The Korean Journal of Applied Statistics (2010)
23(2), 357–366

Variable Selection with Regression Trees

Youngjae Chang[1]

[1]Research Department, The Bank of Korea

(Received January 2010; accepted February 2010)

Abstract

Many tree algorithms have been developed for regression problems. Although they are regarded as good algorithms, most of them suffer from loss of prediction accuracy when there are many noise variables. To handle this problem, we propose the multi-step GUIDE, which is a regression tree algorithm with a variable selection process. The multi-step GUIDE performs better than some of the well-known algorithms such as Random Forest and MARS. The results based on simulation study shows that the multi-step GUIDE outperforms other algorithms in terms of variable selection and prediction accuracy. It generally selects the important variables correctly with relatively few noise variables and eventually gives good prediction accuracy.

Keywords: Regression tree, random forest, variable selection, bagging.

1. Introduction

The aim of regression analysis is to discover the relationships between the response variable and the predictor variables, and eventually to use the relationships to make predictions based on the information. After a tentative model is fitted, we can assess how well the model fits and modify it to improve the prediction. In this process, it is very important to decide which variables are to be included or removed in the model. If there are many noise variables, the variable selection procedure may play a much more important role in the prediction. Doksum *et al.* (2006) also pointed out this problem. We consider a multi-step regression tree algorithm to solve it.

The regression tree is built through a process known as binary recursive partitioning. This is an iterative process of splitting the data into partitions, and then splitting it further on each of the branches. Figure 1.1 shows an example of a regression tree, where the root node contains all the training observations, and the training data are recursively partitioned by values of the input variables until reaching the terminal nodes (t_4, t_5, t_6, t_8 and t_9) where the predictions are made.

GUIDE (Generalized, Unbiased, Interaction Detection and Estimation (Loh, 2002)) is a flexible regression tree method. The algorithm has little variable selection bias, and it can detect local interactions. Another algorithm is Random Forest (Breiman, 2001), which is a collection of tree predictors such that each tree depends on the values of a bootstrap sample. It has been observed that Random Forest can outperform bagging, and its performance is comparable to that of boosting (Svetnik *et al.*, 2003).

[1]Research Department, The Bank of Korea, 8110, Namdaemunno 3-ga, Chung-gu, Seoul 110-794, Republic of Korea. E-mail: yjchang@bok.or.kr

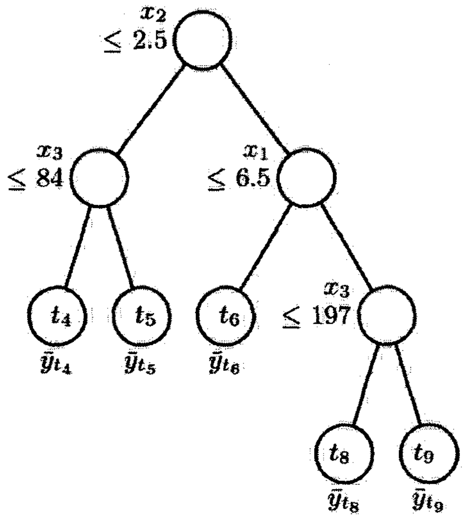

Figure 1.1. Example of a regression tree: At each intermediate node, a case goes to the left child node if and only if the condition is satisfied.

Although they are regarded as good regression tree algorithms, they both suffer from loss of prediction accuracy when there are many noise variables, which is shown by a simulation study. We consider the simulation model,

$$y_i = \mu(X_i) + \epsilon_i, \quad i = 1, \ldots, n$$

with a mean function,

$$\mu(X_i) = 10\sin(\pi x_{i1} x_{i2}) + 20(x_{i3} - 0.5)^2 + 10x_{i4} + 5x_{i5} \tag{1.1}$$

due to Friedman (1991), where X_i is a d-dimensional vector of predictors (*i.e.* $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$) whose component variables $x_{i1}, x_{i2}, \ldots, x_{id}$ are *i.i.d* from $U(0,1)$ with $\epsilon_i \sim N(0, 0.1^2)$. Consequently, there are only five important variables in the model $(x_{i1}, x_{i2}, \ldots, x_{i5})$, and the remainders $(x_{i6}, x_{i7}, \ldots, x_{id})$ are noise variables. We generate 1,000 X_i's for a learning sample and 5,000 X_i's for a testing sample, which means $n = 6000$ in this case. We use the estimated MSE to measure the prediction accuracy in the simulation study, defined by

$$\text{MSE} = \frac{\sum_{test} (\hat{\mu}(X_i{}^*) - \mu(X_i{}^*))^2}{N_{test}}, \tag{1.2}$$

where X^* denotes the testing sample, N_{test} denotes the number of observations in the testing sample, and $\hat{\mu}(X_i{}^*)$ denotes the predicted value of y_i. We use the average MSE based on 100 simulation replicates. Figure 1.2 shows the problem as the number of noise variables increases. The average MSE increases as more noise variables are added to the model.

We propose the multi-step GUIDE to solve this problem. Our algorithm is composed of two parts: a variable selection procedure; and an actual fitting procedure with a reduced number of variables. Regardless of the number of noise variables, the algorithm shows good and stable prediction performance overall.

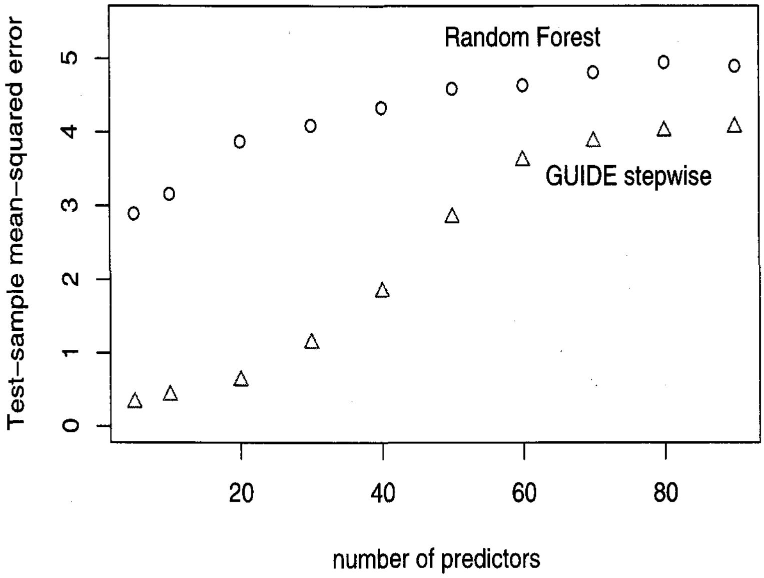

Figure 1.2. Performance of algorithms as noise variables increase for the simulation model (1.1)

This paper is organized as follows. In Section 2 we describe the features of algorithms such as two-step GUIDE, multi-step GUIDE, and Random Forest in the presence of noise variables, followed by the simulation study and real data analysis. In Section 3, we propose some possible future work based on the results.

2. Robustifying Regression Trees in the Presence of Noise Variables

2.1. Review of the GUIDE algorithm

Loh (2002) proposed the GUIDE algorithm, which has negligible selection bias and relatively low computational cost. GUIDE is also known as a smart data mining tool with flexible model fitting methods at each node. The procedure for fitting a stepwise linear regression model in GUIDE is as follows:

1. Let t denote the current node. Use stepwise regression which allows addition and deletion of variables. The default values of F-to-enter and F-to-remove are 4.00 and 3.99, respectively.
2. Do not split a node if the R^2 of the fitted model is greater than 0.99 or the current node has less than $2n_0$ observations, where n_0 is a previously specified number.
3. For each observation, define the class variable Z by the sign of its residual for each observation. That is, Define $Z = 1$ if the observation is associated with a positive residual. Otherwise, define $Z = 0$.
4. Construct a $2 \times m$ cross-classification table for each predictor variable X. The rows of the table are the values of Z, while the columns of the table are 4 intervals at the sample quartiles if X is a numerical variable($m = 4$). If X is a categorical variable, its m distinct values form the columns of the table. Compute a p-value for the chi-squared test for each X based on the table.

5. In addition to the above "curvature" tests, perform chi-squared tests to detect interactions between pairs of same type variables (*i.e.*, numerical variable pairs, categorical variable pairs) or between pairs of different types of predictors. If a pair of variables from these interaction tests gives the smallest p-value, the split variable is one of two variables depending on the composition of the pairs.
6. Select the split variable X from the previous steps. Let t_L and t_R denote the left and right subnodes of t.
   - If X is a numerical variable, search for the split point which gives the lowest total of the sums of squared residuals in t_L and t_R, provided that the number of observations at each node is at least n_0 or user-specified value.
   - If X is a categorical variable, search for the split of the form $X \in C$, which gives the lowest weighted sum of the variances of Z in t_L and t_R, provided that the number of observations at each node is at least n_0. Here C is a subset of the values taken by X, and weights are proportional to sample sizes.
7. After splitting has stopped, prune the tree with a test sample or by cross-validation.

There are several models we can fit at each node of the regression tree other than stepwise linear option described above. For example, we can fit "constant", "simple linear" or "multiple linear" model at each node. Generally, the constant regression tree is a little larger than others in terms of the number of terminal nodes.

2.2. Proposed algorithms

2.2.1. Two-step GUIDE We focus on the fact that only some variables are selected for the split variables in GUIDE, which can be regarded as important ones. This is the motivation of using GUIDE as a variable selection tool. So, we can think about a simple two-step GUIDE algorithm, which is summarized as follows.

1. Follow the steps in GUIDE described in the previous section to grow a tree and prune it back.
2. Extract variable usage information from the tree. Note that variables which appear at the upper nodes of a tree for splitting or fitting could be regarded as more important variables. Variables not used for splitting or fitting are removed. We call this the screening step or first step.
3. The learning sample with the remaining variables is used to grow a new tree. We use GUIDE once again for this purpose. We call this the fitting step or second step.

2.2.2. Multi-step GUIDE We also consider multi-step screening instead of only two steps. We could see that stepwise GUIDE as a screening step tends to select more variables than needed in the simulation study (Table 2.3). This brings an idea of multi-step screening with stepwise GUIDE until the variable selection result gets stable. In other words, we do GUIDE stepwise screening step repeatedly until the variable selection result remains unchanged. And we fit a model with a reduced number of variables.

2.3. Random forest

Breiman (2001) proposed Random Forest which is an accurate algorithm having the unusual ability to handle many variables without deletion or deterioration of accuracy. Its prediction accuracy is known to be fairly good. We present the basic steps of Random Forest in the regression case.

1. Learning sample is a bootstrap sample from the original learning sample.
2. A split variable is chosen among *mtry* variables which are randomly selected at each node. The value of *mtry* is a pre-specified integer. For a split, the Gini impurity criterion is used.
3. After a test vector X is put down each tree, the predicted value from this single tree is the mean of the dependent variables of the learning sample at the node it reaches.
4. The average of these mean values of the dependent variables over all trees in the forest is the predicted value for X.

2.4. Simulation and real data analysis

We use three simulation models and ten real datasets for the comparison of the algorithms. We compare various algorithms in terms of prediction accuracy. Variable selection results by the regression tree methods are also presented.

2.4.1. Simulation models and results Following Friedman (1991), we do simulation experiments as follows.

As described before, the simulation model is,

$$y_i = \mu(X_i) + \epsilon_i, \quad i = 1, \ldots, n,$$

where X_i is a d-dimensional vector of predictors (*i.e.* $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$) of which component $x_{i1}, x_{i2}, \ldots, x_{id}$ are generated in the form of $U(0,1)$ and $\epsilon_i \sim N(0, 0.1^2)$. In addition to (1.1), the following mean functions are used.

$$\mu(X_i) = 0.1e^{4x_{i1}} + \frac{4}{1 + e^{-20(x_{i2}-0.5)}} + 3x_{i3} + 2x_{i4} + x_{i5} \tag{2.1}$$

and

$$\mu(X_i) = \begin{cases} x_{i1} + x_{i2} + x_{i3} + x_{i4} + x_{i5}, & \text{if } x_{i1} + x_{i2} \leq 1, \\ 5, & \text{otherwise.} \end{cases} \tag{2.2}$$

There are 1000 points for learning sample (*i.e.* $i = 1, \ldots, 1000$) and 5000 points for testing sample (*i.e.* $i = 1001, \ldots, 6000$) out of $n = 6000$ observations. Estimated MSE in the form of (1.2) is used to measure the prediction performance of these algorithms. We take the average of the 100 $\widehat{\text{MSE}}$'s for each d (10 $\widehat{\text{MSE}}$'s for bagging), where d ranges from 5 (no noise variable) to 90.

GUIDE stepwise(Gs), GUIDE bagging(stepwise)(BG), two-step GUIDE(stepwise-stepwise; Gss, constant-stepwise; Gcs), multi-step GUIDE(multi-stepwise; Gs..s), Random Forest(RF) and Multivariate Adaptive Regression Splines(MARS) algorithms are compared. MARS is a nonparametric regression procedure proposed by Friedman (1991), which constructs the relationship between the dependent and independent variables from a set of coefficients and basis functions driven from the regression data. We also consider the GUIDE multiple linear with pure dataset(Gm') which does not have any noise variable for comparison purpose. So, if any method's result stands in line with the result of Gm' method, it could be regarded as very good one.

The simulation result is summarized in Table 2.1. Overall, the two-step and multi-step methods perform very well showing relatively close results to that of Gm' which is the GUIDE multiple linear model with a pure dataset without any noise variables.

Table 2.1. Average MSE's by algorithms for simulation models (standard error)

Models	number of predictors	Gs	BG	Gcs	Gss	Gs..s	MARS	Gm'
(1.1)	$d = 5$	0.2283 (0.003)	0.0876 (0.006)	0.3790 (0.004)	0.2283 (0.003)	0.2452 (0.009)	1.884 (0.006)	0.2480 (0.008)
	$d = 30$	0.9378 (0.04)	0.5022 (0.03)	0.4307 (0.03)	0.3902 (0.02)	0.3484 (0.01)	1.995 (0.007)	0.2360 (0.006)
	$d = 60$	3.603 (0.07)	1.209 (0.05)	0.6538 (0.07)	0.4350 (0.02)	0.3186 (0.01)	2.132 (0.01)	0.2426 (0.009)
	$d = 90$	4.068 (0.02)	1.569 (0.07)	0.7484 (0.08)	0.3268 (0.01)	0.2851 (0.005)	2.283 (0.01)	0.2419 (0.01)
(2.1)	$d = 5$	0.0089 (0.0004)	0.005 (0.0004)	0.0161 (0.0002)	0.0089 (0.0004)	0.0086 (0.0002)	0.0137 (0.0005)	0.009 (0.0002)
	$d = 30$	0.0353 (0.002)	0.0155 (0.0008)	0.0420 (0.004)	0.0148 (0.0006)	0.0125 (0.0003)	0.0132 (0.0006)	0.0089 (0.0001)
	$d = 60$	0.0873 (0.004)	0.0387 (0.002)	0.0655 (0.005)	0.0238 (0.002)	0.0147 (0.0007)	0.0147 (0.0006)	0.0088 (0.0002)
	$d = 90$	0.1359 (0.005)	0.0574 (0.002)	0.0631 (0.005)	0.0321 (0.002)	0.0152 (0.001)	0.0166 (0.0006)	0.0088 (0.0002)
(2.2)	$d = 5$	0.2239 (0.005)	0.1478 (0.005)	0.2164 (0.005)	0.2239 (0.005)	0.2330 (0.005)	0.6114 (0.001)	0.2086 (0.003)
	$d = 30$	0.3633 (0.01)	0.3425 (0.02)	0.2486 (0.006)	0.2721 (0.006)	0.2810 (0.01)	0.6386 (0.002)	0.2075 (0.003)
	$d = 60$	0.4663 (0.009)	0.4893 (0.01)	0.2550 (0.006)	0.2703 (0.007)	0.2637 (0.008)	0.6842 (0.003)	0.2065 (0.003)
	$d = 90$	0.5176 (0.006)	0.5429 (0.009)	0.2666 (0.006)	0.2661 (0.007)	0.2489 (0.006)	0.7337 (0.004)	0.2171 (0.004)

Table 2.2. Variable selection result by Gc out of 100 replicates for each d; For the number of selected variables ≥ 7 of $d = 30, 60$ and 90, 7~9 are selected for the model (1.1), 8~9 variables for the model (2.1) and 7~8 variables for the model (2.2) on average.

Models	number of predictors	# selected ≤ 4	5	6	≥ 7
(1.1)	$d = 5$	0(0)	100(100)	0(0)	0(0)
	$d = 30$	2(2)	50(50)	29(29)	19(19)
	$d = 60$	10(10)	61(59)	19(19)	10(10)
	$d = 90$	13(13)	67(66)	17(15)	3(3)
(2.1)	$d = 5$	0(0)	100(100)	0(0)	0(0)
	$d = 30$	13(13)	18(10)	19(17)	50(46)
	$d = 60$	27(27)	27(19)	23(13)	23(17)
	$d = 90$	29(29)	28(19)	23(18)	20(15)
(2.2)	$d = 5$	1(1)	99(99)	0(0)	0(0)
	$d = 30$	27(25)	24(21)	26(26)	23(22)
	$d = 60$	37(35)	30(25)	21(19)	12(12)
	$d = 90$	55(49)	19(14)	12(12)	14(14)

We are also interested in the accuracy of variable selection results. The variable selection results of GUIDE constant(Gc) and GUIDE stepwise(Gs) are summarized in the Table 2.2 and 2.3. The numbers in the parenthesis are number of cases in which all the selected variables are important (when 5 or less variables are selected), or all the five important variables are included among the selected variables (when 6 or more variables are selected).

We also try to improve RF method adding a screening step like Gc and Gs before RF is actually

Table 2.3. Variable selection result by Gs out of 100 replicates for each d; For the number of selected variables ≥ 7 of $d = 30, 60$ and 90, 15$\sim$17 are selected for the model (1.1), 16$\sim$25 variables for the model (2.1), and 12$\sim$14 variables for the model (2.2) on average.

Models	number of predictors	# selected ≤ 4	5	6	≥ 7
(1.1)	$d = 5$	0(0)	100(100)	0(0)	0(0)
	$d = 30$	0(0)	0(0)	0(0)	100(100)
	$d = 60$	0(0)	0(0)	0(0)	100(100)
	$d = 90$	0(0)	0(0)	0(0)	100(100)
(2.1)	$d = 5$	0(0)	100(100)	0(0)	0(0)
	$d = 30$	0(0)	0(0)	0(0)	100(100)
	$d = 60$	0(0)	0(0)	0(0)	100(100)
	$d = 90$	0(0)	0(0)	0(0)	100(100)
(2.2)	$d = 5$	0(0)	100(100)	0(0)	0(0)
	$d = 30$	0(0)	2(2)	5(5)	93(93)
	$d = 60$	0(0)	1(1)	5(5)	94(94)
	$d = 90$	0(0)	0(0)	3(3)	97(97)

executed, which is similar to two-step GUIDE and get a fairly good result from the GcRF(Gc for the first step and RF for the second step) and GsRF(Gs for the first step and RF for the second step). All the simulation results are presented in Figure 2.1.

2.4.2. Real data analysis We prepare ten real datasets (Table 2.4) to compare the performance of the algorithms. The algorithms used for this purpose are two-step GUIDE(Gcs, Gss), multi-step GUIDE(Gs..s), GUIDEd-Random Forest(GsRF), GUIDE stepwise without noise variable(Gs') and MARS. Unlike the result of previous simulation study, GcRF performs very poor in the real data analysis. This is due to too aggressive variable selection result from the screening step of Gc for the real data sets, which means too many variables are removed by the first step of Gc. In addition, with categorical variables being included, Gs tends to select the important variables better compared to Gc. Therefore, we use GsRF method instead of GcRF for the real data.

Let M denote total number of predictors without any noise variables. The data sets we use are generated as follows. For each data set, $0.5 \times M$ noise variables following $N(0, 1)$ are added and denoted by "noise0.5". Similarly, $1 \times M$ noise variables are added and denoted by "noisel", $2 \times M$ by "noise2" and $4 \times M$ by "noise4". Finally, these data sets with noise variables are randomly divided into ten parts, and each one of them is used for the testing sample, and the rest of them, nine parts are merged to be used for the learning sample. This ten-fold cross validation gives ten MSE's for each case, and we take the average to compare. Therefore, we can see the pattern through each average MSE of "noise0.5", "noisel", "noise2" and "noise4" for the algorithms, which is generally stable for all two-step methods on each data set. Since most of algorithms produce less MSE's than Gs', they could be regarded as good ones in terms of prediction accuracy. We could also compare the algorithms by the geometric mean of relative MSE's compared with that of Gs' across ten data sets. All the algorithms except MARS look quite stable (Figure 2.2).

3. Conclusion and Future Work

The simulation study shows that the performance of multi-step GUIDE is effective, demonstrating much improvement over a single tree algorithm. It performs better than Random Forest and MARS even when many irrelevant variables are added to the model. It generally selects the important variables correctly with relatively few irrelevant variables, which gives good prediction accuracy.

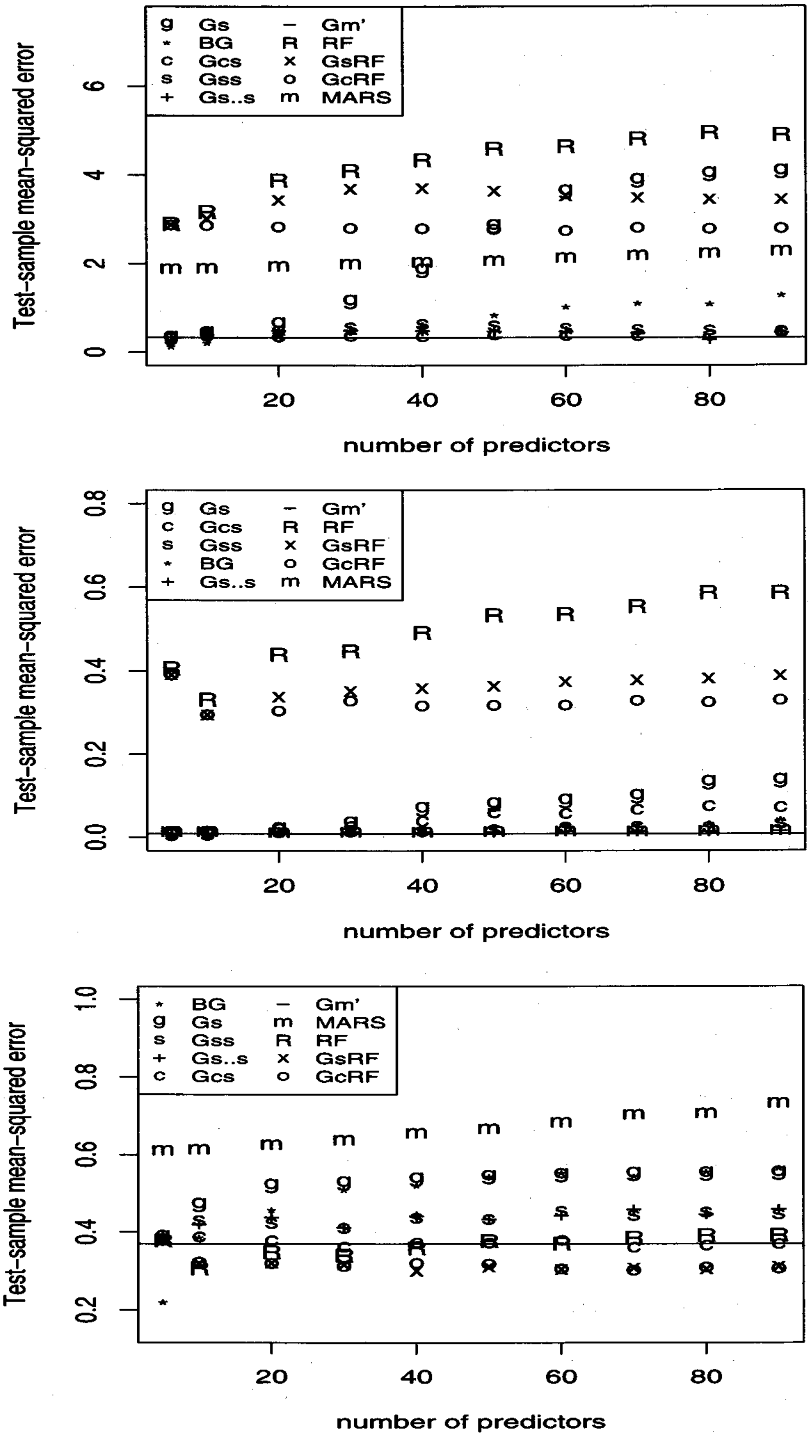

Figure 2.1. Comparison of algorithms for simulation model (1.1) , (2.1) and (2.2)

Table 2.4. Real data sets description (No noise variable)

Dataset Name	# of sample	# of numerical variable	# of categorical variable	Source
Abalone	4177	7	1	UCI
Ais	202	11	1	Cook and Weisberg (1994)
Alcohol	2467	12	6	Kenkel and Terza (2001)
Amenity	3044	19	2	Chattopadhyay (2003)
Baseball	263	16	4	Statlib
Boston	506	13	0	Belsley (1980)
Cane	3775	6	3	Denman and Gregory (1998)
College	694	23	1	Statlib
Deer	654	10	3	Onoyama *at al.* (1998)
Enroll	258	6	0	Liu and Stengos (1999)

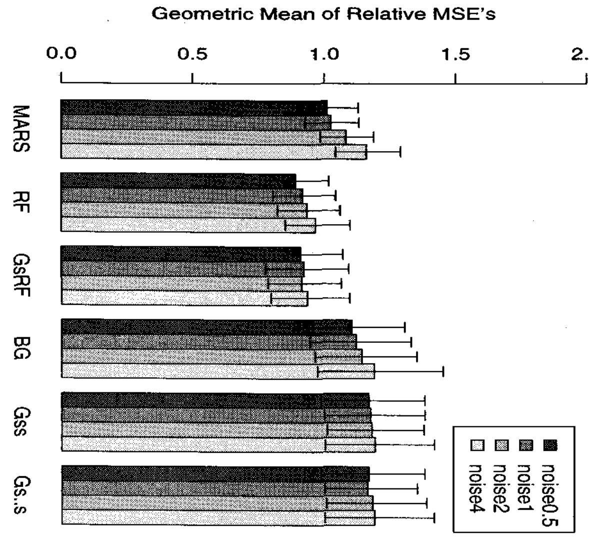

Figure 2.2. Comparison of algorithms for real data

In addition to prediction accuracy, the multi-step GUIDE is an economical algorithm in terms of computation time. After the first iteration of variable selection step is executed, the remaining steps go very fast since quite a few noise variables are removed in the first iteration. So, the multi-step algorithm does not take much more time than running GUIDE just once. For example, in the case of 90 predictors, multi-step GUIDE takes less than 7 seconds per simulation while simple GUIDE takes about 5 seconds(based on the machine with Intel 2.8Ghz Pentium 4 processor). In fact, GUIDE itself is known to be a fast algorithm.

As a variable selection tool, GUIDE screening step can be used for other algorithms. We see that Random Forest shows good prediction accuracy in the real data example, but it could be improved with GUIDE variable selection step when there are many noise variables. We can think about the application of multi-step approach to the classification. In the similar setting as the one used for regression trees in this paper, we can consider efficient ways to detect important variables in the classification problem. One approach is using the classification tree algorithms. It could be an easy and simple shortcut to do such a job among many possible methods for this problem.

References

Belsley, D. A. (1980). On the efficient computation of the nonlinear full-information maximum-likelihood estimator, *Journal of Econometrics*, **14**, 203–225.

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

Chattopadhyay, S. (2003). Divergence Between the Hicksian Welfare Measures: The Case of Revealed Preference for Public Amenities, *Journal of Applied Econometrics*, **17**, 641–66.

Cook, D. and Weisberg, S. (1994). *An introduction to Regression Graphics*, Wiley, New York.

Denman, N. and Gregory, D. (1998). Analysis of sugar cane yields in the mulgrave area, for the 1997 sugar cane season, *Technical report, MS305 Data Analysis Project*, Department of Mathematics, University of Queensland.

Doksum, K., Tang, S. and Tsui, K. W. (2006). Nonparametric variable selection: The EARTH algorithm, *Journal of the American Statistical Association*, **103**, 1609–1620.

Friedman, J. H. (1991). Multivariate adaptive regression splines, *Annals of Statistics*, **19**, 1–67.

Kenkel, D. and Terza, J. (2001). The effect of physician advice on alcohol consumption: countregression with an endogenous treatment effect, *Journal of applied econometrics*, **16**, 165–184.

Liu, Z. and Stengos, T. (1999). Non-linearities in cross country growth regressions: A semiparametric approach, *Journal of Applied Econometrics*, **14**, 527–538.

Loh, W. Y. (2002). Regression trees with unbiased variable selection and interaction detection, *Statistica Sinica*, **12**, 361–386.

Onoyama, K., Ohsumi, N., Mitsumochi, N. and Kishihara, T. (1998). Data analysis of deer-train collisions in eastern Hokkaido, *Data Science, Classification, and Related Methods* (ed. by Hayashi, C., Ohsumi, N., Yajima, K., Tanaka, Y., Bock, H.-H., Baba, Y.), 746–751, Japan.
BMC Bioinformatics, 8:25

Svetnik, V., Liaw, A., Tong, C. and Culberson, J. C. (2003). Random forest: A classification and regression tool for compound classification and QSAR modeling, *Journal of Chemical Information and Computer Sciences*, **43**, 1947–1958.