Variable Selection for Classification and Regression in Large p, Small n Problems

Wei-Yin Loh

Abstract Classification and regression problems in which the number of predictor variables is larger than the number of observations are increasingly common with rapid technological advances in data collection. Because some of these variables may have little or no influence on the response, methods that can identify the unimportant variables are needed. Two methods that have been proposed for this purpose are EARTH and Random forest. This article presents an alternative method, derived from the GUIDE classification and regression tree algorithm, that employs recursive partitioning to determine the degree of importance of the variables. Simulation experiments show that the new method improves the prediction accuracy of several nonparametric regression models more than Random forest and EARTH. The results indicate that it is not essential to correctly identify all the important variables in every situation. Conditions for which this occurs are obtained for the linear model. The article concludes with an application of the new method to identification of rare molecules in a large genomic data set.

1 Introduction

Consider the problem of fitting a nonparametric regression model to a response variable y on p predictor variables, $\mathbf{x}_p = (x_1, x_2, \ldots, x_p)$. Let $\mu = \mu(\mathbf{x}_p) = E(y \,|\, \mathbf{x}_p)$ denote the conditional mean of y given $\mathbf{x}_p$ and let $\hat{\mu}_n(\mathbf{x}_p)$ be the value of μ at $\mathbf{x}_p$ estimated from a training sample of size n. The expected squared error is $E[\hat{\mu}_n(\mathbf{x}_p^*) - \mu(\mathbf{x}_p^*)]^2$, where $\mathbf{x}_p^*$ is an independent copy of $\mathbf{x}_p$ and the expectation is over the training sample and $\mathbf{x}_p^*$. In many applications, the mean function $\mu(\mathbf{x}_p)$ may depend on only a small but unknown subset of the x_i variables. We call the latter

Wei-Yin Loh
University of Wisconsin, Madison, WI 53706, e-mail: loh@stat.wisc.edu

To appear in: *Probability Approximations and Beyond* (2012), A Barbour, HP Chan and D Siegmund (eds.), *Lecture Notes in Statistics—Proceedings*, vol 205, Springer, pp 133–157

variables “important” and the others “unimportant.” If n is fixed and the number of unimportant variables increases, the expected squared error typically increases too. This occurs even for modern nonparametric fitting algorithms that perform variable selection on their own.

To see this, let $n = 100$, $p \geq 5$, and $\mathbf{x}_p$ be a vector of mutually independent and uniformly distributed variables on the unit interval. Consider the six models

$$y = 5[2\sin(\pi x_1 x_2) + 4(x_3 - 1)^2 + 2x_4 + x_5] + \varepsilon/5 \tag{1}$$

$$y = 10^{-1}\exp(4x_1) + 4[1 + \exp(-20x_2 + 10)]^{-1} + 3x_3 + 2x_4 + x_5 + \varepsilon \tag{2}$$

$$y = x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 + \varepsilon \tag{3}$$

$$y = 5[2\sin(4\pi x_1 x_2) + 4(x_3 - 1)^2 + 2x_4 + x_5] + \varepsilon/5 \tag{4}$$

$$y = 10(x_1 + x_2 + x_3 + x_4 + x_5 - 5/2)^2 + \varepsilon/10 \tag{5}$$

$$y = \mathrm{sgn}[(2x_1 - 1)(2x_2 - 1)](3x_3 + 4x_4 + 5x_5) + \varepsilon \tag{6}$$

where ε is independent standard normal. Models (1) and (2) are used in [5]. Model (3) is linear and Model (4) is a minor modification of (1) with 4π in place of π. Models (5) and (6) have strong interaction effects.

Figure 1 shows estimated values of the expected squared errors of MARS [5], GUIDE [6], and Random forest [1] for these six models for $p = 5$, 20, 50, 100, 200, and 500. Each estimate is based on 600 simulation trials; the simulation standard error bars are too small to be shown in the plots. GUIDE fits a piecewise-linear regression tree using stepwise regression in each node of the tree. Random forest is an average of 500 piecewise-constant regression trees. The initial rapid rise in the expected squared error as p increases is obvious. MARS is best in one model and worst in three; Random forest is best in two and worst in three; and GUIDE is best in two and worst in none.

Can the expected squared errors of these regression methods be reduced by preselecting a subset of the predictor variables? To this end, several approaches for assigning “importance scores” to the predictors have been proposed. Random forest itself produces importance scores as by-products. Recall that the algorithm constructs an ensemble of piecewise-constant regression trees from bootstrap samples of the training data. The observations not in a bootstrap sample are called the “oob” (out of bag) sample. To measure the predictive power of a variable x_i, the expected squared error of each tree is estimated twice with the oob sample, once with and once without randomly permuting their x_i values. A small difference between the two error estimates indicates that the variable has low predictive power. The importance score assigned to x_i is the average of the differences across the trees in the ensemble.

A strength of Random forest is its applicability to all data types, including data with missing values. Simulations show, however, that its importance scores can be unreliable because their variances depend on the type of predictor variable. Variables that allow more splits, such as categorical variables with many categories, have scores with larger variances. One proposed solution [10] replaces the split selection

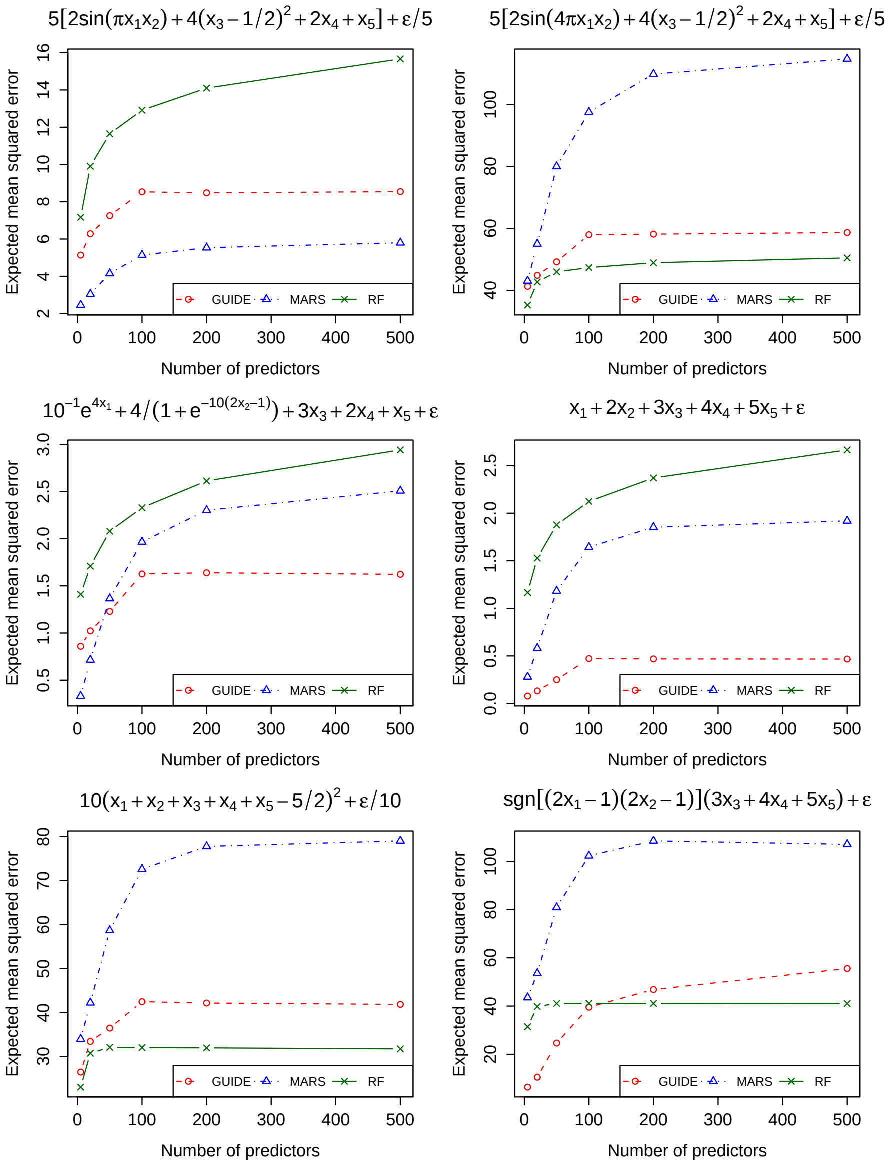


Fig. 1 Simulated values of $E(\hat{\mu}_n-\mu)^2$ of GUIDE, MARS and Random forest (RF) versus number of unimportant predictor variables, with ε standard normal. Simulation standard errors are about the size of the plot symbols.

procedure with permutation tests and changes bootstrap sampling to sampling without replacement.

Neither Random forest nor this modification [10] gives a threshold value of the scores for identifying the important variables. This problem is solved in [11] by supplementing the training sample with a set of artificially created variables obtained by randomly permuting the real predictor variables. A variable is declared important if its importance score is larger than the 75th percentile of the scores of the artificial variables. The process is repeated several times on the residuals to select additional real variables. One disadvantage of adding artificial variables is that it increases the computational requirements. A simpler solution [4] adds thirty independent and uniformly distributed artificial variables to the training data and takes the threshold to be two times the mean of the importance scores from the artificial variables. Because Random forest is biased toward selecting variables that allow more splits, however, this approach yields incorrect results if all the x_i variables are nominal-valued.

EARTH [4] tries a different approach by ranking the x_i variables according to the strength of its relationship with the y variable. For each x_i, a user-specified number, m, of points from the training sample are randomly chosen. A short, narrow tube is constructed around each chosen point, with axis in the x_i direction. A polynomial (usually first order) model is fitted to the data in the tube and the F-statistic for testing the null hypothesis that $E(y)$ is constant within the tube is computed. The tube length is gradually increased to find the largest value of the F-statistics. The importance score $l(x_i)$ for x_i is the average of the square roots of the maximal F-statistics over the m points. To determine a threshold for the scores, the whole process is repeated with the y-values randomly permuted to obtain the corresponding scores $l^*(x_i)$. Variable x_i is declared unimportant if the difference $l(x_i) - l^*(x_i)$ is less than a pre-specified multiple of the standard deviation of the $l^*(x_i)$. Simulation results in [4] show that if EARTH is used to select variables before application of GUIDE or MARS, their expected squared errors can be reduced. EARTH is not applicable, however, if either y or some x_i are categorical (i.e., nominal-valued) variables.

Yet another method [3], applicable only to discrete-valued x_i, randomly selects subsets of the x_i variables to optimize the total variation of the y values within the partitions defined by the values of the selected variables. The method appears to be practicable only for binary-valued x_i variables, and it is not applicable to categorical y variables. In the next section, we introduce a new variable selection method based on the GUIDE algorithm that does not have such limitations.

2 GUIDE variable selection

A classification or regression tree algorithm typically partitions the data in a node of a tree with a split of the form “$x_i \leq c$” (if x_i is an ordered variable) or “$x_i \in S$” (if x_i is a categorical variable). Many algorithms, such as CART [2], search for the best x_i and c or S simultaneously, by optimizing a measure of node impurity such as entropy (for classification) or sum of squared residuals (for regression). Besides being computationally expensive, this approach creates a bias toward selecting variables that allow more splits of the data—see [6, 7]. To avoid the bias and to reduce computational cost, GUIDE uses chi-squared tests to choose the x_i variable before searching for c or S.

Consider first the classification problem, where y is a categorical variable. At each node t and for each x variable, GUIDE computes the significance probability $q(x,t)$ of the chi-squared contingency table test of independence between y and x, with the values of y forming the rows of the table. If x is a categorical variable, its labels form the columns of the table. If x is an ordered variable, its range is split into K intervals to form the columns. The value of K is determined by the sample size $n(t)$ in t. If $n(t) < 40$, then $K = 3$; otherwise $K = 4$. The specific steps for a J-valued y variable may be briefly stated as follows.

Algorithm 1 *Variable and split selection for classification.*

1. *For each ordered variable x_i:*
   a. *Group the values of x_i into K intervals with approximately equal numbers of observations in each group.*
   b. *Form a $J \times K$ contingency table, with the values of y as rows and the intervals of x_i as columns.*
2. *For each categorical variable x_i:*
   a. *Let m_i denote the number of distinct values of x_i in t.*
   b. *Form a $J \times m_i$ contingency table, with the values of y as rows and the categories of x_i as columns.*
3. *Compute the P-value $q(x_i,t)$ of the chi-squared test of independence.*
4. *Find $\chi_1^2(x_i,t)$, the upper $q(x_i,t)$-quantile of the chi-squared distribution with one degree of freedom.*
5. *Let x_i^* be the variable with the smallest $q(x_i,t)$. If x_i^* is an ordered variable, split t into two subnodes at the sample median of x_i^*. If x_i^* is categorical, split t with the procedure detailed in [7].*

This algorithm is applied recursively to construct a binary tree with four levels of splits. The importance score of variable x is

$$\mathrm{IMP}(x) = \sum_t \sqrt{n(t)}\,\chi_1^2(x,t) \tag{7}$$

where the sum is over the intermediate nodes of the tree. A similar procedure is followed for regression, except that at each node, y is first converted to a binary-valued categorical variable y' that takes value 1 if y is above its node mean and 0 otherwise.

If x is independent of y, the score $\mathrm{IMP}(x)$ is a weighted sum of approximately independent chi-squared random variables, each having one degree of freedom. By the Satterthwaite [8] method, its distribution can be approximated by a scaled chi-squared distribution. We use the upper p^{-1}th-quantile of the latter distribution as the threshold for identifying the important variables.

Figures 2–4 compare the probabilities with which variables $x_1, x_2, \ldots, x_5$ are selected by the our GUIDE method, EARTH and Random forest (the last using the thresholding method of [4]) for simulation models (1)–(6). The results are based on 600 simulation trials with $n = 100$ and $p = 5$, 20, 50, 100, 200, and 500, yielding standard errors of 0.02 or smaller. For $p = 5$, i.e., when there are no unimportant variables, our method is almost always best, sometimes by wide margins—see Figure 4. But when there are many unimportant variables, e.g., when $p = 500$, Random forest is best and our method is a distant third.

The large probabilities with which EARTH and Random forest select the important variables come at the cost of larger numbers of unimportant variables being selected as well, as shown in Figure 5 which plots the average number versus p (on the logarithmic scale) for each model. The higher false positive rates may be seen in Figure 6 too, which shows the mean number of variables selected by each method when $E(y)$ is constant, independent of all the x variables. In this situation, EARTH and Random forest have false positive rates of about 10% compared to 1% for our method.

To see how the results change if some of the x variables are correlated, we follow [4] by generating $x_i = \Phi(z_i)$, $i = 1, 2, \ldots, 9$, where Φ is the standard normal distribution function, $(z_1, z_2, \ldots, z_9)$ is multivariate normal with zero mean and covariance matrix

$$\Sigma = \begin{pmatrix}
1.0 & & & & 0.9 & & & & \\
 & 1.0 & & & & 0.9 & & & \\
 & & 1.0 & & & & 0.5 & & \\
 & & & 1.0 & & & & 0.2 & 0.2 \\
0.9 & & & & 1.0 & & & & \\
 & 0.9 & & & & 1.0 & & & \\
 & & 0.5 & & & & 1.0 & & \\
 & & & 0.2 & & & & 1.0 & 0.2 \\
 & & & 0.2 & & & & 0.2 & 1.0
\end{pmatrix} \tag{8}$$

and x_i independent and uniformly distributed on the unit interval for $i = 10, 11, \ldots, p$. Thus x_1 and x_5 are highly correlated, as are x_2 and x_6; x_3 is moderately correlated with x_7, and x_4 is moderately correlated with x_8 and x_9. Note that x_6, x_7, x_8 and x_9 do not appear explicitly in models (1)–(6). Figures 7–9 show the resulting selection probabilities for $p = 10$, 20, 50, 100, 200, and 500. The high correlation between x_1 and x_5 increases their selection probabilities for all three methods in models (1), (2), (3), and (5) and decreases them in model (4). The odd exception is model (6), where the probabilities are increased for Random forest but decreased for EARTH and GUIDE.

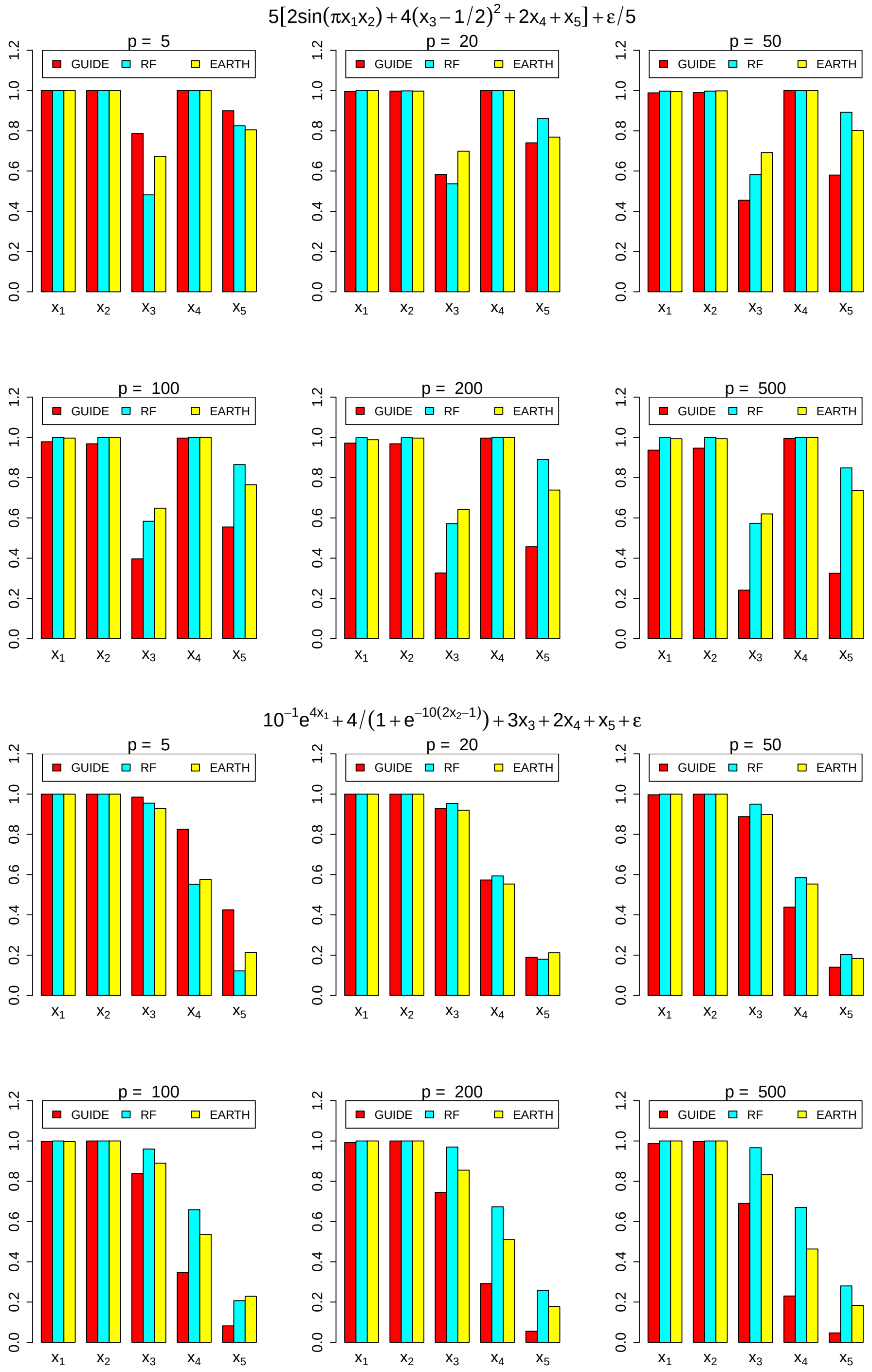


Fig. 2 Variable selection probabilities; x_i independent; simulation SE < 0.02

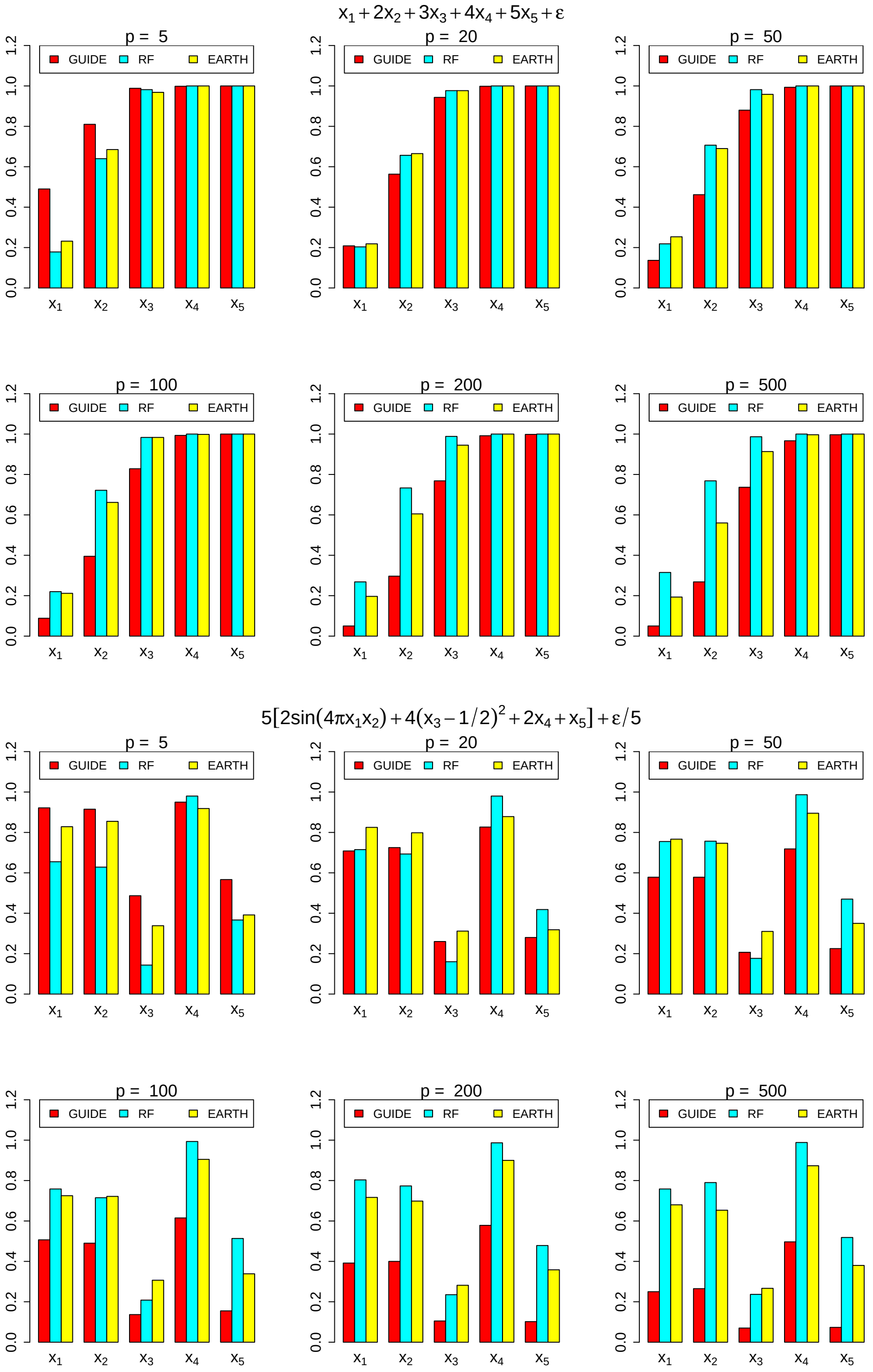


Fig. 3 Variable selection probabilities; x_i independent; simulation SE < 0.02 (cont'd.)

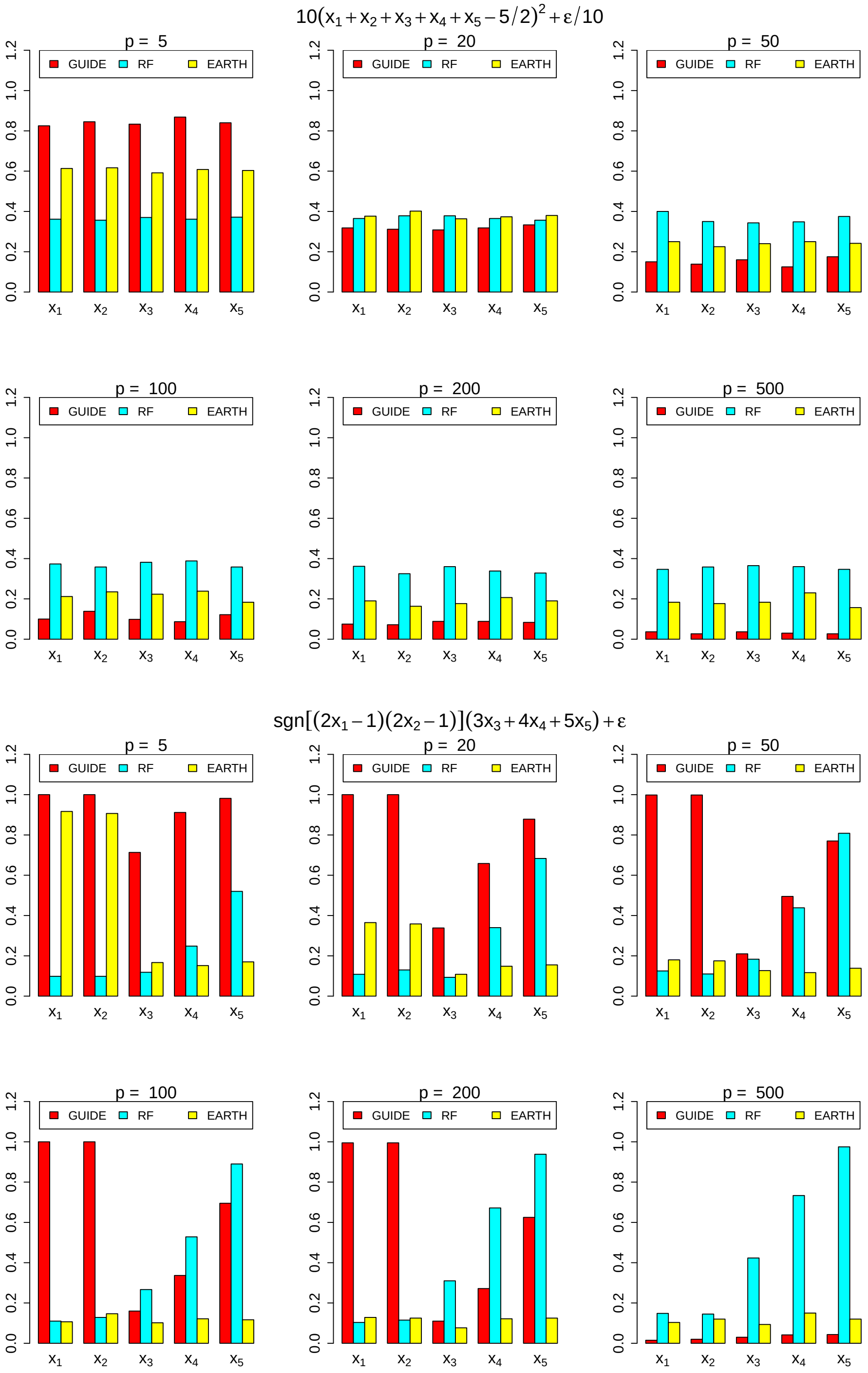


Fig. 4 Variable selection probabilities; x_i independent; simulation SE < 0.02 (cont'd.)

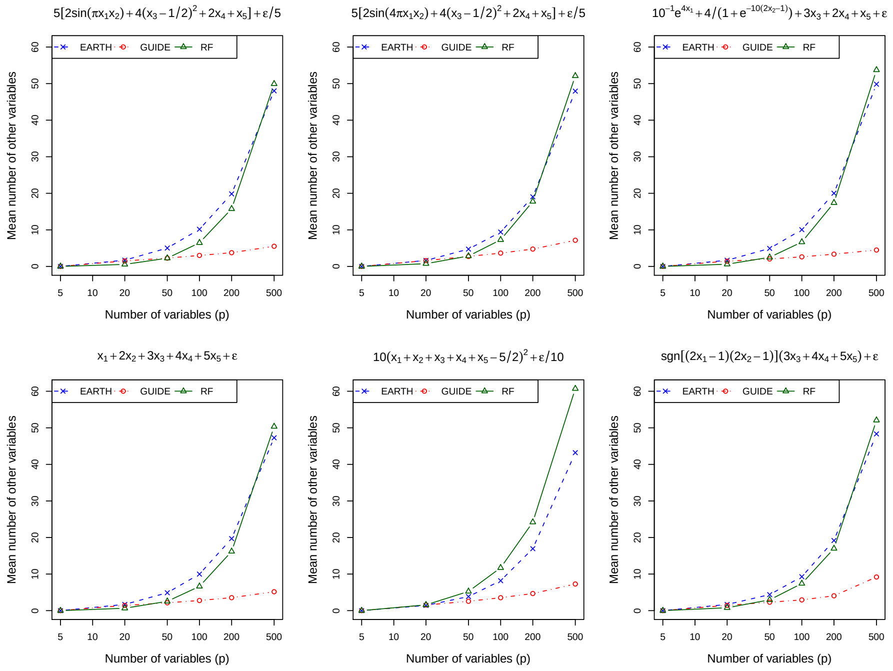


Fig. 5 Mean number of unimportant variables selected; x_i mutually independent

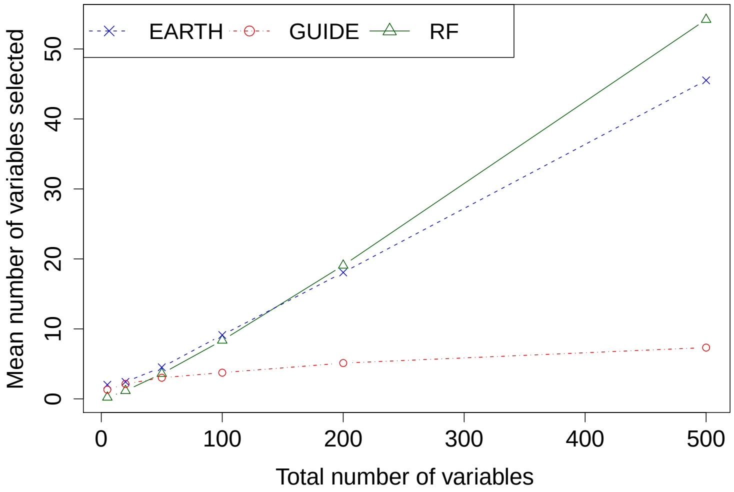


Fig. 6 Mean number of variables selected for the constant model $y = \varepsilon$ and mutually independent x variables

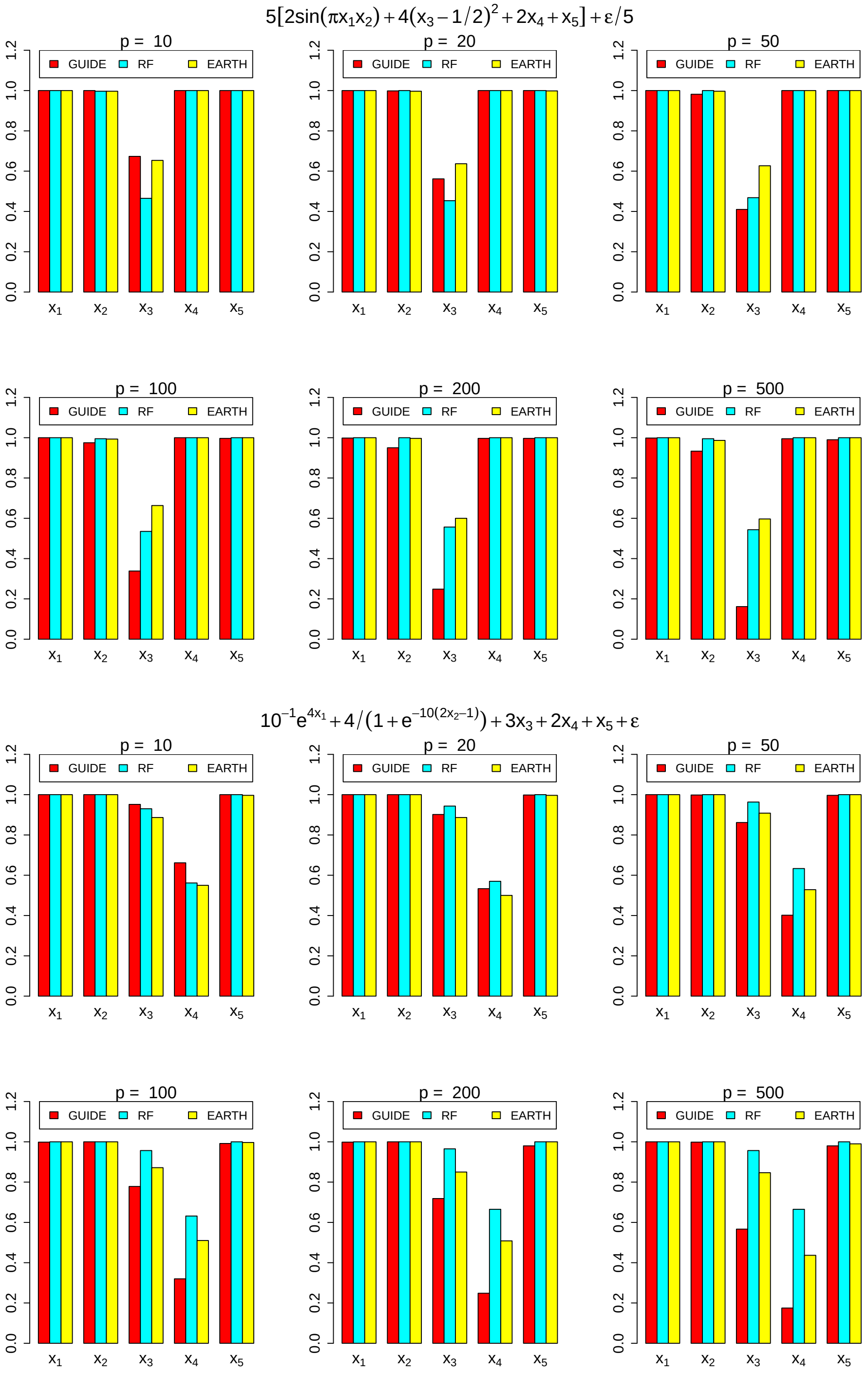


Fig. 7 Variable selection probabilities; x_i dependent; simulation SE < 0.02

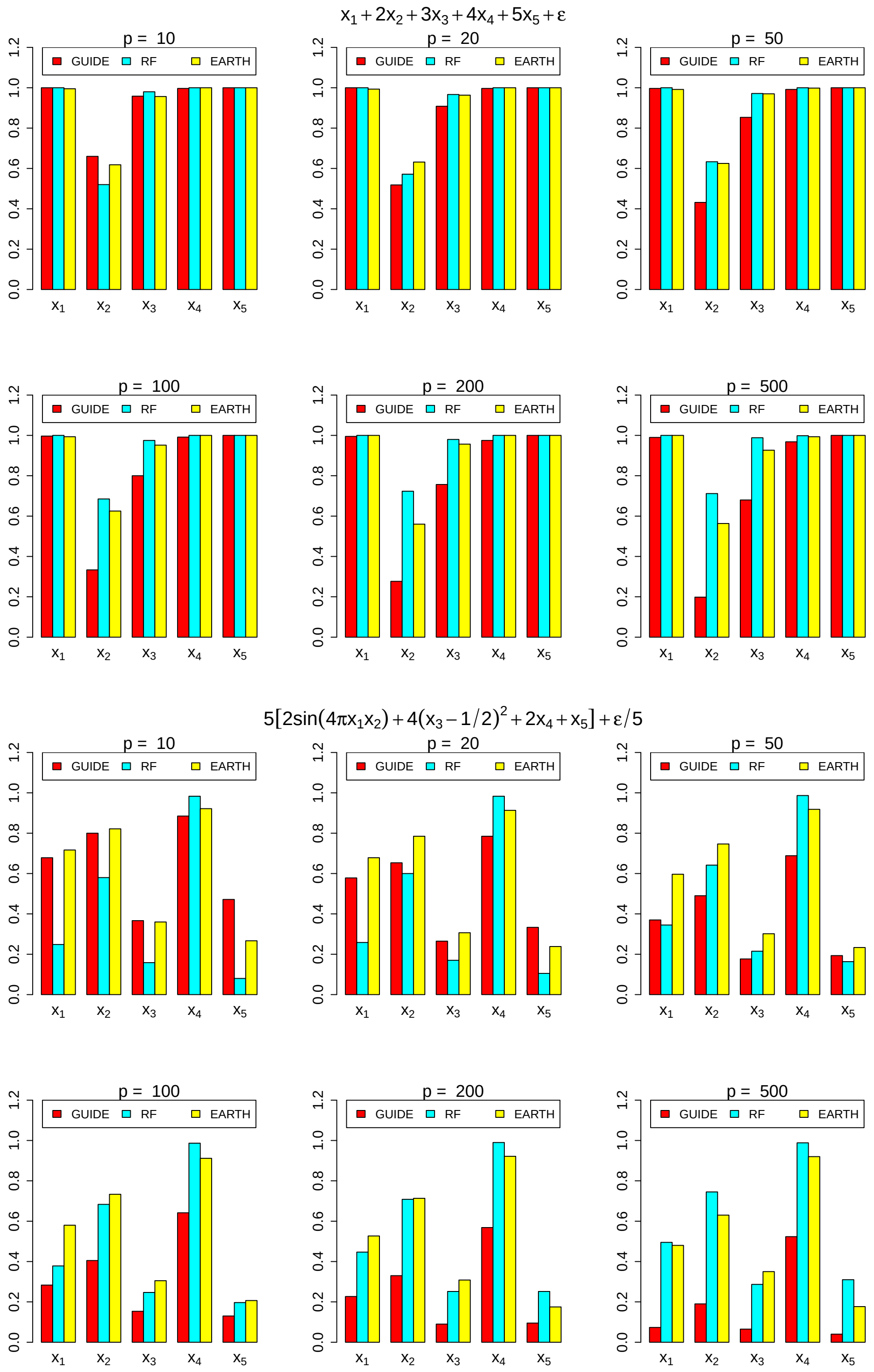


Fig. 8 Variable selection probabilities; x_i dependent; simulation SE < 0.02 (cont'd.)

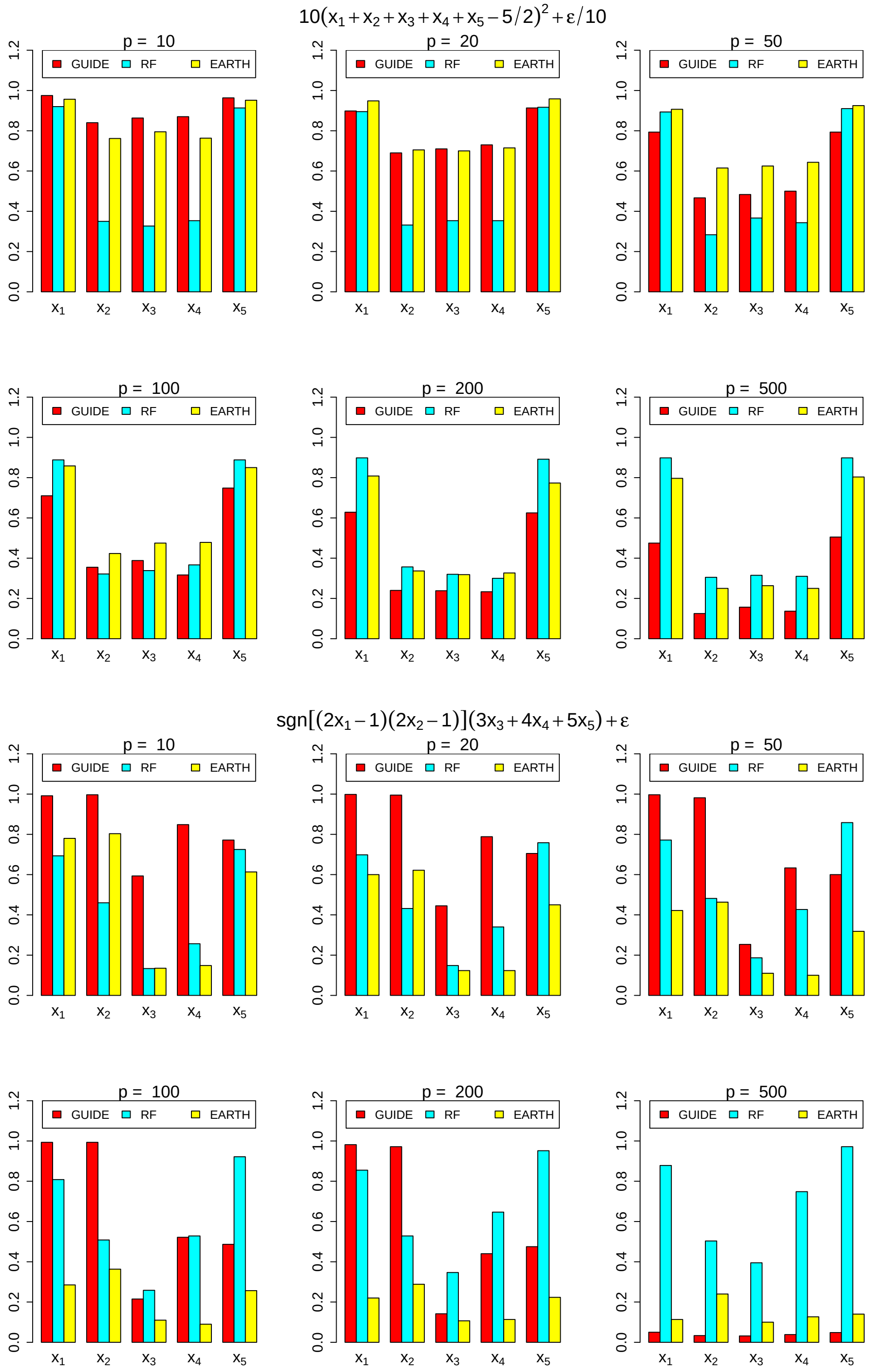


Fig. 9 Variable selection probabilities; x_i dependent; simulation SE < 0.02 (cont'd.)

3 Expected squared error

Because increasing the probability of selecting the important variables inevitably leads to more unimportant ones being chosen, a better way to compare the variable selection methods is in terms of their effect on prediction error. Figure 10 shows the simulated expected squared errors of GUIDE, MARS, and Random forest with (solid lines) and without (dashed lines) each of the three variable selection methods, for the constant model with mutually independent predictor variables. The training sample size is 100, test sample size is 1000, and $p = 5, 20, 50, 100, 200, 500$. Owing to its lengthy computation time, the results for EARTH when $p = 500$ are based on 300 simulation trials; the others are based on 600 trials. Simulation standard errors are less than 0.015.

The results show that the expected squared error of MARS is reduced substantially by all three variable selection methods, with the GUIDE selection method giving the greatest reduction. On the other hand, all three variable selection methods increase slightly the expected squared error of Random forest, although its values are already low to begin with. The GUIDE selection method is the only one that reduces the expected squared error of the GUIDE fitting method for all values of p—see the middle panel of Figure 10.

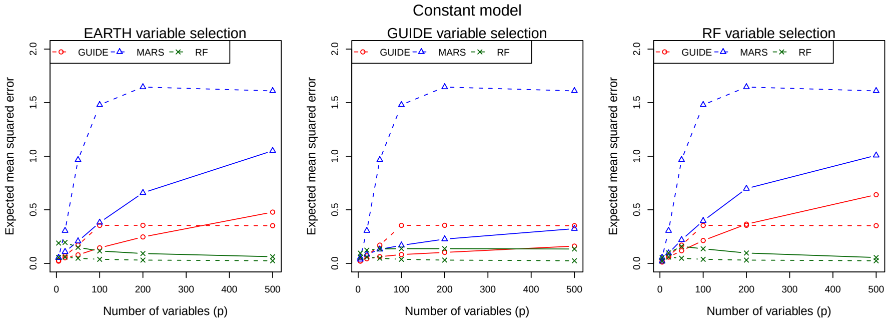


Fig. 10 Expected squared errors for the constant model $y = \varepsilon$, with ε standard normal and independent predictors; simulation SE < 0.015. Dashed and solid lines indicate before and after variable selection.

Figures 11 and 12 give the corresponding results for the non-constant models (1)–(6). The conclusions are similar: the GUIDE selection method tends to reduce the expected squared error of all three regression methods more than the EARTH and Random forest selection methods. Figures 13 and 14 show the results when the x_i variables have the dependence structure in (8). Again the GUIDE selection method is more effective than EARTH and Random forest in reducing the expected squared error of all three regression methods. Figure 15 shows the computation times (in sec.) required by each method for each model and various values of p. EARTH is by far the most time consuming and GUIDE is the least. Further, the computation time of EARTH increases with p much faster than that of the other two methods.

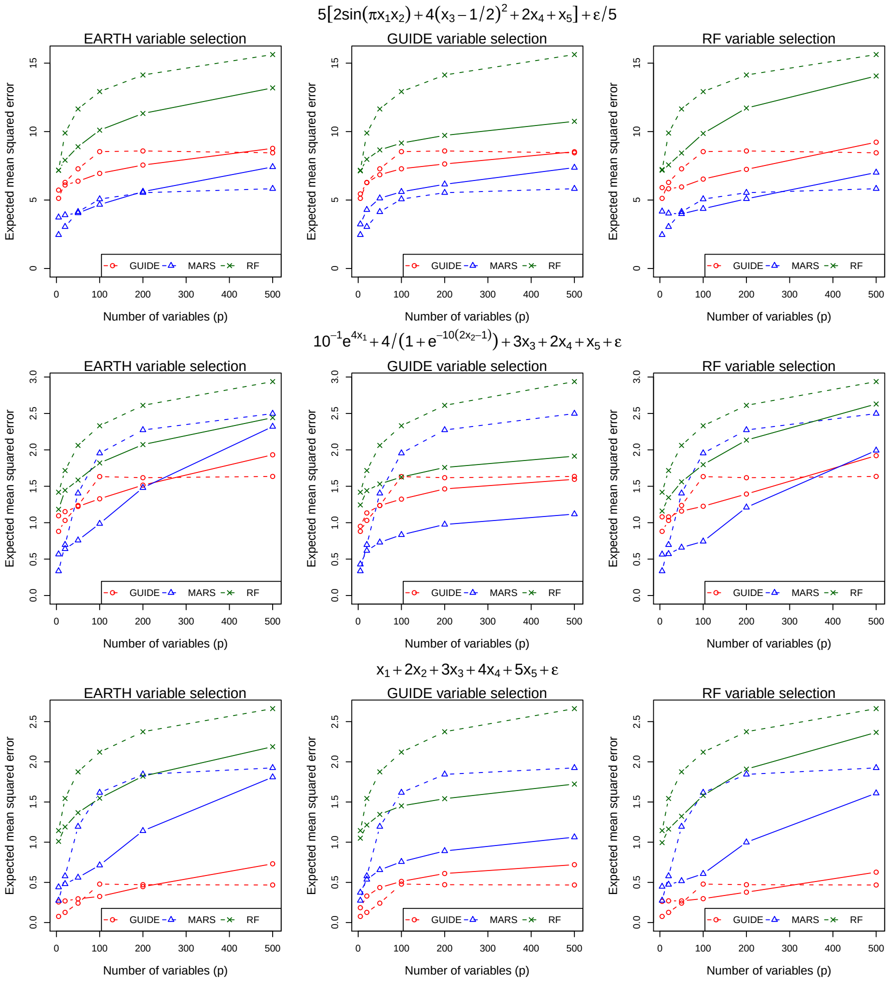


Fig. 11 expected squared errors for models 1–3, with ε standard normal and independent predictors. Dashed and solid lines correspond to before and after variable selection. Simulation error bars are too small to be shown.

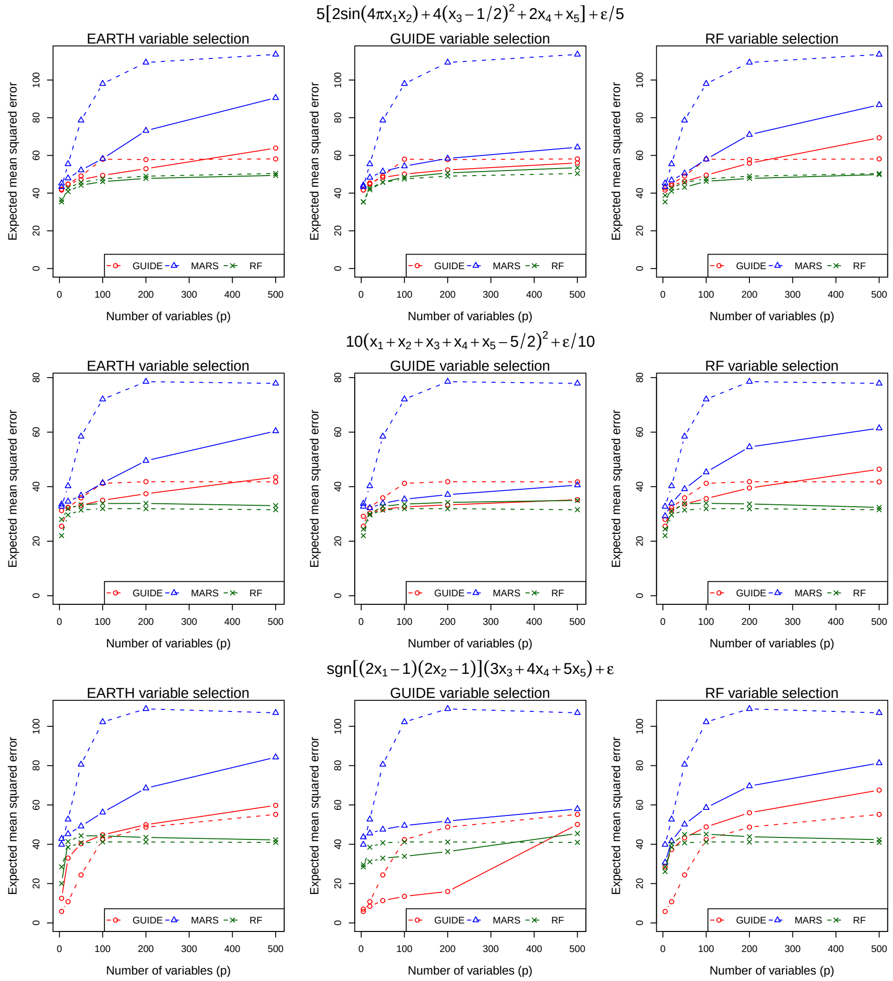


Fig. 12 expected squared errors for models 4–6, with ε standard normal and independent predictors. Dashed and solid lines correspond to before and after variable selection. Simulation error bars are too small to be shown.

4 Some theory for linear models

It is natural to expect a variable selection procedure to degrade the performance of a fitting method if there are no unimportant variables in the data. Careful inspection of Figures 11 and 12 shows, however, that all three variable selection methods (GUIDE, EARTH and Random forest) decrease the expected squared error of Random forest in all six simulation models even for $p = 5$, where every variable is

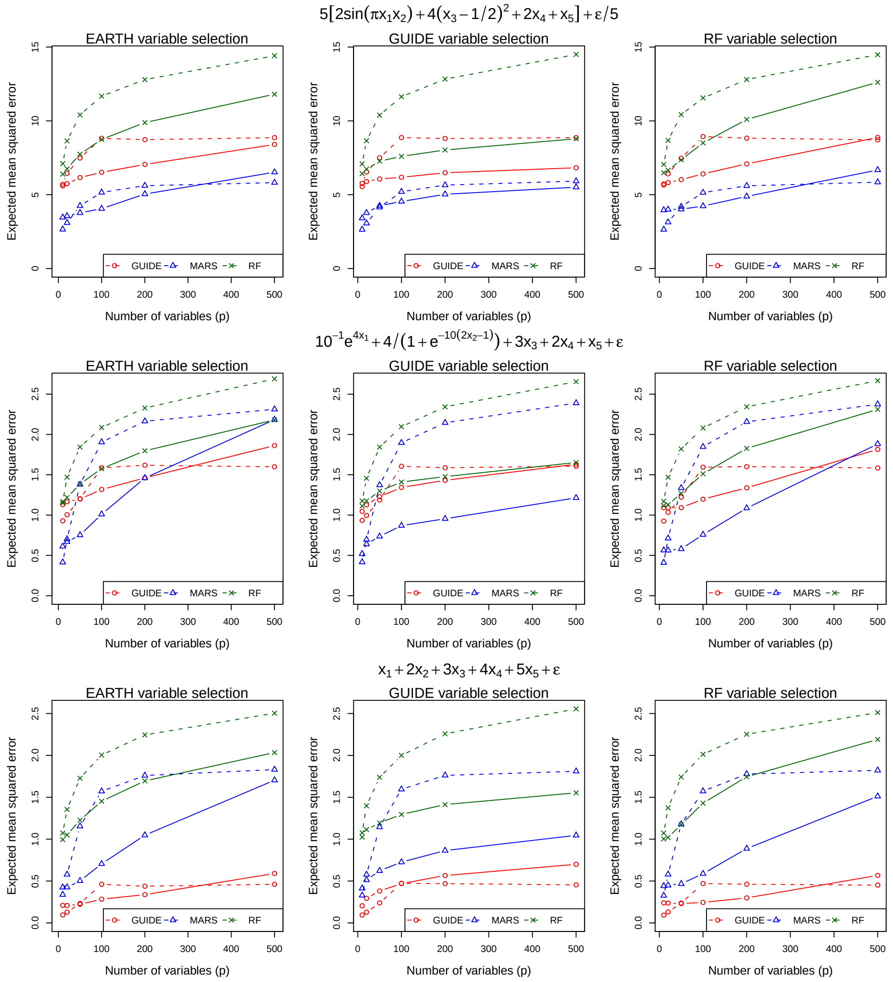


Fig. 13 expected squared errors for models 1–3, with ε standard normal and dependent predictors. Dashed and solid lines correspond to before and after variable selection. Simulation error bars are too small to be shown.

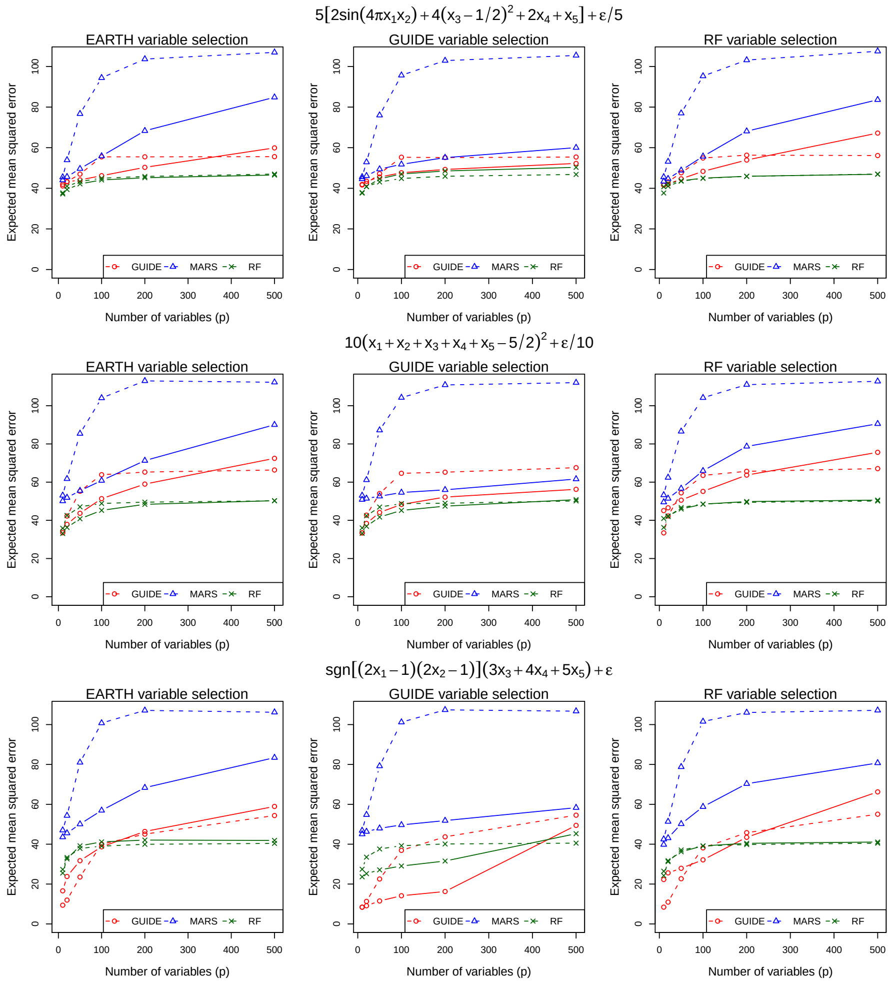
important! This rather counter-intuitive behavior can be shown to occur in linear models too.

Let β_i be a p_i-dimensional vector and $\mathbf{X}_i$ an $n \times p_i$-dimensional matrix, for $i = 1, 2, 3$, such that

$$\mathbf{y} = \mathbf{X}_1\beta_1 + \mathbf{X}_2\beta_2 + \mathbf{X}_3\beta_3 + \varepsilon. \tag{9}$$

Assume throughout that

$$\beta_3 = \mathbf{0} \tag{10}$$



$5[2\sin(4\pi x_1 x_2) + 4(x_3 - 1/2)^2 + 2x_4 + x_5] + \varepsilon/5$

$10(x_1 + x_2 + x_3 + x_4 + x_5 - 5/2)^2 + \varepsilon/10$

$\mathrm{sgn}[(2x_1 - 1)(2x_2 - 1)](3x_3 + 4x_4 + 5x_5) + \varepsilon$

Fig. 14 expected squared errors for models 4–6, with ε standard normal and dependent predictors. Dashed and solid lines correspond to before and after variable selection. Simulation error bars are too small to be shown.

that is, the variables in $\mathbf{X}_3$ are unimportant. The correct model is then

$$\mathbf{y} = \mathbf{X}_1\beta_1 + \mathbf{X}_2\beta_2 + \varepsilon. \tag{11}$$

Let $\mathbf{Z}_2 = (\mathbf{X}_1, \mathbf{X}_2)$ and $\beta = (\beta_1, \beta_2)'$, with least squares estimate $\hat{\beta} = (\mathbf{Z}_2'\mathbf{Z}_2)^{-1}\mathbf{Z}_2'\mathbf{y}$. Let $\mathbf{x}_i$ be a p_i-dimensional vector, for $i = 1, 2, 3$. The mean of $\mathbf{y}$ at $(\mathbf{x}_1', \mathbf{x}_2')'$ is $\mu = \mathbf{x}_1'\beta_1 + \mathbf{x}_2'\beta_2$ with least-squares estimate $\hat{\mu}_0 = (\mathbf{x}_1', \mathbf{x}_2')\hat{\beta}$. For $i = 2$ and 3, define $\mathbf{H}_1 = \mathbf{X}_1(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'$, $\mathbf{L}_i = (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'\mathbf{X}_i$, and $\mathbf{M}_i = (\mathbf{X}_i'(\mathbf{I} - \mathbf{H}_1)\mathbf{X}_i)^{-1}$. Then (see,

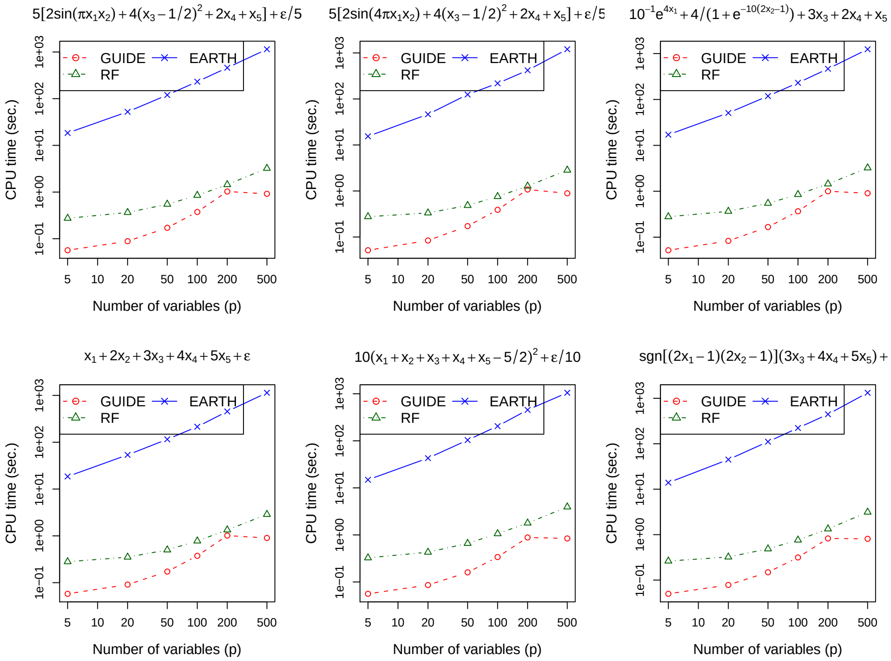


Fig. 15 Variable selection computation time per data set plotted on log scales

e.g., [9, p. 231])

$$(\mathbf{Z}_2'\mathbf{Z}_2)^{-1} = \begin{pmatrix} (\mathbf{X}_1'\mathbf{X}_1)^{-1} + \mathbf{L}_2\mathbf{M}_2\mathbf{L}_2' & -\mathbf{L}_2\mathbf{M}_2 \\ -\mathbf{M}_2\mathbf{L}_2' & \mathbf{M}_2 \end{pmatrix}.$$

Let $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)$. The expected squared error is

$$\begin{aligned} E(\hat{\mu}_0 - \mu)^2 &= E[\mathrm{Var}\{(\mathbf{x}_1', \mathbf{x}_2')\hat{\beta} \mid \mathbf{X}, \mathbf{x}_1, \mathbf{x}_2\}] \\ &= \sigma^2 E\left\{ (\mathbf{x}_1', \mathbf{x}_2')(\mathbf{Z}_2'\mathbf{Z}_2)^{-1} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \right\} \\ &= \sigma^2 E\left\{ (\mathbf{x}_1', \mathbf{x}_2') \begin{pmatrix} (\mathbf{X}_1'\mathbf{X}_1)^{-1} + \mathbf{L}_2\mathbf{M}_2\mathbf{L}_2' & -\mathbf{L}_2\mathbf{M}_2 \\ -\mathbf{M}_2\mathbf{L}_2' & \mathbf{M}_2 \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \right\} \\ &= \sigma^2 E\left\{ (\mathbf{x}_1', \mathbf{x}_2') \begin{pmatrix} (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{x}_1 + \mathbf{L}_2\mathbf{M}_2(\mathbf{L}_2'\mathbf{x}_1 - \mathbf{x}_2) \\ -\mathbf{M}_2(\mathbf{L}_2'\mathbf{x}_1 - \mathbf{x}_2) \end{pmatrix} \right\} \\ &= \sigma^2 E\{\mathbf{x}_1'(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{x}_1 + (\mathbf{L}_2'\mathbf{x}_1 - \mathbf{x}_2)'\mathbf{M}_2(\mathbf{L}_2'\mathbf{x}_1 - \mathbf{x}_2)\}. \end{aligned} \tag{12}$$

Suppose that we mistakenly exclude $\mathbf{X}_2$ and include $\mathbf{X}_3$ instead. That is, we fit the incorrect model

$$\mathbf{y} = \mathbf{X}_1\beta_1 + \mathbf{X}_3\beta_3 + \varepsilon. \tag{13}$$

Let $\mathbf{Z}_3 = (\mathbf{X}_1, \mathbf{X}_3)$. The estimated mean at $(\mathbf{x}_1', \mathbf{x}_2', \mathbf{x}_3')'$ is $\hat{\mu}_1 = (\mathbf{x}_1', \mathbf{x}_3')(\mathbf{Z}_3'\mathbf{Z}_3)^{-1}\mathbf{Z}_3'\mathbf{y}$ and

$$\begin{aligned}
\hat{\mu}_1 - \mu &= (\mathbf{x}_1', \mathbf{x}_3')(\mathbf{Z}_3'\mathbf{Z}_3)^{-1}\mathbf{Z}_3'\mathbf{y} - \mathbf{x}_1'\boldsymbol{\beta}_1 - \mathbf{x}_2'\boldsymbol{\beta}_2 \\
&= (\mathbf{x}_1', \mathbf{x}_3')(\mathbf{Z}_3'\mathbf{Z}_3)^{-1}\mathbf{Z}_3'(\mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{X}_2\boldsymbol{\beta}_2 + \varepsilon) - \mathbf{x}_1'\boldsymbol{\beta}_1 - \mathbf{x}_2'\boldsymbol{\beta}_2 \\
&= (\mathbf{x}_1', \mathbf{x}_3') \begin{pmatrix} (\mathbf{X}_1'\mathbf{X}_1)^{-1} + \mathbf{L}_3\mathbf{M}_3\mathbf{L}_3' & -\mathbf{L}_3\mathbf{M}_3 \\ -\mathbf{M}_3\mathbf{L}_3' & \mathbf{M}_3 \end{pmatrix} \begin{pmatrix} \mathbf{X}_1' \\ \mathbf{X}_3' \end{pmatrix} (\mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{X}_2\boldsymbol{\beta}_2 + \varepsilon) \\
&\quad - \mathbf{x}_1'\boldsymbol{\beta}_1 - \mathbf{x}_2'\boldsymbol{\beta}_2 \\
&= (\mathbf{x}_1', \mathbf{x}_3') \begin{pmatrix} (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1' + \mathbf{L}_3\mathbf{M}_3(\mathbf{L}_3'\mathbf{X}_1' - \mathbf{X}_3') \\ -\mathbf{M}_3(\mathbf{L}_3'\mathbf{X}_1' - \mathbf{X}_3') \end{pmatrix} (\mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{X}_2\boldsymbol{\beta}_2 + \varepsilon) \\
&\quad - \mathbf{x}_1'\boldsymbol{\beta}_1 - \mathbf{x}_2'\boldsymbol{\beta}_2 \\
&= \{\mathbf{x}_1'(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1' + (\mathbf{x}_1'\mathbf{L}_3 - \mathbf{x}_3')\mathbf{M}_3(\mathbf{L}_3'\mathbf{X}_1' - \mathbf{X}_3')\}(\mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{X}_2\boldsymbol{\beta}_2 + \varepsilon) \\
&\quad - \mathbf{x}_1'\boldsymbol{\beta}_1 - \mathbf{x}_2'\boldsymbol{\beta}_2 \\
&= \{\mathbf{x}_1'\mathbf{L}_2 - \mathbf{x}_2' + (\mathbf{x}_1'\mathbf{L}_3 - \mathbf{x}_3')\mathbf{M}_3(\mathbf{L}_3'\mathbf{X}_1' - \mathbf{X}_3')\mathbf{X}_2\}\boldsymbol{\beta}_2 \\
&\quad + \{\mathbf{x}_1'(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1' + (\mathbf{x}_1'\mathbf{L}_3 - \mathbf{x}_3')\mathbf{M}_3(\mathbf{L}_3'\mathbf{X}_1' - \mathbf{X}_3')\}\varepsilon
\end{aligned}$$

where we use the identity $\mathbf{L}_3'\mathbf{X}_1'\mathbf{X}_1 = \mathbf{X}_3'\mathbf{X}_1$. Therefore its expected squared error is

$$\begin{aligned}
E(\hat{\mu}_1 - \mu)^2 &= E[\{\mathbf{x}_1'\mathbf{L}_2 - \mathbf{x}_2' + (\mathbf{x}_1'\mathbf{L}_3 - \mathbf{x}_3')\mathbf{M}_3(\mathbf{L}_3'\mathbf{X}_1' - \mathbf{X}_3')\mathbf{X}_2\}\boldsymbol{\beta}_2]^2 \\
&\quad + \sigma^2 E\left[\{\mathbf{x}_1'(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1' + (\mathbf{x}_1'\mathbf{L}_3 - \mathbf{x}_3')\mathbf{M}_3(\mathbf{L}_3'\mathbf{X}_1' - \mathbf{X}_3')\}\right. \\
&\quad \left. \times \{\mathbf{X}_1(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{x}_1' + (\mathbf{X}_1\mathbf{L}_3 - \mathbf{X}_3)\mathbf{M}_3(\mathbf{L}_3'\mathbf{x}_1 - \mathbf{x}_3)\}\right] \\
&= E[\{\mathbf{x}_1'\mathbf{L}_2 - \mathbf{x}_2' + (\mathbf{x}_1'\mathbf{L}_3 - \mathbf{x}_3')\mathbf{M}_3(\mathbf{L}_3'\mathbf{X}_1' - \mathbf{X}_3')\mathbf{X}_2\}\boldsymbol{\beta}_2]^2 \\
&\quad + \sigma^2 E[\mathbf{x}_1'(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{x}_1 + (\mathbf{L}_3'\mathbf{x}_1 - \mathbf{x}_3)'\mathbf{M}_3(\mathbf{L}_3'\mathbf{x}_1 - \mathbf{x}_3)]
\end{aligned}$$

and the increase in expected squared error is

$$\begin{aligned}
&E(\hat{\mu}_1 - \mu)^2 - E(\hat{\mu}_0 - \mu)^2 = \\
&\quad E[\{(\mathbf{L}_2'\mathbf{x}_1 - \mathbf{x}_2)' + (\mathbf{L}_3'\mathbf{x}_1 - \mathbf{x}_3)'\mathbf{M}_3(\mathbf{L}_3'\mathbf{X}_1' - \mathbf{X}_3')\mathbf{X}_2\}\boldsymbol{\beta}_2]^2 \\
&\quad + \sigma^2 E[(\mathbf{L}_3'\mathbf{x}_1 - \mathbf{x}_3)'\mathbf{M}_3(\mathbf{L}_3'\mathbf{x}_1 - \mathbf{x}_3) - (\mathbf{L}_2'\mathbf{x}_1 - \mathbf{x}_2)'\mathbf{M}_2(\mathbf{L}_2'\mathbf{x}_1 - \mathbf{x}_2)]. \qquad (14)
\end{aligned}$$

Consider the following three situations:

1. **Underfitting.** Suppose that $p_3 = 0$. Then $\mathbf{X}_3, \mathbf{L}_3$ and $\mathbf{M}_3$ vanish and

   $$E(\hat{\mu}_1 - \mu)^2 - E(\hat{\mu}_0 - \mu)^2 = E[(\mathbf{L}_2'\mathbf{x}_1 - \mathbf{x}_2)'\boldsymbol{\beta}_2]^2 - \sigma^2 E[(\mathbf{L}_2'\mathbf{x}_1 - \mathbf{x}_2)'\mathbf{M}_2(\mathbf{L}_2'\mathbf{x}_1 - \mathbf{x}_2)].$$

   Thus $E(\hat{\mu}_1 - \mu)^2 < E(\hat{\mu}_0 - \mu)^2$ if and only if

   $$E[(\mathbf{L}_2'\mathbf{x}_1 - \mathbf{x}_2)'\boldsymbol{\beta}_2]^2 < \sigma^2 E\{(\mathbf{L}_2'\mathbf{x}_1 - \mathbf{x}_2)'\mathbf{M}_2(\mathbf{L}_2'\mathbf{x}_1 - \mathbf{x}_2)\}. \qquad (15)$$

   Further,

$$\begin{aligned}\frac{E(\hat{\mu}_1-\mu)^2}{E(\hat{\mu}_0-\mu)^2} &= 1+\frac{E[(\mathbf{L}_2'\mathbf{x}_1-\mathbf{x}_2)'\beta_2]^2-\sigma^2E[(\mathbf{L}_2'\mathbf{x}_1-\mathbf{x}_2)'\mathbf{M}_2(\mathbf{L}_2'\mathbf{x}_1-\mathbf{x}_2)]}{\sigma^2E\{\mathbf{x}_1'(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{x}_1+(\mathbf{L}_2'\mathbf{x}_1-\mathbf{x}_2)'\mathbf{M}_2(\mathbf{L}_2'\mathbf{x}_1-\mathbf{x}_2)\}} \\ &\to 1-\frac{E[(\mathbf{L}_2'\mathbf{x}_1-\mathbf{x}_2)'\mathbf{M}_2(\mathbf{L}_2'\mathbf{x}_1-\mathbf{x}_2)]}{E\{\mathbf{x}_1'(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{x}_1+(\mathbf{L}_2'\mathbf{x}_1-\mathbf{x}_2)'\mathbf{M}_2(\mathbf{L}_2'\mathbf{x}_1-\mathbf{x}_2)\}}\end{aligned}$$

as $\beta_2 \to \mathbf{0}$. If $p_2 = 1$, i.e., β_2 is real-valued, condition (15) reduces to

$$\beta_2^2E[(\mathbf{L}_2'\mathbf{x}_1-\mathbf{x}_2)^2] < \sigma^2E[\mathbf{M}_2(\mathbf{L}_2'\mathbf{x}_1-\mathbf{x}_2)^2].$$

Figure 16 shows a graph of the ratio of expected squared errors as a function of β_2/σ for $p_1 = 5,10,30,50,70,90$, $p_2 = 1$, $n = 100$, the first predictor variable being 1 and the other predictors independent and uniformly distributed on the unit interval. The ratios are estimated by simulation with 1000 test samples and 1000 simulation trials, yielding simulation standard errors less than 0.01. We see that the threshold value of β_2/σ for which underfitting is advantageous increases with p_1.

2. **Overfitting.** Suppose instead that $p_2 = 0$. Then β_2, $\mathbf{X}_2, \mathbf{L}_2$, and $\mathbf{M}_2$ vanish and the increase in expected squared error is non-negative because $\mathbf{M}_3$ is positive definite and $E(\hat{\mu}_1-\mu)^2-E(\hat{\mu}_0-\mu)^2 = \sigma^2E[(\mathbf{L}_3'\mathbf{x}_1-\mathbf{x}_3)'\mathbf{M}_3(\mathbf{L}_3'\mathbf{x}_1-\mathbf{x}_3)] \geq 0$.
3. **Under and overfitting.** Suppose that $p_2 = p_3$ and the distribution of $(\mathbf{x}_1,\mathbf{x}_2)$ is the same as that of $(\mathbf{x}_1,\mathbf{x}_3)$. Then the increase in expected squared error is always positive, because

$$\begin{aligned}&E(\hat{\mu}_1-\mu)^2-E(\hat{\mu}_0-\mu)^2 \\ &= E[\{(\mathbf{L}_2'\mathbf{x}_1-\mathbf{x}_2)'+(\mathbf{L}_3'\mathbf{x}_1-\mathbf{x}_3)'\mathbf{M}_3(\mathbf{L}_3'\mathbf{X}_1'-\mathbf{X}_3')\mathbf{X}_2\}\beta_2]^2.\end{aligned}$$

5 Application to real data

We now compare the variable selection methods in an application to quantitative high-throughput screening of the enzyme pyruvate kinase. The data, obtained from the National Chemical Genomics Center (NCGC), consist of measurements on $p = 5444$ chemical properties (x variables) of 46,229 compounds. Each compound is also measured for its level of inhibition (y variable) of the biological activity of pyruvate kinase. A compound is considered to be an inhibitor if $y < -5$. Figure 17 shows a histogram of the y values; only one percent of the compounds are inhibitors. Our goals are: (i) to identify the chemical properties that are predictive of an inhibitor and (ii) to use this information to predict whether a new compound is an inhibitor. We employ ten-fold cross-validation to compare the methods. That is, we randomly divide the data set into ten roughly equal parts, use each part in turn as the training set to identify the important variables and to build a prediction model, and then use the other nine tenths as a test set to assess the accuracy of the

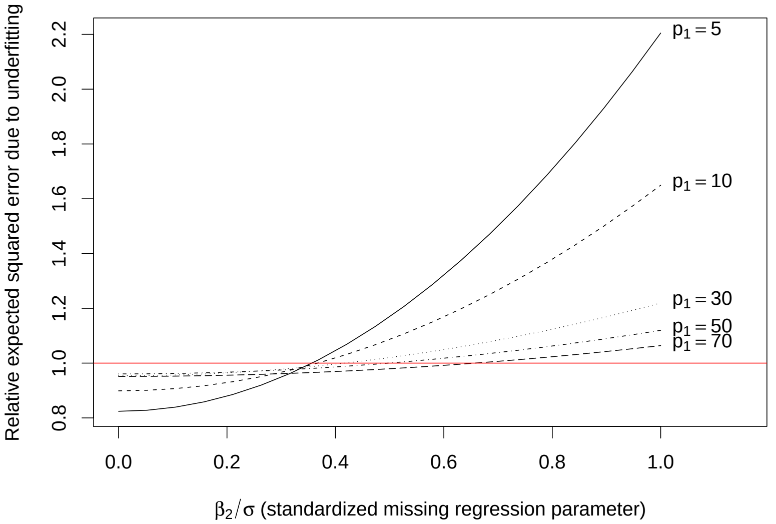


Fig. 16 Simulated values of $E(\hat{\mu}_1 - \mu)^2/E(\hat{\mu}_0 - \mu)^2$ versus β_2/σ for $p_2 = 1, p_3 = 0$ and different values of p_1; simulation standard errors are less than 0.01.

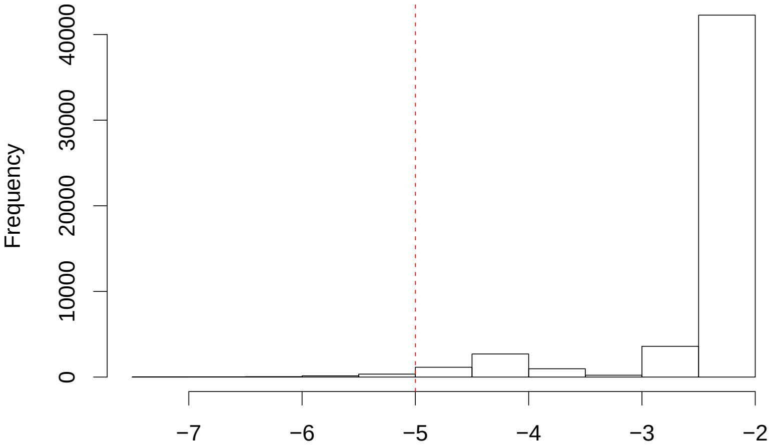


Fig. 17 Histogram of biological activity levels of 46,229 compounds. A compound is an inhibitor if its level is below -5.

predictions. Thus the number of compounds, n, in each training set is approximately 4623, which is less than p.

First, we treat this as a regression problem, i.e., we use our GUIDE and Random forest variable selection methods to identify the important variables and then apply three different nonparametric regression methods (GUIDE piecewise-linear regression tree, MARS, and Random forest) to the selected variables to predict the test-sample y values. Figure 18 shows boxplots of the ten cross-validation mean

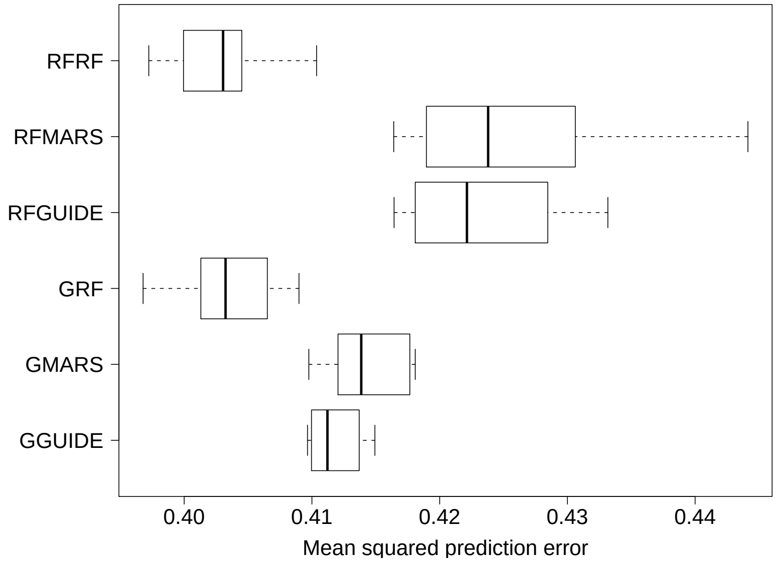


Fig. 18 Boxplots of cross-validation expected squared errors. GGUIDE, GMARS, and GRF refer to GUIDE variable selection followed by GUIDE piecewise-linear, MARS, and Random forest model fitting. Similarly, RFGUIDE, RFMARS, and RFRF refer to Random forest variable selection followed by GUIDE piecewise-linear, MARS, and Random forest model fitting.

Table 1 Average cross-validation results for NCGC data; smaller values are better

Variable selection method	Number of variables selected	Mean squared prediction error		
		GUIDE tree	MARS	Random forest
GUIDE	331	0.412	0.414	0.403
Random forest	225	0.423	0.426	0.403
		Mean rank of inhibitor		
		GUIDE forest	Stepwise logistic	Random forest
GUIDE	34	9181	9874	10102
Random forest	470	10374	10699	12015

squared prediction errors of the six methods. The top half of Table 1 gives their average as well as the average number of variables identified as important. GUIDE chooses about 50% more variables than Random forest (331 vs. 225). For variable selection, GUIDE is as good or better than Random forest, but the latter is best for model fitting. The differences are, however, less than 5 percent.

High accuracy in predicting y does not imply high accuracy in predicting whether a compound is an inhibitor. Since the latter is a classification problem, consider a binary response variable that takes value 1 if $y < -5$ and 0 otherwise. The problem is then the estimation of the probability, $P(y < -5)$, that a compound is an inhibitor, for which stepwise logistic regression offers a ready solution. Some sort of variable

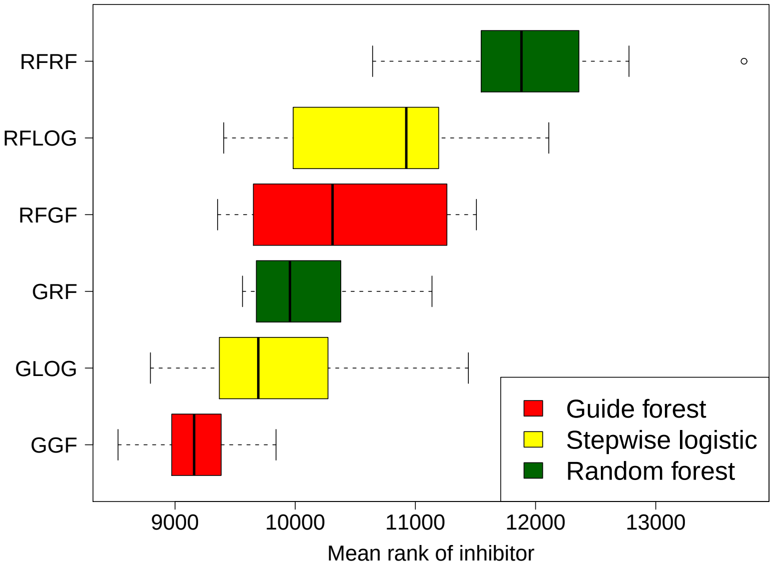


Fig. 19 Boxplots of cross-validation mean rank of inhibitors. GGF, GLOG, and GRF refer to GUIDE variable selection followed by GUIDE forest, stepwise logistic, and Random forest model fitting. Similarly, RFGF, RFLOG, and RFRF refer to Random forest variable selection followed by GUIDE forest, stepwise logistic, and Random forest model fitting. Methods with small mean ranks are better.

selection is needed, however, because $n < p$. Since the Random forest and GUIDE variable selection methods are applicable to classification problems, we use them to do this. After the variables are selected, we fit a stepwise logistic regression model to the training sample and use it to estimate the probability of an inhibitor for each compound in the test sample. We also employ prediction models constructed by Random forest and GUIDE forest. The latter is an ensemble method similar to Random forest except that the GUIDE classification tree algorithm is used to split the nodes of the trees. This yields a total of six combination methods—two variable selection methods crossed with three model fitting methods. Given a compound in the test sample, each combination method yields an estimated probability that it is an inhibitor. We rank the test compounds in decreasing order of these probabilities and take the average of the ranks of the true inhibitors among them. Thus small values of the average ranks are indicative of high prediction accuracy.

Figure 19 shows boxplots of the ten cross-validation mean ranks for the six combination methods. GUIDE variable selection is consistently better than Random forest in improving the prediction of all three fitting methods. Among fitting methods, GUIDE forest is better than stepwise logistic regression, which in turn is better than Random forest. The bottom half of Table 1 gives the average of the ten cross-validation mean ranks as well as the mean number of variables selected for each method. Random forest selects on average fourteen times as many variables as GUIDE (470 vs. 34).

Table 2 Average computation time (min.) for one cross-validation iteration on a 2.66 Ghz Intel Core 2 Quad Extreme processor with 8 GB memory

	Selection	Selection	GUIDE		Random
	method	time	tree	MARS	forest
Regression	GUIDE	0.54	8.20	0.48	5.87
	Random forest	32.18	1.38	0.17	1.14
	Selection	Selection	GUIDE	Stepwise	Random
	method	time	forest	logistic	forest
Classification	GUIDE	1.13	7.01	0.87	0.17
	Random forest	48.48	53.87	185.37	3.31

Table 2 shows the average computation time for each variable selection and model fitting method for both the regression and classification problems. GUIDE variable selection is 40–60 times faster than Random forest variable selection: 0.54 vs. 32.18 min. for regression and 1.13 vs. 48.48 min. for classification. For regression model fitting, MARS is much faster than both Random forest and GUIDE piecewise-linear tree. For classification, Random forest is fastest. Stepwise logistic regression is faster than GUIDE forest when there are few variables (0.87 min. when GUIDE is the selection method) but its speed rapidly decreases when the number of variables is large (185.37 min. when Random forest is the selection).

6 Conclusion

We introduced a variable selection method for use prior to application of any classification and regression fitting algorithm. Because the method is a by-product of the GUIDE algorithm, it is applicable to all kinds of data, including categorical and non-categorical response and predictor variables as well as data with missing values. We compared the method with EARTH and Random forest in terms of their probabilities of selecting the important variables in simulated regression models. The results show that the new method is as good as or better than the other two when there are few unimportant variables. When there are numerous unimportant variables, the probability that the new method selects the important variables is much lower than that of EARTH and Random forest. The higher detection rates of the latter two methods are, however, accompanied by correspondingly higher false positive detection rates. For example, if the true regression model is a constant, EARTH and Random forest have false positive rates of about ten percent compared to about one percent for the new method.

High false positive rates can adversely affect the prediction accuracy of the fitted models. We demonstrated this by coupling each of the three variable selection methods with each of three regression fitting methods: MARS, Random forest and GUIDE piecewise-linear tree. Our simulation results show that while all three fitting methods generally benefit from prior variable selection, the new selection method tends to offer the greatest benefit. Further, the new method requires much less computation time than EARTH and Random forest.

One explanation for the greater effectiveness of the new method in reducing the prediction error of fitting algorithms may be its lower false positive detection rate. We support this conjecture by showing that in the case of a linear model with some variables having weak effects and no unimportant variables, an under-fitted model can possess lower expected squared error than a fully fitted one.

We also compared the new method with Random forest on a real data set with so many predictor variables that variable selection is a necessary step before model fitting. We analyzed the data twice, first as a regression problem and then as a classification problem. In the case of regression, the new method is more effective than Random forest selection in reducing the mean squared prediction error of MARS and GUIDE piecewise-linear regression tree models, but it is less effective when applied to the Random forest model. On the other hand, the new method consistently beats Random forest selection across all three fitting methods for the classification problem. In terms of computation time, the new method also is substantially faster than Random forest.

Acknowledgment

This research was partially supported by the U.S. Army Research Office under grants W911NF-05-1-0047 and W911NF-09-1-0205. The author is grateful to K. Doksum, S. Tang, and K. Tsui for helpful discussions and to S. Tang for the computer code for EARTH.

References

1. Breiman, L. (2001). Random forests, *Machine Learning* **45**: 5–32.
2. Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. (1984). *Classification and Regression Trees*, Wadsworth, Belmont, California.
3. Chernoff, H., Lo, S.-H. & Zheng, T. (2009). Discovering influential variables: a method of partitions, *Annals of Applied Statistics* **3**: 1335–1369.
4. Doksum, K., Tang, S. & Tsui, K.-W. (2008). Nonparametric variable selection: the EARTH algorithm, *Journal of the American Statistical Association* **103**: 1609–1620.
5. Friedman, J. (1991). Multivariate adaptive regression splines (with discussion), *Annals of Statistics* **19**: 1–141.
6. Loh, W.-Y. (2002). Regression trees with unbiased variable selection and interaction detection, *Statistica Sinica* **12**: 361–386.
7. Loh, W.-Y. (2009). Improving the precision of classification trees, *Annals of Applied Statistics* **3**: 1710–1737.
8. Satterthwaite, F. E. (1946). An approximate distribution of estimates of variance components, *Biometrics Bulletin* **2**: 110–114.
9. Seber, G. A. F. & Lee, A. J. (2003). *Linear Regression Analysis*, 2nd edn, Wiley.
10. Strobl, C., Boulesteix, A., Zeileis, A. & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution, *BMC Bioinformatics* **8**.

11. Tuv, E., Borisov, A. & Torkkola, K. (2006). Feature selection using ensemble based ranking against artificial contrasts, *IJCNN ’06. International Joint Conference on Neural Networks*, Vancouver, Canada.