Using Cross-Validation, Probing, And Lasso In Gradient Boosting Variable Selection

Tahir R. Dikheel[1], **Shahad H. Alwa**[1*]

[1]Department of Statistics, University of Al-Qadisiyah, Iraq, Email: tahir.dikheel@qu.edu.iq,

[*]**Corresponding Author:** Shahad H. Alwa
[*]Department of Statistics, University of Al-Qadisiyah, Iraq, Email: shahad6b@gmail.com

Abstract
Combining the predictions of various models often results in a model with increased predictive performance in numerous problem domains. Recently, machine learning methods such as boosting method, have been widely used in many scientific fields. Boosting is an example of a method that has shown a lot of potentials. on the practical side, Experimental studies have demonstrated that combining models using boosting methods creates more accurate regression models. it was presented methods for variable selection dependent on model-based gradient boosting, Model-based boosting is a method that fits a statistical model and selection variables. Certain machine learning algorithms handle high-dimensional data processing to improve data visualization. by comparing the three methods (lasso, probing, and cross-validation) dependent on the value of MSE, the probing has the lowest MSE so it prefers. The simulation and the real-data example both indicate that the proposed method (probing) outperforms the other current methods.

Keywords: Cross-validation, Probing, variable selection, machine learning, gradient boosting, lasso.

Introduction

The fast advancement of computer technology in recent decades has resulted in several of the new and increasingly computationally intensive statistical data analysis methodologies emerging from the area of machine learning.

Machine learning algorithms make a framework based on sample data, referred to as "training data," and then use it to make predictions or judgments without having to be explicitly programmed to do so (Clarke, et al.2008). Furthermore, Popular statistical regression techniques, such as ordinary least squares, are unable to estimate model coefficients at many of these states cause of the uniqueness of the covariance matrix. We explore statistical techniques for the analysis of high-dimensional data, with a strong reliance on recent developments including such gradient boosting algorithms, which fall under the category of "Model-based boosting algorithms". Gradient boosting algorithms have begun to emerge as one of the most important methods in modern medical research because they integrate a sophisticated ensemble optimization technique developed in the field of machine learning with classical regression modeling Boosting algorithms are among the most promising data analysis methodological techniques that have been created in the last two decades. (Mayr, et al. 2014).

The initial algorithm emerged from machine learning, where it quickly gained popularity and was regarded as a potent tool for predicting events. Applied in statistical modeling, where it could be used to choose with assessing predictors' impact on a single dependent variable in many regression situations. Gradient boosting is one type of boosting. The gradient boosting algorithms are regulated by setting hyperparameters that govern the degree of penalization (Friedman, et al.2001). While resampling methods such as cross-validation and other related models are routinely used to calculate these hyperparameters. We suppose that p is an amount bigger than n, as indicated as p> n. Almost all of the time, we study a situation in which there are more covariables than n, such as a linear model.

$$Y = X\beta + \varepsilon_i \qquad (1)$$

Where $Y = (Y1, \ldots, Yn)^{\mathrm{T}}$ is a response vector that is univariate, The nxp design matrix is represented by X. Column kth includes the covariable, Xk = ($X_1^k, \ldots, X_n^k$)τ, and ε_i =(ε1, ..., ε_n)T is the error (noise) term, with independent and identically distributed (i.i.d.) components.

$$E(\varepsilon_i) = 0 \text{ and } \mathrm{Var}\ (\varepsilon_i) = \sigma^2\mathcal{E} \qquad (2)$$

When p> n, classical statistical methods, such as ordinary least squares (OLS). estimation, Could not be utilized for estimate β, σ2ε as they would overfit to data and generate serious identifiability problems (Bühlmann, et al. 2014). Equation 1 estimate of a high-dimensional linear model with p>n necessitates an amount of optimization. The construction of p values that regulate some type I error measure with having a high power for spotting alternatives, is a difficulty in high-dimensional models (avoiding some type II error). The simplicity with which statistical boosting algorithms may well be interpreted is one of the reasons for their success (Mayr, et al.2014).

Can use an interpretable function E $(Y \mid X = x)$ = f(x), the statistical model seeks to quantify the link between one or more observable predictor variables as well as the expected outcome E (Y). When there are multiple predictors, the effects of the individual variables are often put together to generate an additive model:

$$f(X)=\beta 0+h1(x1) + \ldots + hp(xp) \qquad (3)$$

Where β0 is an intercept and h1(.),..., hp(.) combine the impacts of predictors x1,...,xp that are components of X, as well as the idea is to apply the link-function g (.) to model the expected value of the dependent variables based on observed predictors.

$$g\big(E(Y|X = x)\big) = \beta o + \textstyle\sum_{k-1}^{p} h_p\ (x_p) \qquad (4)$$

For example, h1 represents the partial influence of predictor x1 (.).

The stopping iteration, typically abbreviated as "mstop," is the most important parameter for boosting algorithms and tune. the initial choice of probing is used as a criterion for stopping (Thomas, et al.. 2017). Determining the correct stopping iteration of a boosting algorithm is critical since it reduces data overfitting and often increases prediction accuracy.

2. Gradient Boosting

Gradient boosting is an efficient strategy for estimating and selecting predictor effects in various regression models by applying concepts from the field of statistical learning. It's a regression issue machine learning technique. The basic idea of Boosting is to integrate simple rules to build an ensemble in which each ensemble member's efficiency is increased or boosted (converting many weak learners to form a single strong learner). Given a learning problem with a data set D= (Xi, Yi) i=1,...,n observations i.i.d. from a distribution over the joint space X×Y, the p-dimensional input space $X = (X_1 \times X_2 \cdot\cdot\cdot\times$ Xk) and an output space Y(e.g., $Y = R$). The target of regression is to find a function, f(x), $X \rightarrow Y$, which maps as many variables in the feature space as feasible to the output items (Thomas, et al.2018).

Gradient boosting algorithms seek to minimize a specified loss function, ρ (yi, f(xi)), which measures the difference between a predicted result value of f(xi) and a true (yi), based on the perceptions of boosting in function space as gradient descent. A disparity is minimized by fitting ineffective prediction functions, known as base learners, on previous errors repeatedly in integrating them into a powerful ensemble (Xingyu, 2016). Model-based gradient boosting (Algorithm 1).

The technique iteratively updates a prediction using a tiny fraction of the base learner with either the perfect fit on the loss function's negative gradient. (Thomas, et al.. 2017), assuming at m = 0 w and a constant loss, the minimal initial value is $\widehat{f^{(0)}}\ \ (x) = c$:

(1) Make m = m + 1 the iteration counter.

(2) When $m \leq$ mstop, calculate the negative gradient vector of the loss function.:

$$u^{(i)} = -\frac{\partial\rho(y,f)}{\partial f}\bigg|_{f'=f^{\complement[m-1]}(x^{(i)}),y=y^{(i)}} \qquad (5)$$

(3) Accommodate each base learner hk $[m]$(xk) to a negative gradient vector u individually.

(4) Determine $\hat{h}$k $[m]$ (xk$*$), any a best-fitting base learner:

$$j^* = \underset{l \le k \le P}{\arg min} \sum_{i=1}^{n} \left(u^{(i)} - \hat{h}_k^{[m]}\left(x_k^{(i)}\right)\right)^2 \qquad (6)$$

(5) Replace a tiny fraction $0 \le v \le 1$ of this portion in the predictor:

$$\hat{f}(x)^{[m]} = \hat{f}(x)^{[m-1]} + v.\hat{h}^{[m]}(x \quad) \qquad (7)$$

The algorithm minimizes the following empirical risk during these steps.

$$\frac{1}{N}\sum_{i=1}^{N} \rho\left(y\ ,f(X)\right). \qquad (8)$$

The challenge of estimating a regression function f (.) for a statistical model, which links the predictor variables (X) to the outcome (Y), is described as follows:

$$\hat{f}(.) = \underset{f(.)}{\arg min} \left\{E_{\underline{Y,X}}[\rho(Y, f(X))]\right\} \qquad (9)$$

The L2 loss function is the most typical, which is ρ(y, f (.)) = (y – f (.))2, which leads to classical least squares regression of the mean: $f(x) = E(Y/X = x)$.

In practice, the empirical risk is reduced by using a learning sample of observations (y1, x1),..., (yn, xn):

$$\hat{f}\,(\,.\,) = \arg_{f(.)} \min\ \frac{1}{N}\sum_{i=1}^{N} \rho^i \left(y^i\ ,f(x\ )\right). \qquad (10)$$

3. The least absolute shrinkage and selection operator (lasso)

A closely analogous approach recently introduced for linear regression issues is the lasso. It can generate sparse models and be used for both estimation and variable selection. In practice, this method is often adjusted to reach the best forecast accuracy. the prediction accuracy is employed as the criterion for selecting the tuning parameter, the approach is in general inconsistent in terms of variable selection. That is, the variable sets chosen are not consistent in identifying the genuine set of essential variables (Leng, et al. 2006). Because the optimization problem is convex, one appealing characteristic of the lasso is its computational feasibility for big p. In addition, the lasso can select variables via shrinking some estimated coefficients to 0.

$$\sum_{i=1}^{n} \left(y_j - \sum_j x_{ij}^{ik}\beta_j\right)^2 + \lambda \sum_{j=1}^{P} \left|\beta_j^k\right| \qquad (10)$$

Tibshirani offered the lasso as a possible option (1996). Despite the differences in methodological methods, the lasso like gradient boosting can mimic the outcomes of standard linear regression models when used of low dimension p < N settings. The lasso simultaneously performs continuous shrinkage and automatic variable selection (Hepp, et al. 2016). As variable selection becomes more crucial in current data analysis, the lasso representation is becoming more appealing owing to its sparse representation. It is a penalized least squares method that disincentivizes the regression coefficients with an L1 penalty . The use of lasso as a primary variable selection strategy has grown in popularity in empirical finance in recent years (Sohrabi, and Movaghari. 2020).

Although the lasso has proven to be effective in a variety of situations, it has some limitations:

a) In the p > n situation, the lasso chooses at most n variables before saturating owing to the convex optimization problem. unless the bound on the L1 norm of a coefficient is less than a certain value, the lasso is not properly characterized.

(b) If a collection of variables has a high pairwise correlation, the lasso will tend to pick only one variable from the group, regardless of which one it is.

(c) It has been empirically shown that the lasso prediction performance is dominated by ridge regression in typical n>p cases when there are high correlations between variables.

In some cases, scenarios (a) and (b) make the lasso an ineffective variable selection strategy. As a result, the lasso prediction power can be improved further. While the optimization parameter in it changed from zero to one, distinct regression parameters remain non-zero (Tarr, et al. 2015). When the tuning parameter is λ large enough in lasso, however, the L1 penalty has the effect of driving some of the coefficient estimations to be exactly equal to zero. Consequently, the lasso approach likewise produces sparse models. Moreover, it is computationally efficient, and it selects the true model as the sample size n increases. As the value of λ rises, the value of coefficients decreases, lowering the variance. This increase in λ is good because it merely reduces variance (thus avoiding overfitting) while preserving all of the data's critical features.

4.Cross- validation

Cross-validation is perhaps the simplest and perhaps most extensively apply a method for estimating prediction error. This method directly calculates the excess sample error Err= $E\ [L\ (Y, k^{*}(X))]$, which is the generalization. error when the method k*(X) is implemented to an independent test. sample from the joint. distribution of X and Y (Friedman, et al. 2001). It is one of the most extensively used resampling methods.

The essential notion behind cross-validation is that part of the data is applied. to fit the model, while the rest is used to evaluate the model that has been built. V-fold cross-validation divides the data set into V equal or nearly equal pieces at random (Friedman, et al. 2001). V-fold cross-validation employs a portion of the available data to fit the model and a different portion to test it to fine-tune the problem. For the cross-validation estimate of prediction error, we separated the data into V approximately equivalent parts:

$$CV = \frac{1}{N}\sum_{i=1}^{N} L\left(y_i\ , \hat{f}^{-V(i)}(x_i\ )\right). \tag{12}$$

It is one of the most common resampling approaches, and it is getting a lot of attention and being used a lot for variable selection. Typical V values are 5 or 10, with V = N referred to as leave-one-out cross-validation. It should be noticed that methods such as recurrent cross-validation assist in the stabilization of findings by absorbing the impact of particularly bad splits, frequently resulting in slight model complexity reductions as well. (Hepp, et al. 2016).

5. Probing

A proposed method for determining the appropriate number of iterations in model-based boosting for variable selection that is inspired by probing, a technique commonly used in machine learning and microarray analysis. The main idea behind probing is to artificially inflate data using random noise variables, often known as probes or shadow variables. It is especially appealing for usage with more computationally costly boosting techniques because it doesn't require any resampling (Thomas, et al.2017).

When the probing concept is applied to the sequential structure of model-based gradient variable selection through stability selection or cross-validation, one model fit is sufficient to uncover relevant variables, and no costly model refitting is necessary. Unlike classical cross-validation there is no need for any pre-specification such as the required to evaluate (mstop) for cross-validation; instead, probing focuses on optimal variable selection rather than the algorithm's prediction performance. Because this usually entails halting considerably sooner, the effect estimates for the selected variables are likely to be heavily regularized, making them unsuitable for forecasting.

Probing for variable selection in model-based boosting (Algorithm 2):

(1) Create randomly shuffled images $\tilde{x}$k for each of the k=1,. . . , p in the data set X variables xk such that $\tilde{x} \in$ Sxk , with Sxk denoting the symmetric group containing all n! xk possible permutations.

(2) Establish a boosting model based on the megascopic data set, X= [x1 · · · xp $\tilde{x}$1 · · .$\tilde{x}p$], and start to begin repeats m = 0.

(3) If the first $\tilde{x}$j is selected. (Algorithm1, step3)

fit each base learner hk [m](xk) to a negative gradient vector u individually

(4) Only the variables from the original data collection X should be returned. (Thomas, et al. 2017).

Simulation

We undertake benchmark simulation research to evaluate the performance of variable selection methods, in which we compare three strategies (probing, lasso, cross validation) note their results, and simulate n data points from a multivariate. normal distribution for p variables.

consider the following generate $n \times p$, X matrix with standard normal distribution with mean=0, covariance identity matrix σ; X∼N(0, σI). The used model which we depend on can be written as.y=Xβ+Ɛ.

Where Y is a matrix n×1, X is a matrix with n×p, and β= is a regression coefficient We repeat the experiment R= 500. To estimate the model using three methods (lasso, probing, and cross-validation). Where variables P={ 20, 100, 400}, sample size n={ 50, 150, 250}, and we choose S= (3, 7,9) (S) refer to significant variables in the model, which are the important variables.

Case 1 (for sparse)

$$S=9\ ,\ \beta=\{\underbrace{2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2}_{9}\ldots\underbrace{0\ 0\ldots0\ 0\ 0}_{p-9}\}.$$

Table 1. Mean square error for methods when S= 9

N	P	MSE	
50	20	Lasso	0.8868
		Probing	0.6529
		C.V	0.7169
	100	Lasso	1.4058
		Probing	0.2206
		C.V	0.6267
	400	Lasso	1.3483
		Probing	0.0347
		C.V	0.2358
150	20	Lasso	1.0578
		Probing	0.8740
		C.V	0.8952
	100	Lasso	1.0974
		Probing	0.4687
		C.V	0.7484
	400	Lasso	1.1963
		Probing	0.1474
		C.V	0.4557
250	20	Lasso	1.0605
		Probing	0.9202
		C.V	0.9325
	100	Lasso	1.0983
		Probing	0.6654
		C.V	0.8445
	400	Lasso	1.1567
		Probing	0.3519
		C.V	0.7088

Shown in a table (1) are the results of comparing three methods (lasso, probing, cross validation). The probing method is more stable, it had the lowest MSE value. So probing is the best method and it is performing comparatively better than lasso and cross validation. Note the probing method was decrease when the number of variables (p) increased for the stable number of sample sizes (N), and the MSE value of the probing method increased when the increasing number of (N) for the same number of p. In addition to the MSE value of cross validation (CV) method decrease for the increasing number of p for the same value of N, while the MSE value for the cross validation method increase for the same number of (p) and the increasing number of (N).

In addition to the lasso which had the highest value of MSE. It increases when the number of variables (p) increases for the same numbers of sample size (N), and it is decreased when the number of (N) increases for the same number of (p). Therefore probing method was preferred because it had the smallest MSE value compared to the other two methods.

Case 2 (for sparse)

$$S=7\ ,\ \beta=\{\underbrace{2\ 2\ 2\ 2\ 2\ 2\ 2}_{7}\ \ldots\underbrace{0\ 0\ldots0\ 0\ 0}_{p-7}\}.$$

Table 2. Mean square error for methods when S= 7

N	p	MSE	
50	20	Lasso	1.0807
		Probing	0.6267
		C.V	0.6446
	100	Lasso	1.1376
		Probing	0.2926
		C.V	0.4087
	400	Lasso	1.9091
		Probing	0.0696
		C.V	0.7122
150	20	Lasso	1.0829
		Probing	0.8636
		C.V	0.8931
	100	Lasso	1.1129
		Probing	0.4730
		C.V	0.7252
	400	Lasso	1.2186
		Probing	0.1312
		C.V	0.5674
250	20	Lasso	1.0631
		Probing	0.9166
		C.V	0.9335
	100	Lasso	1.1029
		Probing	0.6574
		C.V	0.8631
	400	Lasso	1.1710
		Probing	0.3312
		C.V	0.7815

Shown in Table (2) the results of comparing three methods (lasso, probing, cross validation), that probing is the best method and it is performing comparatively better than lasso and cross validation because it had the lowest MSE value. We notice the value of probing decrease when the number of variables (p) increases for the same sample size (N) and the value of probing increase when the number of(N) increase for the same number of variables(p). the MSE value of the cross validation (cv) method decreases for the increasing number of (p) for the same value of (N). While the MSE value for the cross validation method increases for the same number of (p) and increasingly of (N). while the lasso method which has the highest MSE value increases when the number of variables P increase for the same numbers of sample size (N), and it is decreased when the number of (N) increases for the same number of (p). Therefore, it prefers probing because it had the smallest MSE value comparing the other two methods.

Case 3 (for very sparse)

S= 3

$$\beta = \{\underbrace{2\ \ 2\ \ 2}_{3}\ \ .\,.\,.\ \underbrace{0\ \ 0\ .0\ \ldots\ .0\ 0\ \ 0}_{p-3}\}.$$

Table 3. Mean square error for methods when S = 3

N	P	MSE	
50	20	Lasso	1.1634
		Probing	0.6095
		C.V	0.7354
	100	Lasso	1.2917
		Probing	0.0745
		C.V	0.4982
	400	Lasso	1.3410
		Probing	0.0380
		C.V	0.3664
150	20	Lasso	1.1121
		Probing	0.8661
		C.V	0.9283
	100	Lasso	1.1455
		Probing	0.4555
		C.V	0.8536
	400	Lasso	1.1975
		Probing	0.1128
		C.V	0.7873
250	20	Lasso	1.0830
		Probing	0.9177
		C.V	0.9535
	100	Lasso	1.1093
		Probing	0.6533
		C.V	0.9215
	400	Lasso	1.1271
		Probing	0.2880
		C.V	0.8817

results of comparing three methods (lasso, probing, cross validation). Probing had the lowest MSE value, so probing is the best method and it is performing comparatively better than lasso and cross validation. It appears that the MSE for the probing method decrease when the number of variables (p) increase for the stable sample size (N), and the MSE value of the probing method increase when the increasing number of (N) for the same number of (p). As well as the MSE value of the cross validation (CV) method decrease for the increasing number of (p) for the same value of (N). While the MSE value for the cross validation method increases for the same number of (p) and the increasing number of (N). In addition to the lasso which had the highest value of MSE. , it increases when the number of variables (p) increases for the same numbers of sample size(N), and it is decreased when the number of (N) increases for the same number of (p). Therefore probing method was preferred because it had the smallest MSE value comparing the other two methods.

Furthermore, the value of MSE of the probing decreases as the number of variables increases for the same sample size, whereas the value of MSE of the probing increases as the sample size increases for the same number of variables. The value of MSE for cross-validation decreases as the (p) increases, and it increases when (N) increases. lasso was increased when variables(p) increase, and it decrease when (N). As a result, we conclude that increasing the number of variables (p) improves the probing method's effectiveness. As a result, it prefers the probing method because it has the smallest MSE.

Cardiovascular disease data

The proposed methods are useful for selection variables relevant and estimating parameters. The comparing between methods in terms of variable selection and accurate prediction. The performance methods are illustrated using Data. These data were collected from a Nasiriyah Heart Center in Nasiriyah City, Iraq.

They represent the number of people with Cardiovascular disease in Nasiriyah City from the first half of 2021. These data consist of 100 observations, 50 independent variables, and one response variable(Red Blood cell). The selected dataset Includes medical analyses and personal and social information about the study population serving as covariates. The Comparison depends on MSE as shown in table 4 below. the probing method is the best performance because it has the lowest value (33.8432). Thus, probing is the favorite method while lasso, cross-validation having sequentially (39.8250, 37.9251).

Table 4. Mean square error for real data to n = 100, p= 50

Methods	MSE
Lasso	39.8250
Probing	33.8432
C.V	37.9251

All approaches rely on model-based boosting algorithms. the probing method is the best performance because it has the lowest value (33.84321); thus, probing is the favorite method while lasso, cross-validation having sequentially (39.82509, 37.92518). To evaluate the results provided by the three ' approaches', (lasso, probing, and cross validation) to analysis the data to compare. Table 5 shows the total number of variables selected by each method along with the size of the intersection between the sets.

Table 5. Total number of selected variables with size n = 100 and p = 50 for three variable selection techniques (probing, lasso, and cross-validation) on expression data sets.

X	Lasso	Probing	cross-validation
X3		0.000131204	
X4		0.004330695	
X5		0.012762972	
X6		-0.005323548	
X8		0.00438118	3.233429e-04
X9		-0.015856054	
X10		-0.003887381	
X11		-1.501926e-02	-3.262778e-03
X12		-0.002617102	
X13		-0.00044107	
X14		-0.002794923	
X15		-0.007333231	
X18		0.000380884	
X20		-0.011486408	
X21		-0.018976889	
X22		0.010743992	
X23		0.021598493	5.735875e-03
X24		-0.002150323	
X25	-0.02898	-0.029844264	-2.984426e-02
X28		0.000674936	2.150974e-04
X29		-0.001401589	
X30		-0.000290187	
X31	-0.01534	-0.022305315	-2.230531e-02
X33		-1.800042e-03	
X34	0.022819	1.508288e-02	1.508288e-02
X35		-3.933573e-04	-1.147322e-04
X36		5.445344e-04	4.441935e-04
X37	0.013315	0.019711519	1.971152e-02
X39		0.00013824	
X40		-6.658958e-04	
X42		-0.001079428	-4.474280e-04
X43		0.002782754	
X44		-0.009478171	
X46	-0.01603	-0.009425567	-9.425567e-03
X47	-0.00162	-0.002223108	-9.385160e-04
X49	0.045793	0.041523771	4.152377e-02
X50		0.000838157	
X51		-1.001479e-02	

For the suggested methods and the other methods in comparison, as indicated in the table (5). We can see that the lasso only chose the variables (x25, x31, x34, x37, x46, x47, x49). As seen in figure 2, the smallest "mean square error" correlates to a relatively big (λ), which may make it difficult to choose predictors.

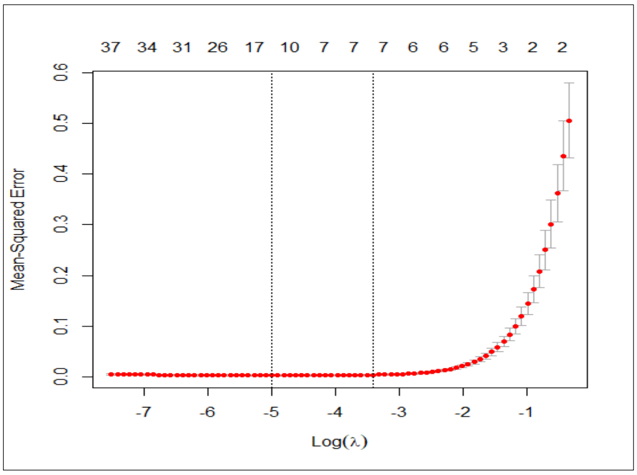

Figure 1. generating the beta sparse coefficient paths.

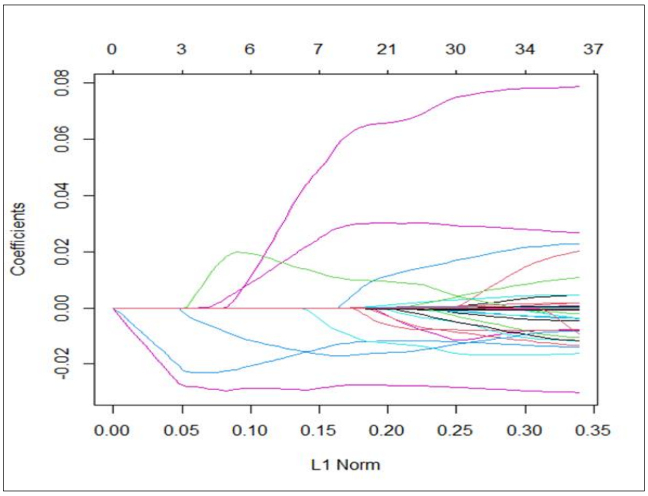

Figure 2. Minimum "mean square error" corresponds to large (λ)

We want to choose the best coefficients that have small mean square error. While probing selected (x3, x4, x5, x6, x8, x9, x10, x11, x12, x13, x14, x15,x18,x20,x21,x22,x23,x24,x25, x28, x29, x30, x31, x33, x34, x35, x36, x37, x39, x40, x42, x43, x44, x46, x47, x49, x50, x51), as shown in figure 3.

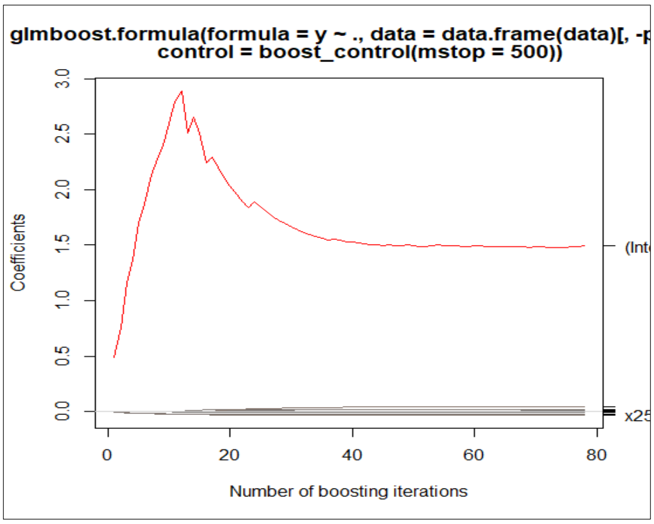

Figure 3. coefficients predicter with number iterations =500 using probing

The cross validation select (x8, x11, x19, x23, x25, x31, x28, x34, x35, x36, x37, x38, x42, x46, x47, x49), in table 5, and this explain in figures (4, 5). boosting error may increase with the number of iterations, when the data is noisy, as in the figure 4.

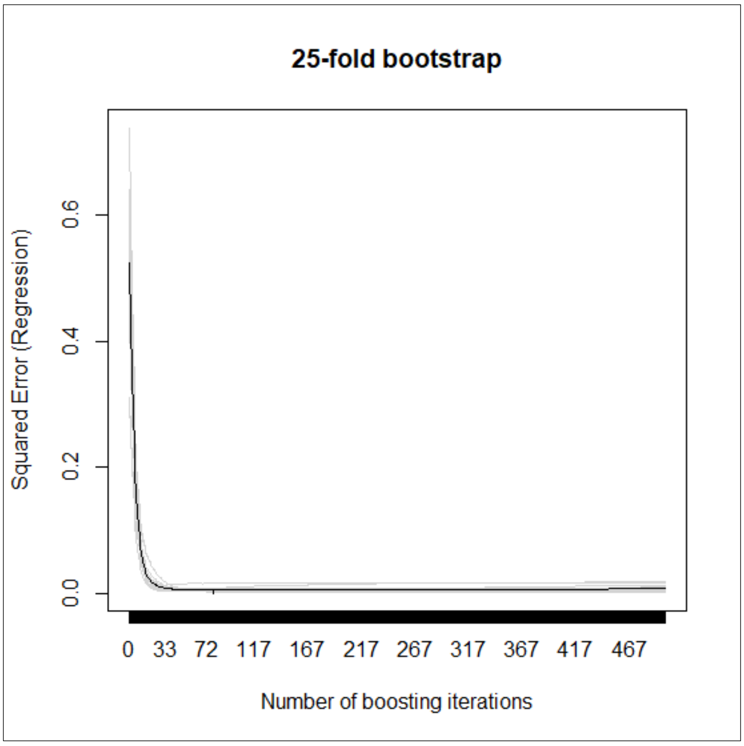

Figure .4. 25-fold cross validation & MSE

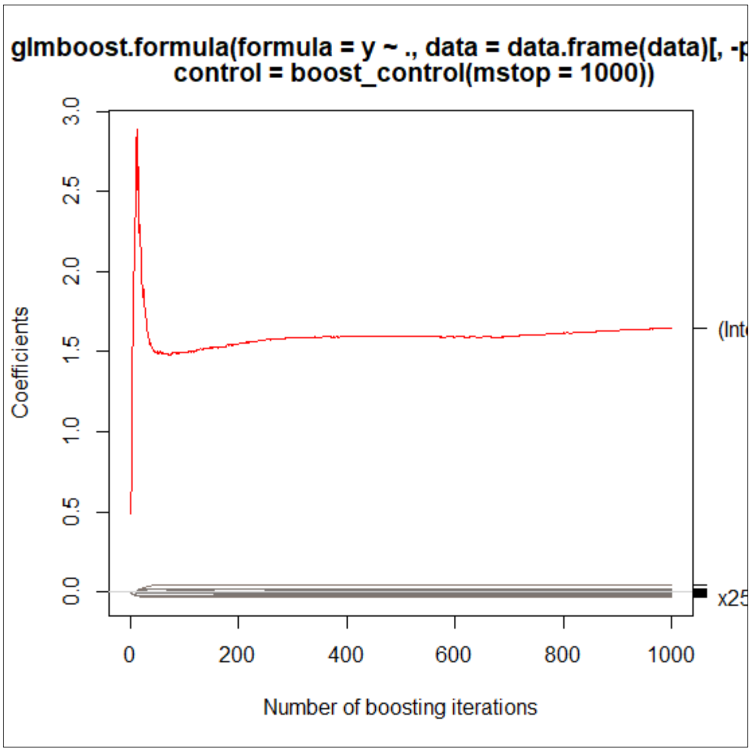

Figure 3. Coefficients predicter with number iterations =500 using C.V

It was clear as a consequence, the set of variables considered to be informative further shrinks in all three scenarios. boosting with probing leads to the largest set of selected variables in all methods.

Conclusion

Three methods (lasso, probing, and cross-validation) for determining the best number of iterations for sparse and rapid variable. selection with model-based boosting were proposed. It was demonstrated that the methods are both prober and. useful strategies for variable selection through simulation and a real data analysis example. Unlike most model-based. boosting tuning processes, which rely on resampling to improve prediction accuracy.

Relying on mean. Square error, the results show that the probing approach had the least MSE compared with other methods Consequently, it was concluded, through simulation and real data analysis example. That the probing method is the best approach because it can choose the largest number of selected. variables, which makes it is performance better for model estimation.

References

1. Bühlmann, P., Kalisch, M., & Meier, L. (2014). High-dimensional statistics with a view toward applications in biology. Annual Review of Statistics and Its Application, 1, 255-278.
2. Clarke, R., Ressom, H. W., Wang, A., Xuan, J., Liu, M. C., Gehan, E. A., & Wang, Y. (2008). The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. Nature reviews cancer, 8(1), 37-49.
3. Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York: Springer series in statistics.
4. Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York: Springer series in statistics.
5. Hepp, T., Schmid, M., Gefeller, O., Waldmann, E., & Mayr, A. (2016). Approaches to regularized regression–a comparison between gradient boosting and the lasso. Methods of information in medicine, 55(05), 422-430.
6. Leng, C., Lin, Y., & Wahba, G. (2006). A note on the lasso and related procedures in model selection. Statistica Sinica, 1273-1284.
7. Mayr, A., Binder, H., Gefeller, O., & Schmid, M. (2014). The evolution of boosting algorithms. Methods of information in medicine, 53(06), 419-427.-75
8. Ridgeway, G. (1999). The state of boosting. Computing science and statistics, 172-181.
9. Romero, R., Espinoza, J., Gotsch, F., Kusanovic, J. P., Friel, L. A., Erez, O., ... & Tromp, G. (2006). The use of high-dimensional biology (genomics, transcriptomics, proteomics, and metabolomics) to understand the preterm parturition syndrome. BJOG: An International Journal of Obstetrics & Gynaecology, 113, 118-135.
10. Sohrabi, N., & Movaghari, H. (2020). Reliable factors of Capital structure: Stability selection approach. The Quarterly Review of Economics and Finance, 77, 296-310.
11. Tarr, G., Müller, S., & Welsh, A. (2015). mplot: An R package for graphical model stability and variable selection procedures. arXiv preprint arXiv:1509.07583.
12. Thomas, J., Hepp, T., Mayr, A., & Bischl, B. (2017). Probing for sparse and fast variable selection with model-based boosting. Computational and mathematical methods in medicine, 2017.
13. Thomas, J., Mayr, A., Bischl, B., Schmid, M., Smith, A., & Hofner, B. (2018). Gradient boosting for distributional regression: faster tuning and improved variable selection via noncyclical updates. Statistics and Computing, 28(3), 673-687.
14. Xingyu, T. (2016). BOOSTING FOR PARTIALLY LINEAR ADDITIVE MODELS.