The Korean Journal of Applied Statistics (2011) DOI: http://dx.doi.org/10.5351/KJAS.2011.24.4.587
24*(4), 587–595*

Generalized Partially Linear Additive Models for Credit Scoring

Ju-Hyun Shim[1] · Young K. Lee[2]

[1]Department of Statistics, Seoul National University
[2]Department of Statistics, Kangwon National University

(Received May 2011; accepted July 2011)

Abstract

Credit scoring is an objective and automatic system to assess the credit risk of each customer. The logistic regression model is one of the popular methods of credit scoring to predict the default probability; however, it may not detect possible nonlinear features of predictors despite the advantages of interpretability and low computation cost. In this paper, we propose to use a generalized partially linear model as an alternative to logistic regression. We also introduce modern ensemble technologies such as bagging, boosting and random forests. We compare these methods via a simulation study and illustrate them through a German credit dataset.

Keywords: Logistic regression, generalized partially linear additive model, bagging, logitboost, random forests.

1. Introduction

A major concern of credit card companies and private banks is how to evaluate each customer's credit risk. As the demands for loan and industrial competition increase, an objectively and automatically operated risk assessment tool is required. Credit scoring is used to assess customers' credit risks by predicting the likelihood of a customer default in the near future. It is mainly used to make a decision or establish strategies for credit-related companies, see Mays (2001) and Thomas *et al.* (2002) for various applications of credit scoring. There are two types of decisions. The first is for customers who apply for a credit loan or a credit card; companies determine who they approve for credit, how much credit they give, and the appropriate interest rate. The second is for existing customers; companies select profitable customers to increase credit limits or renew the loan contract. These are called application score(AS) and behavioral score(BS), respectively. In addition there are many credit scores in AS and BS. Although the purposes, available information and strategies of different credit scores are different, the main focus of credit scoring is a two-class classification problem.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2010-0021396).

[2]Corresponding author: Assistant professor, Department of Statistics, Kangwon National University, Hyoja 2 Dong, Chuncheon 200-701, Korea. E-mail: youngklee@kangwon.ac.kr

Many statistical models or machine learning methods are used to classify between good and bad customers. Logistic regression is most popular for credit scoring since it has advantages in interpretation and computing costs; however, it cannot detect a nonlinear relationship between the response variable and predictors because it is based on the assumption that the response variable is related to the linear combination of predictors. Although there are several methods that can detect the nonlinear relationship, such as classification and regression trees(CART) and neural network, they typically have poor interpretation ability and sometimes poor performance compared to logistic regression, see Breiman (1994) for other disadvantages of these methods.

In this paper we consider using a generalized partially linear additive model to improve performance in terms of credit scoring. There are several ways of fitting the model. One is to apply ordinary backfitting with profiling. The ordinary backfitting technique was proposed by Buja *et al.* (1989) as a way of fitting a nonparametric additive model. However, its theoretical properties when applied to a generalized partially linear additive model is unknown. The smooth backfitting technique proposed by Mammen *et al.* (1999) is known as a powerful technique for fitting structured nonparametric models. Yu and Lee (2010) studied smooth backfitting with profiling as a way of fitting the model under study; however, its practical implementation is computationally quite expensive. In addition, Yu and Lee (2010) failed to give numerical properties of the method. In this paper we suggest to use the regression spline technique. In a simulation and a real data analysis, we compare the proposed method with logistic regression, CART, and also some ensemble methods such as bagging, boosting and random forests.

In the next section we introduce the generalized partially linear additive model and the regression spline method to fit the model. We also discuss some properties of the estimators of the model parameters. In Section 3, we review the three ensemble methods briefly. In Section 4, we give the results of the simulation study and the real data analysis. We finish the paper by giving some concluding remarks in Section 5.

2. Generalized Partially Linear Additive Model

A fully parametric regression model is too restrictive to accommodate various complicated relations between the response and predictors. It easily fails when the true model is far from the assumed model. A fully nonparametric regression model is flexible; however, its usage is sometimes limited, particularly in the case where there are qualitative predictors. A semiparametric model may be a good compromise between the two extremes. With semiparametric modeling, one may put qualitative predictors (or those whose effects on the response are believed to be linear) into a parametric part, and the others into a nonparametric part. The generalized partially linear additive model(GPLAM) is such a model.

With the *i.i.d.* observations $(Y^i, \mathbf{X}^i, \mathbf{Z}^i), \ldots, (Y^n, \mathbf{X}^n, \mathbf{Z}^n)$ of a random vector $(Y, \mathbf{X}, \mathbf{Z})$ where $\mathbf{X} = (X_1, \ldots, X_p)^\top \in \mathbb{R}^p$ and $\mathbf{Z} = (Z_1, \ldots, Z_d)^\top \in \mathbb{R}^d$, GPLAM assumes

$$E\left(Y^i | \mathbf{X}^i = \mathbf{x}, \mathbf{Z}^i = \mathbf{z}\right) = g^{-1}\left(\boldsymbol{\beta}^\top \mathbf{x} + \sum_{j=1}^{d} m_j(z_j)\right) \tag{2.1}$$

for a link function g which is strictly increasing. Here, $\boldsymbol{\beta}$ is the p-vector of unknown parameters and $m_j(\cdot)$'s are unknown functions such that $Em_j(Z_j) = 0$ for $j = 1, \ldots, d$. The link function g allows one to apply the model to a discrete response variable as well. For a continuous response Y, one may use the identity link $g(u) = u$, and in that case the model (2.1) reduces to the partially linear additive model. Two important examples of discrete response variables are $Y \in \{0,1\}$ and $Y \in \{0,1,2,\ldots\}$. In those cases, one typically uses the logit link $g(u) = \log(u/(1-u))$ and the log link $g(u) = \log(u)$, respectively.

We consider the quasi-likelihood approach (Severini and Staniswalis, 1994) to estimate the parametric part $\boldsymbol{\beta}$ and the nonparametric additive function $m(\mathbf{z}) = m_1(z_1) + \cdots + m_d(z_d)$. It requires a modeling of the conditional variance $v(\mathbf{x},\mathbf{z}) = \mathrm{var}(Y|\mathbf{X}=\mathbf{x},\mathbf{Z}=\mathbf{z})$ in terms of the conditional mean $f(\mathbf{x},\mathbf{z}) \equiv E(Y|\mathbf{X}=\mathbf{x},\mathbf{Z}=\mathbf{z})$. If one assumes $v(\mathbf{x},\mathbf{z}) \equiv V(f(\mathbf{x},\mathbf{z})$ for a known function V, then the quasi-likelihood function $Q(\mu,y)$ is defined by

$$\frac{\partial}{\partial\mu}Q(\mu,y) = \frac{y-\mu}{V(\mu)}, \tag{2.2}$$

and the quasi-likelihood is given by

$$L_n(\boldsymbol{\beta},m) = n^{-1}\sum_{i=1}^{n} Q\left(g^{-1}\left(\boldsymbol{\beta}^\top\mathbf{X}^i + m\left(\mathbf{Z}^i\right)\right), Y^i\right). \tag{2.3}$$

The case where the conditional distribution of Y given $\mathbf{X}=\mathbf{x}$, $\mathbf{Z}=\mathbf{z}$ belongs to an exponential family may be put into the quasi-likelihood framework. If the conditional distribution has a density of the form

$$\mathrm{pdf}_{Y|\mathbf{X},\mathbf{Z}}(y|\mathbf{x},\mathbf{z}) = \exp\left[a(\phi)^{-1}\left\{y\theta(\mathbf{x},\mathbf{z}) - b(\theta(\mathbf{x},\mathbf{z}))\right\} + c(y,\phi)\right]$$

for some known functions $a(\cdot)$, $b(\cdot)$, $c(\cdot,\cdot)$, then $f(\mathbf{x},\mathbf{z}) = b'(\theta(\mathbf{x},\mathbf{z}))$ and $v(\mathbf{x},\mathbf{z}) = a(\phi)b''(\theta(\mathbf{x},\mathbf{z})) = a(\phi)b'' \circ (b')^{-1}(f(\mathbf{x},\mathbf{z}))$. If we let $V(\mu) = a(\phi)b'' \circ (b')^{-1}(\mu)$, then $Q(\mu,y) = a(\phi)^{-1}\{y(b')^{-1}(\mu) - b \circ (b')^{-1}(\mu)\}$ satisfies (2.2). If we take the canonical link $g = (b')^{-1}$, then the likelihood function at (2.3) reduces to

$$L_n(\boldsymbol{\beta},m) = n^{-1}\sum_{i=1}^{n} Y^i\left(\boldsymbol{\beta}^\top\mathbf{X}^i + m\left(\mathbf{Z}^i\right)\right) - b\left(\boldsymbol{\beta}^\top\mathbf{X}^i + m\left(\mathbf{Z}^i\right)\right)$$

up to a constant factor.

To estimate the nonparametric additive function $m(\mathbf{z})$, we apply the regression spline technique. Yu *et al.* (2008), and Yu and Lee (2010), respectively, discussed fitting GAM(generalized additive models) and by kernel smoothing. Although the kernel smoothing techniques of fitting GAM and GPLAM have very nice theoretical properties, they are known to be computationally expensive in a high-dimension.

Given a fixed knot sequence $\xi_1,\ldots,\xi_M$, a function is called a *polynomial spline of order* q if it is a piecewise polynomial of order q on each of the intervals,

$$(-\infty,\xi_1],[\xi_1,\xi_2],\ldots,[\xi_{M-1},\xi_M],[\xi_M,\infty),$$

and has $(q-1)$ continuous derivatives at the knots. Let $B_{j,k}$ for $1 \le k \le M+q+1$ be the B-spline basis functions for the component function m_j, and put $\mathbf{B}_j = (B_{j,1},\ldots,B_{j,M+q+1})^\top$. A function μ_j in the space of polynomial spline of order q can be represented by $\mu_j(z_j) = \sum_{k=1}^{M+q+1}\gamma_{j,k}B_{j,k}(z_j) = \boldsymbol{\gamma}_j^\top\mathbf{B}_j(z_j)$ for some constant vector $\boldsymbol{\gamma}_j$. Plugging this expression into the quasi-likelihood at (2.3) gives

$$\ell(\boldsymbol{\beta},\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_d) = n^{-1}\sum_{i=1}^{n} Q\left(g^{-1}\left(\boldsymbol{\beta}^\top\mathbf{X}^i + \sum_{j=1}^{d}\boldsymbol{\gamma}_j^\top\mathbf{B}_j\left(Z_j^i\right)\right), Y^i\right). \tag{2.4}$$

We maximize the quasi-likelihood $\ell(\boldsymbol{\beta}, \boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_d)$ at (2.4) with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_d$. If we denote the maximizer of $\ell(\boldsymbol{\beta}, \boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_d)$ by $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\gamma}}_j$, then estimators of m_j after the normalization are given by

$$\hat{m}_j(x_j) = \hat{\boldsymbol{\gamma}}_j^\top \mathbf{B}_j(z_j) - n^{-1} \sum_{i=1}^{n} \hat{\boldsymbol{\gamma}}_j^\top \mathbf{B}_j(Z_j^i), \quad 1 \le j \le d. \tag{2.5}$$

The theoretical properties of the estimators $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\gamma}}_j$ have been established in Wang *et al.* (2011). In fact, under certain conditions it follows that

$$\int \left[\hat{m}_j(z_j) - m_j(z_j)\right]^2 p_{Z_j}(z_j)\, dz_j = O_p\left(\left(Mn^{-1}\log n\right)^{\frac{1}{2}}\right), \quad 1 \le j \le d,$$

where p_{Z_j} denotes the marginal density function of Z_j. For the estimator $\hat{\boldsymbol{\beta}}$, it holds that $\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \to N(\mathbf{0}, \boldsymbol{\Sigma})$ for some positive matrix $\boldsymbol{\Sigma}$.

3. Ensemble

Ensemble is a machine learning technique that combines many weak learners to improve prediction accuracy. A weak leaner is a procedure that is slightly better than random prediction. An example is simple tree. A simple tree has no structure so that it may detect possible nonlinear features of the predictors. One disadvantage of the method is that it has low interpretability, which is an obstacle for real data application. In this paper, we consider three ensemble methods based on the simple tree: bagging, logit-boosting, and random forests.

3.1. Bagging

The idea of bagging is to produce weak learners based on bootstrap samples, see Breiman (1996). In classification one seeks a fine learner to predict the class of a point $\mathbf{x}$ based on a training dataset $\mathcal{S}$. Let $\mathcal{S} = \{(Y^1, \mathbf{X}^1), \ldots, (Y^n, \mathbf{X}^n)\}$ be the training dataset, where $Y^i \in \{1, \ldots, J\}$ denotes the class where $\mathbf{X}^i \in \mathbb{R}^p$ belongs to. If one has a sequence of training sets $\{\mathcal{S}_k\}$ and weak learners $f(\cdot, \mathcal{S}_k)$, then one can aggregate the weak learners $f(\cdot, \mathcal{S}_k)$ by majority voting: classify a test point $\mathbf{x}$ by assigning the class which is most frequent among those $f(\mathbf{x}, \mathcal{S}_k)$ predict. Bagging uses bootstrap samples $\{\mathcal{S}_k\}$ obtained from the training set $\mathcal{S}$. Here is the algorithm of bagging.

Bagging Algorithm

(1) Draw bootstrap samples $\mathcal{S}_k$, $1 \le k \le K$, from a training set $\mathcal{S}$;

(2) Fit a weak learner to $\mathcal{S}_k$, $1 \le k \le K$, to find $f(\,\cdot\,, \mathcal{S}_k)$. In this paper, we consider the regression decision tree as a weak learner;

(3) Classify $\mathbf{x}$ by majority voting, *i.e.*, $f_{\text{bagg}}(\mathbf{x}, \mathcal{S}) = \underset{1 \le j \le J}{\operatorname{argmax}} \sum_{k=1}^{K} I\left[f(\mathbf{x}, \mathcal{S}_k) = j\right]$.

3.2. Logitboost

Boosting is a generic term for improving the accuracy of any machine learning algorithm. Among the boosting, Adaboost (Freund and Schapire, 1996) is a popular boosting technique. It is known to be stronger than bagging in most cases. Logitboost (Friedman *et al.*, 2000) is for two-class classification problems and is a statistical version of Adaboost to fit additive logistic regression models using maximum likelihood. The algorithms is stated below.

Logitboost Algorithm

(1) Input a training set $\mathcal{S} = \{(Y^1, \mathbf{X}^1), \ldots, (Y^n, \mathbf{X}^n)\}$, where $Y \in \{-1, 1\}$ and $\mathbf{X} \in \mathbb{R}^p$;

(2) Initialize the weights $w_i^{(0)} \equiv 1/n$, $1 \leq i \leq n$, the committee function $F^{(0)}(\mathbf{x}) = 0$ and the probabilities $p^{(0)}(\mathbf{x}) = P(Y = 1|\mathbf{X} = \mathbf{x}) = 1/2$;

(3) Repeat $k = 1, \ldots, K$

  (a) Compute the weights $w_i^{(k)} = p^{(k-1)}(\mathbf{X}^i)[1 - p^{(k-1)}(\mathbf{X}^i)]$, $1 \leq i \leq n$, and the working responses $Z^i = [(Y^i + 1)/2 - p^{(k-1)}(\mathbf{X}^i)]/w_i^{(k)}$;

  (b) Fit a weak leaner to the dataset $\{(\mathbf{X}^1, Z^1), \ldots, (\mathbf{X}^n, Z^n)\}$ to find a classifier $f^{(k)}$ by weighted least squares regression with the weights $w_i^{(k)}$. In this paper, we consider the regression decision tree as a weak learner;

  (c) Update $F^{(k-1)}(\mathbf{x})$ by $F^{(k)}(\mathbf{x}) = F^{(k-1)}(\mathbf{x}) + f^{(k)}(\mathbf{x})/2$ and $p^{(k-1)}(\mathbf{x})$ by $p^{(k)}(\mathbf{x}) = e^{F^{(k)}(\mathbf{x})}/[e^{F^{(k)}(\mathbf{x})} + e^{-F^{(k)}(\mathbf{x})}]$;

(4) Output the final classifier $f_{\text{LgtBoost}}(\mathbf{x}, \mathcal{S}) = \text{sgn}(F^{(K)}(\mathbf{x}))$.

3.3. Random forests

Random forest proposed by Breiman (2001) is a collection of decision trees to improve classification power. A random forest consists of tree-based classifiers $\{f(\,\cdot\,, \Theta_k) : k \geq 1\}$, where Θ_k are independent identically distributed random vectors. A test point $\mathbf{x}$ is classified by majority voting: each tree makes a single vote $f(\mathbf{x}, \Theta_k)$ and the most popular class is selected.

Random Forest Algorithm

(1) Input data set $\mathcal{S} = \{(Y^1, \mathbf{X}^1), \ldots, (Y^n, \mathbf{X}^n)\}$, where $Y \in \{1, \ldots, J\}$ and $\mathbf{X} \in \mathbb{R}^p$. Choose the number of trees K and positive integers $d \ll p$ and ℓ.

(2) Repeat $k = 1, \ldots, K$

  (a) Get a boostrap sample $\mathcal{S}_k$ from the training set $\mathcal{S}$.

  (b) At each node, choose d input variables at random among $\{X_1, \ldots, X_p\}$. Based on these variables and $\mathcal{S}_k$, find the best split.

  (c) The tree is grown until each terminal node contains no more than ℓ training sample points. There is no pruning. Let $f(\cdot, \mathcal{S}_k)$ denote the constructed tree.

(3) Output the classifier $f_{\text{RandFrst}}(\mathbf{x}, \mathcal{S}) = \underset{1 \leq j \leq J}{\text{argmax}} \sum_{k=1}^{K} I\,[f(\mathbf{x}, \mathcal{S}_k) = j]$.

4. Numerical Experiment

4.1. Simulation study

The main objective of this section is to compare the GPLAM approach and the three ensemble methods in two-class classification. With the GPLAM approach, one predicts $Y = 1$ for an observed point $(\mathbf{X}, \mathbf{Z}) = (\mathbf{x}, \mathbf{z})$ if the associated predictor vectors give a value greater than 1/2 for the estimator of the conditional mean $E(Y|\mathbf{X} = \mathbf{x}, \mathbf{Z} = \mathbf{z})$. To compare different classifiers, we consider the misclassification error as a measure of performance. It is defined as the ratio of the number of misclassified observations with respect to the total number of observations in the test dataset:

$$\mathrm{ME} = \frac{\text{(number of misclassified observations in test dataset)}}{\text{(total number of observations in test dataset)}}. \tag{4.1}$$

Another measure of performance is the AUROC which is defined as the area under the ROC(Receiver Operation Characteristic) curve. Higher value of AUROC means better performance. We also consider the KS(Kolmogorov-Smirnov) statistic, which measures the maximal difference between the cumulative distribution functions of "good" and "bad". To be specific, it is defined as

$$\mathrm{KS} = \max_{\mathbf{x}} |F_G(\mathbf{x}) - F_B(\mathbf{x})|, \tag{4.2}$$

where F_G denote the distribution function of the sample classified as "good", and F_B the one for the sample classified as "bad". In credit scoring it is widely accepted that, if the value of KS statistic is smaller than 0.2, the model is useless; if it is from 0.2 and to 0.4, the model is fair; if it is from 0.4 to 0.6, the model is good; if it is from 0.6 to 0.75, the model is awesome; if it is greater than 0.75, the model is too good to be true.

In the simulation study, we generated 100 training datasets of size $n = 300$, and 500 observations for a test dataset from the model $Y \sim \text{Bernoulli}(g(\boldsymbol{\beta}^\top \mathbf{X} + \boldsymbol{\gamma}^\top \mathbf{m}(\mathbf{Z})))$ for some parameter values $\boldsymbol{\beta} \in \mathbb{R}^7$ and $\boldsymbol{\gamma} \in \mathbb{R}^4$, where $g(u) = \exp(u)/(1 + \exp(u))$ and $\mathbf{m}(\mathbf{z}) = (m_1(z_1), \ldots, m_4(z_4))^\top$. This means we chose $p = 7$ and $d = 4$ in the GPLAM (2.1). We considered three different scenarios with the model. One is the case where there are only linear effects. This corresponds to a parametric logistic linear model. Another is the case where both linear and nonlinear effects exist in the model, which results in a semiparametric model. The last one is the case where only nonparametric effects enter the model, which is a nonparametric generalized additive model. For the first case, we put $\boldsymbol{\beta} = (0.5, 0.1, -0.5, -0.5, 0.3, -1, 0.5)^\top$ and $\boldsymbol{\gamma} = \mathbf{0}$. For the second, we chose $\boldsymbol{\beta} = (0.5, 0, 0, -0.5, 0, -1, 0.5)^\top$ and $\boldsymbol{\gamma} = (3, 0, 0, -1)^\top$. For the last one, we took $\boldsymbol{\beta} = \mathbf{0}$ and $\boldsymbol{\gamma} = (3, -1, 1, -1)^\top$. For the observations of the predictors X_j, we generated X_1 from Bernoulli(0.7), X_2 from Bernoulli(0.3) independently of X_1, and $(X_3, \ldots, X_7)$ from the multivariate normal distribution independently of (X_1, X_2) with mean vector $(0.5, 1, 1.5, 2, 2.5)^\top$ and covariance matrix $\mathbf{V} = (v_{ij})$ where $v_{ij} = 0.2$ for $i \neq j$ and 1 for $i = j$. For the predictors Z_j, we generated them from the uniform distribution on $[0, 1]^4$. The nonparametric functions m_j were chosen as follows: $m_1(z_1) = \sin(3z_1)$, $m_2(z_2) = \sin(3z_2^2)$, $m_3(z_3) = \cos(3z_3)$, $m_4(z_4) = \cos(3z_4^2)$. The smoothing parameter M in fitting the GPLAM was chosen by the GCV criterion.

The results obtained from the 100 pseudo training samples of size $n = 300$ are contained in Table 4.1–4.3. The values in Table 4.1 are the average of ME, defined at (4.1), across the 100 pseudo training samples. The box plots for the 100 values of ME are also displayed in Figure 4.1. Table 4.2 and Table 4.3 give the average values of the AUROC and KS statistics, respectively. In the tables and the figure, we also included the results of fitting a parametric logistic linear model(PLM) and of CART. In the first scenario where only linear effects enter the model, the parametric approach shows the best performance as expected. The GPLAM approach is the next, and CART is the worst. In the second and third scenarios where nonparametric effects are present, the GPLAM gives the best performance among all, and the three ensemble methods outperform the parametric logistic regression approach and CART.

Table 4.1. Average misclssification errors

	Scenario #1	Scenario #2	Scenario #3
PLM	0.28102	0.29920	0.29632
CART	0.35534	0.34048	0.31242
GPLAM	0.28798	0.25728	0.25192
Random Forest	0.30562	0.28400	0.26936
Logitboost	0.29548	0.27114	0.25628
Bagging	0.31410	0.29164	0.27532

Table 4.2. Average values of AUROC statistic

	Scenario #1	Scenario #2	Scenario #3
PLM	0.77539	0.74902	0.68698
CART	0.66516	0.68879	0.66597
GPLAM	0.76665	0.80495	0.77161
Random Forest	0.73560	0.76843	0.73501
Logitboost	0.75459	0.78966	0.76168
Bagging	0.72260	0.75296	0.72431

Table 4.3. Average values of KS statistic

	Scenario #1	Scenario #2	Scenario #3
PLM	0.43396	0.39145	0.30667
CART	0.28789	0.32794	0.30148
GPLAM	0.41892	0.48022	0.43230
Random Forest	0.37647	0.42766	0.37536
Logitboost	0.40022	0.45613	0.41499
Bagging	0.35653	0.40151	0.35964

4.2. German credit data

The German credit dataset consists of one thousand observations on twenty one variables. Among the twenty one variables, one is the response variable taking values 0 and 1, which indicate a ‘good’ and ‘bad’ customer, respectively. There are twenty predictors, among which seven are numeric and thirteen are qualitative attributes. Some predictors have long-tailed distributions, so they are log-transformed. We used 10-fold cross-validation to calculate the misclassification errors. That is, we split the whole dataset into ten groups so that each group has one hundred observations, put aside one group as a test dataset, construct a classifier based on the remaining dataset as a training sample, and then apply the resulting classifier to the test data we put aside. This gives ten values of ME defined at (4.1). We took the average of the ten values as a performance measure.

The results were 0.249 for PLM, 0.268 for CART, 0.239 for GPLAM, 0.235 for random forest, 0.247 for Logitboost, and 0.237 for bagging. The GPLAM, random forest and bagging methods show comparable performance for this particular dataset. One advantage of the GPLAM approach in comparison with the three ensemble methods is that it enables us to estimate the effect of each predictor. The ensemble methods do not have the advantage since they are not based on a structured model.

5. Conclusion

In this paper, we compare several classification methods by simulation and real data analysis. The methods can be applied to credit scoring. Our study suggests that the GPLAM approach has the

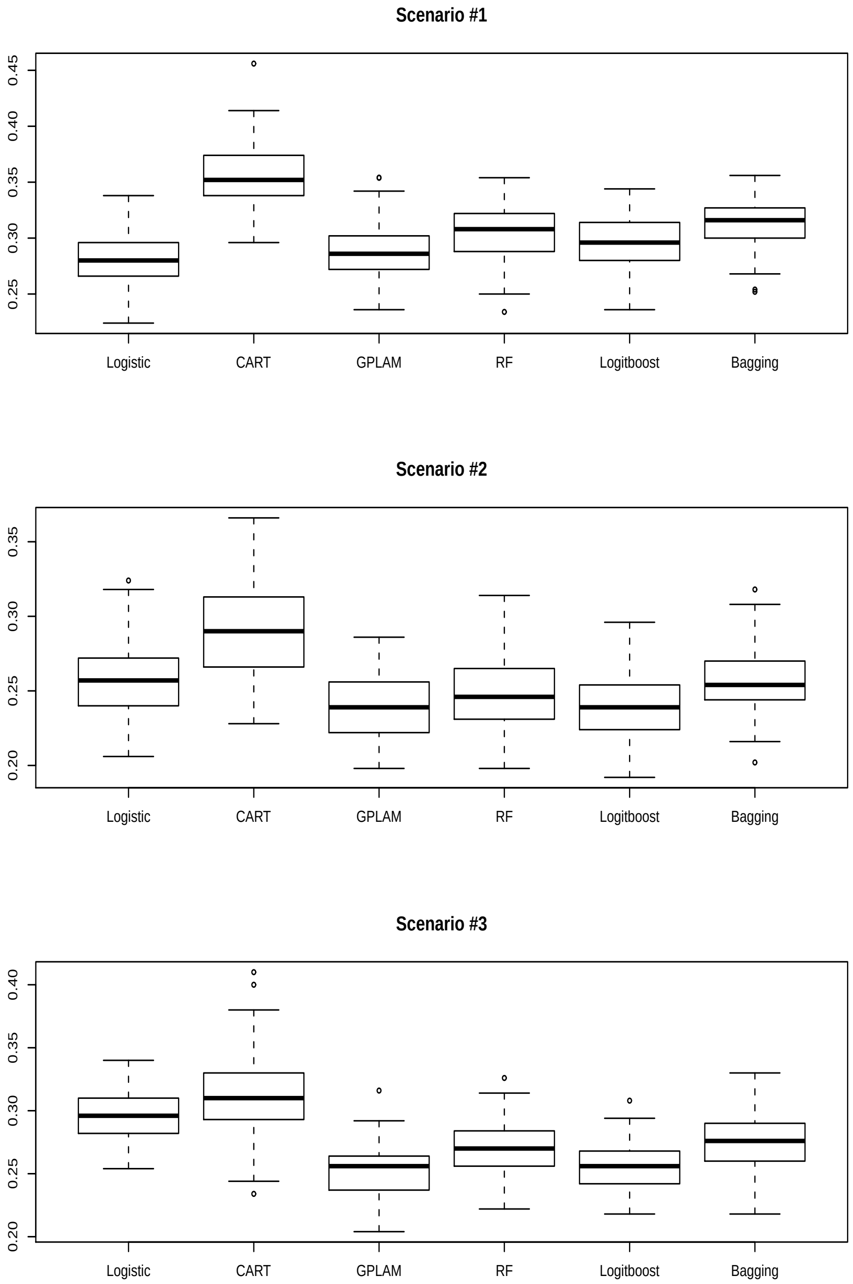

Figure 4.1. Box plots of misclassification errors.

best performance and the three ensemble methods of bagging, logitboost and random forest, are superior to the parametric logistic approach and the traditional CART, when nonlinear effects are present in the model. It also shows that the GPLAM approach is comparable to the parametric approach when the parametric model assumption holds. When one chooses a method for credit scoring, one should consider not only the classification performance but also the interpretability of the effects of the predictor. In this sense, the GPLAM approach seems the best choice since it is based on an estimated model that involves the effects of all predictors.

References

Breiman, L. (1994). *Heuristics of Instability in Model Selection*, Technical Report, Statistics Department, University of California at Berkeley.

Breiman, L. (1996). Bagging predictors, *Machine Learning*, **24**, 123–140.

Breiman, L. (2001). Random forests, *Machine Learning*, **45**, 5–32.

Buja, A., Hastie, T. and Tibshirani, R. (1989). Linear smoothers and additive models (with discussion), *Annals of Statistics*, **17**, 453–555.

Friedman, J., Hastie, T. and Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting, *Annals of Statistics*, **28**, 337–374.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm, In *Machine Learning: Proceedings of the Thirteenth International Conference*, 148–156.

Mammen, E., Linton, O. and Nielsen, J. P. (1999). The existence and asymptotic properties of a backfitting projection algorithm under weak conditions, *Annals of Statistics*, **27**, 1443–1490.

Mays, E. (2001). *Handbook of Credit Scoring*, Fitzroy Dearborn Pub, London.

Severini, T. A. and Staniswalis, J. G. (1994). Quasi-Likelihood estimation in semiparametric models, *Journal of American Statistical Association*, **89**, 501–511.

Thomas, L. C., Edelman, D. B. and Crook, J. N. (2002). *Credit Scoring and Its Applications*, SIAM Society of Industrial and Applied Mathematics, Philadelphia.

Wang, L., Liu, X., Liang, H. and Carroll, R. (2011). Generalized additive partial linear models - polynomial spline smoothing estimation and variable selection procedures, *Annals of Statistics*, in print.

Yu, K. and Lee, Y. K. (2010). Efficient semiparametric estimation in generalized partially linear additive models, *Journal of Korean Statistical Society*, **39**, 299–304.

Yu, K., Park, B. U. and Mammen, E. (2008). Smooth backfitting in generalized additive models, *Annals of Statistics*, **36**, 228–260.