample, which comprises of all insolvent firms and a random sample of the same number of solvent firms in the test set. They reported that the median AR value for 30 bootstrap samples was 60.5%, using $\frac{1}{10}\sum_{i=1}^{10} r(i/10)$ to approximate the integrals in calculating the AR value. The median overall misclassification rate was calculated as 28.2%. If we adopt the same procedure, BACT yields a median AR value of 66.5% and median overall classification rate as 27.2%. So in this example BACT performs better than SVM in identifying the insolvent firms.

5. Concluding remarks

In this paper, we propose the Bayesian Additive Classification Tree as a general nonlinear classification method. We show that, based on the sum of many trees, the BACT can yield flexible class boundaries, and that it is a serious competitor to the logit model, CART, SVM, random forest and gradient boosting, as demonstrated through several benchmark examples and a real application to credit risk modelling.

The BACT is robust to extreme values in the input variables and the results do not change with monotone transformation of any input variable, because the partitions in each tree depend only on the ordering of the values of the input variables rather than the values themselves. Hence little data processing is needed when using the BACT technique.

As other Bayesian methods, the use of BACT would require setting up a prior, implementing an iterative MCMC fitting algorithm and carrying out computation that could spend more time than the logit model, CART, SVM, random forest, gradient boosting or some other non-Bayesian classification methods. Such cost is worth spending in cases when BACT can yield better classification performance and classification performance is more important than implementation and computation time.

Although we only discuss binary classification in this paper, extension to multiclass classification is straightforward. For ordinal classification, we can still assume that there is a latent continuous variable Y^* as in the binary case, but there are multiple cut points of Y^* that determine the value of Y. For nominal classification with K categories, we can assume that there are K latent continuous variables $(Y_1^*, \ldots, Y_K^*)$, each being modelled by a sum of trees, and that Y is equal to the category k with the largest value of Y_k^*. We have carried out some preliminary studies and have found that BACT works pretty well in the multiclass (both ordinal and nominal) context. This line of research will be carried out in future paper.

Acknowledgements

This work was supported by the Deutsche Forschungsgemeinschaft through the SFB 649 "Economic Risk". Junni L. Zhang's research was also sponsored by the Chinese NSF grant 10401003 and USA NIH 1 R03 TW007197-01A2.

References

Asuncion, A., Newman, D., 2007. UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences (http://www.ics.uci.edu/mlearn/MLRepository.html).
Berry, M., Linoff, G., 2000. Mastering Data Mining. John Wiley and Sons.
Breiman, L., 2001. Random forests. Machine Learning 5–32.
Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees. CRC Press.
Chen, S., Härdle, W.K., Moro, R.A., 2009. Modeling default risk with support vector machines. Quantitative Finance (in press).
Chipman, H.A., George, E.I., McCulloch, R.E., 1998. Bayesian CART model search. Journal of the American Statistical Association 935–948.
Chipman, H.A., George, E.I., McCulloch, R.E., 2010. BART: Bayesian Additive Regression Trees. Annals of Applied Statistics (in press).
Denison, D., Mallick, B., Smith, A., 1998. A bayesian CART algorithm. Biometrika 363–377.
Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., 2004. Least angle regression. Annals of Statistics 407–499.
Efron, B., Tibshirani, R.J., 1993. An Introduction to the Bootstrap. Chapman and Hall.
Engelmann, B., Hayden, E., Tasche, D., 2003. Testing rating accuracy. Risk 82–86.
Freund, Y., Schapire, R., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 119–139.
Friedman, J.H., 2001. Greedy function approximation: A gradient boosting machine. Annals of Statistics 1189–1232.
Friedman, J.H., 2002. Stochastic gradient boosting. Computational Statistics and Data Analysis 367–378.
Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B., 1995. Bayesian Data Analysis. Chapman and Hall.
Hastie, T.J., Tibshirani, R.J., 1990. Generalized Additive Models. Chapman and Hall.
Sobehart, J., Keenan, S., 2001. Measuring default risk accurately. Risk, Credit Risk Special Report 14, 31–33.
Sobehart, J., Keenan, S., Stein, R., 2001. Benchmarking quantitative default risk models: A validation methodology. Algo Research Quarterly 4 (1/2), 57–72.
Tanner, M.A., Wong, W.H., 1987. The calculation of posterior distributions by data augmentation (with discussion). Journal of the American Statistical Association 528–550.
Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer, New York, NY.
Vapnik, V., 1997. Statistical Learning Theory. Wiley, New York, NY.
Wu, Y., Tjelmeland, H., West, M., 2007. Bayesian CART: Prior specification and posterior simulation. Journal of Computational and Graphical Statistics 16 (1), 44–66.