P- ISSN 1991-8941 E-ISSN 2706-6703 Journal of University of Anbar for Pure Science (JUAPS) Open Access
2013,(7), (1) :178-186

Application of Boosting Technique to Ridge Regression

Feras Sh. Mahmood

University Of Anbar-College of Education for Pure Science.

ARTICLE INFO

Received: 5 / 4 /2012
Accepted: 15 / 10 /2012
Available online: 19/7/2022
DOI:

Keywords:
Boosting technique,
Ridge Regression Estimator,
Jackknifed Ridge Estimator,
Least Squares Estimator ,
Mean Squared Error,
Multicollinearity.

ABSTRACT

The aim of this paper is to apply the boosting technique in different estimation methods in MLR model when multicollinearity present. We develop algorithms for these regression estimators. Boosting needs to be stopped after a suitable number of iterations to avoid over fitting. The computationally efficient AICC criterion is used. We illustrate the performance of these estimators using Hoerl and Kennard data, the prostate cancer (Tibshirani) data, the prostate cancer data with added multicollinearity, and the simulated data.

1. Introduction

Schapire [14] introduced boosting as a common method which attempts to boost the precision of any learning algorithm. Boosting is one of the most successful and useful methods introduced in the last eighteen years. The goal of boosting is to improve the performance of weak learning algorithms by combining them in a certain way. The first algorithm of this type was developed by Schapire [14]. The second algorithm, developed by Freund [8], it was a more efficient boosting algorithm. It was originally designed for classification problems, but it can be extended to regression. Friedman [9] developed boosting methods for regression which are implemented for optimization using the RSS function: this is what we call L2Boosting. Boosting technique can be useful to iteratively fit the current residuals. Duffy and Helmbold [6] gave a brief overview of boosting and contrasted the classification and regression settings. B¨uhlmann and Yu [4] proposed an algorithm called L2Boost in linear model. They constructed it from the L2 loss function. L2 loss is suitable for regression. Therefore, they proposed to use L2Boost for a regression problem and not necessarily with Multicollinearity. Boosting of an estimator means that the estimator is applied iteratively to the residuals of the previous iteration. Efron et al.[7] proposed LARS algorithm as a combination of forward stage-wise linear regression (FSLR), which seems closely related to L2Boost, and the L1penalized Lasso (see [16]). Some results for boosting include Jiang [13], and Zhang and Yu [18]. Except for Jiang [13] result, these authors consider versions of boosting either with L1 loss function constraints for the boosting coefficients or, as in Zhang and Yu [18], with a version of boosting which we found very difficult to use in practice. Zhang and Yu [18] generalized this result without too much effort to a setting with increasing dimension of the predictor variable. B¨uhlmann [3] extended L2Boost to the special case of high-dimensional linear models, where the number of covariates may exceed the sample size. Boosting has been originally developed in the machine learning community to improve classification procedures [14]. B¨uhl-mann [3] developed an algorithm for estimation and variable selection in high-dimensional linear models. Tutz and Binder [17] applied boosting to ridge regression. They introduced a partial boosting algorithm as a parsimonious model like Lasso. In this paper, we apply boosting technique in different estimation methods in MLR model when multicollinearity present. We compare these estimators under boosting

* Corresponding author at: University Of Anbar-College of Education for Pure Science.;
ORCID: https://orcid.org/0000-0001-5859-6212 .Mobil:777777
E-mail address: dean_coll.science@uoanbar.edu.iq

technique with their same estimators before boosting. We find the improvement in these methods by boosting. We introduce boosting of Ordinary Ridge Regression (ORR) [10], Ordinary Jackknife Ridge (OJR) [15], Modified Jackknife Ridge (MJR) [1] and (r, k) class [2] estimators by computation method. Here we propose a computationally approach for the different estimators in boosting, that is, all these estimator are used to fit iteratively the current residuals yielding an alternative to their usual estimators under multicollinearity. We develop algorithm for all these regression estimators. Boosting needs to be stopped at a suitable number of iterations, to avoid over fitting. The computationally efficient AICC criterion (see [3]) is used in this paper. We illustrate the performance of these estimators by the Hoerl and Kennard [11], the prostate cancer data [16], the prostate cancer data with multi-collinearity and the simulated data. Results of these computations are given in Tables (1 - 10).

2. Boosting of Various Ridge Type Estimators

We apply the boosting technique to some estimators mentioned in this paper.

2.1Boosted Ridge Regression

Our goal is to assign degrees of freedom for boosting. Denote by

$$D_j = \frac{X_j X_j'}{\|X_j\|^2} \quad (1)$$

the n × n hat-matrix for the linear least squares fitting operator using only the j-th predictor variable, $X_j = \left(X_j^1, X_j^2, \cdots, X_j^p\right)'$. Let $\|x\|^2 = x'x$ denote the Euclidean norm for a vector x ∈ R^n. The L2Boost hat-matrix, using the step size z, $0 < z < 1$, equals

$$\hat{\eta}_m = I - \left(I - zD_j^{w_m}\right)\left(I - zD_j^{w_{m-1}}\right)\cdots\left(I - zD_j^{w_1}\right), \quad (2)$$

where $w_i \in 1, \cdots, p$ denotes the comp-onent selected in the component-wise least squares at the i-th boosting iteration. Using the trace of η_m as degrees of freedom, we use a corrected AIC (AICC)(see [3], [12]) to define a stopping rule for boosting.

$$\text{AICC}(m) = \log\left(\hat{\sigma}^2\right) + \frac{1 + \text{tr}(\eta_m)/n}{1 - \left(\text{tr}(\eta_m) + 2\right)/n}, \quad (3)$$

$$\hat{\sigma}^2 = \frac{1}{2}\sum_{i=1}^{n}\left(Y_i - \left(\hat{\eta}_m Y\right)_i\right)^2, Y = \left(Y_1, Y_2, \cdots, Y_n\right)'. \quad (4)$$

An estimate for the number of boosting iterations is then

$$\hat{M} = \underset{1 \le m \le m_{max}}{\arg\min}\, AICC(m), \quad (5)$$

where m_{max} is a large upper bound for the candidate number of boosting iterations. For the MLR model, the algorithm for boosted ORR is given below.

2.1.1 Algorithm 1: Boosted ORR

Step 1. (initialization).

Given data (X_i, Y_i); $i = 1, 2, \cdots, n$, fit ORR model yielding ORR estimates:

$$\hat{B}_0(k) = \left[X'X + kI_p\right]^{-1} X'Y, \quad \hat{Y} = X\hat{B}_0(k), \quad (6)$$

where $\hat{B}(k)$ is ORR estimated from the original data. Set m = 0

Step 2. Compute residuals $\epsilon_i = Y_i - \hat{Y}_{m-1}$, $i = 1, 2, \cdots, n$ and fit ORR to the current residuals yielding the following solution.

$$\hat{B}_m(k) = \left[X'X + kI_p\right]^{-1} X'\left(Y_i - \hat{Y}_{m-1}\right). \quad (7)$$

The fit is denoted by $\hat{Y}_m^{ORR} = X\hat{B}_m(k)$, which is an estimate based on the original predictor variables and the current residuals. The new estimator is given by

$$\hat{Y}_m = \hat{Y}_{m-1} + \hat{Y}_m^{ORR}. \quad (8)$$

Step 3 (iteration). Increase the iteration index m by one and repeat 2 until a stopping iteration $\hat{M}$ from (5) is achieved.

2.2 Boosted Jackknifed Ridge Regression

We have described OJR introduced by Singh, et al. [15]. We develop an algorithm for boosting OJR and compare the result with OJR before boosting. Similarly, for the MLR model, the algorithm for boosted OJR is described below.

2.2.1 Algorithm 2: Boosted OJR

Step 1. (initialization).

Given data (X_i, Y_i); $i = 1, 2, \cdots, n$, apply OJR yielding OJR estimates:

$$\hat{B}_{(0)k}(J) = \left(I - \left((X'X + kI_p)^{-1} k \right)^2 \right) \times [X'X]^{-1} X'Y,\ \hat{Y}_{(0)J} = X\hat{B}_{(0)k}(J),$$

where $\hat{B}_{(0)k}(J)$ is OJR estimated from the original data. Set m = 0.

Step 2.

Compute residuals $\in_{iJ} = Y_i - \hat{Y}_{(m-1)J}$, $i = 1, 2, \cdots, n$ and fit OJR to the current residuals yielding the following solution.

$$\hat{B}_{(m)k}(J) = \left(I - \left([X'X + kI_p]^{-1} k \right)^2 \right) \times [X'X]^{-1} X' \left(Y_i - \hat{Y}_{(m-1)J} \right). \qquad (9)$$

The fit is denoted by $\hat{Y}^{OJR}_{(m)J} = X\hat{B}_{(m-1)k}(J)$, which is an estimate based on the original predictor variables and the current residuals. The new estimator is obtained as

$$\hat{Y}_{(m)J} = \hat{Y}_{(m-1)J} + \hat{Y}^{OJR}_{(m)J}. \qquad (10)$$

Step 3 (iteration). Increase the iteration index m by one and repeat step 2 until the stopping rule is satisfied after $\hat{M}$ iterations.

2.3 Boosted Modified Jackknife Ridge Regression

We have described MOJR estimator introduced by Batah, et al. [1] to deal with multicollinearity. We now develop an algorithm for boosting MOJR and compare its results with MOJR before boosting. We describe the algorithm for boosted MOJR below.

2.3.1Algorithm 3: Boosted MOJR

Step 1. (initialization).

Given data (X_i, Y_i); $i = 1, 2, \cdots, n$, apply MOJR yielding MOJR estimates.

$$\hat{B}_{(0)MJ}(k) = \left(I - \left((X'X + kI_p)^{-1} k \right)^2 \right) \left(I - \left((X'X + kI_p)^{-1} k \right) \right)$$ where $$\times [X'X]^{-1} X'Y,\ \hat{Y}_{(0)MJ} = X\hat{B}_{(0)MJ}(k),$$

$\hat{B}_{(0)MJ}(k)$ is OJR estimated from the original data. Set m = 0.

Step 2.

Compute residuals $\in_{iMJ} = Y_i - \hat{Y}_{(m-1)MJ}$, $i = 1, 2, \cdots, n$ and fit MJR to the current residuals yielding the following solution.

$$\hat{B}_{(m)MJ}(k) = \left(I - \left([X'X + kI_p]^{-1} k \right)^2 \right) \left[I - \left([X'X + kI_p]^{-1} k \right) \right] \times [X'X]^{-1} X' \left(Y_i - \hat{Y}_{(m-1)MJ} \right). \qquad (11)$$

The fit is denoted by $\hat{Y}^{MJR}_{(m)MJ} = X\hat{B}_{(m-1)MJ}(k)$, which is an estimate based on the original predictor variables and the current residuals. The new estimator is obtained as

$$\hat{Y}_{(m)MJ} = \hat{Y}_{(m-1)MJ} + \hat{Y}^{MJR}_{(m)MJ}. \qquad (12)$$

Step 3 (iteration). Increase the iteration index m by one and repeat step 2 until the stopping rule is satisfied after $\hat{M}$ iterations.

2.4 Boosted (r, k) Class Ridge Regression

We have described (r, k) class estimator introduced by Baye and Parker [2] to improve PCR by ridge regression to overcome multicollin-earity. We now develop an algorithm for boosted (r, k) class ridge regression and compare the results with (r, k) class before boosting. We describe the algorithm for boosted (r, k) class ridge regression as follows.

2.4.1 Boosted (r, k) Class Ridge Regression

Step 1. (initialization).

Given data $(X_i, Y_i);\ i = 1, 2, \cdots, n$, apply the (r, k) class model to obtain (r, k) class estimates.

$\hat{B}_{(0)r}(k) = T_r[T_r'X'XT_r + kI_r]^{-1} T_r'X'Y$, where $\hat{Y}_{(0)r} = X\hat{B}_{(0)r}(k)$,

$\hat{B}_{(0)r}(k)$ is OJR estimated from the original data. Set m = 0.

Step 2.

Compute residuals $\in_{ir} = Y_i - \hat{Y}_{(m-1)r}$, $i = 1, 2, \cdots, n$ and fit (r, k) class to the current residuals yielding the following solution.

$$\hat{B}_{(m)r}(k) = T_r[T_r'X'XT_r + kI_r]^{-1} T_r'X'(Y_i - \hat{Y}_{(m-1)r}). \quad (13)$$

The fit is denoted by $\hat{Y}^{class}_{(m)r} = X\hat{B}_{(m-1)r}(k)$, which is an estimate based on the original predictor variables and the current residuals. The new estimator is obtained as

$$\hat{Y}_{(m)r} = \hat{Y}_{(m-1)r} + \hat{Y}^{class}_{(m)r}. \quad (14)$$

Step 3 (iteration). Increase the iteration index m by one and repeat step 2 until the stopping rule is satisfied after $\hat{M}$ iterations.

1. 3 Illustrative Examples

We use the examples from Hoerl and Kennard [11], Tibshirani [16], and generate simulated data to illustrate the performance of boosting ORR, OJR, MOJR and (r, k) class. We also compare these results with ORR, OJR, MOJR and (r, k) class before boosting. Tables (1 - 10) show the SMSE estimates used for different methods mentioned above.

3.1 Result for HK Data

We find that the effect of boosting on OJR is better than on other estimators. All other estimators are comparable. The result of the example HK data (see Hoerl and Kennard [11]) is reported in Table (1).

3.2 Result for Prostate Data

The result for the prostate cancer data (see [16]) is reported in Table (2).

3.3 Result for Prostate Data with Multicollinearity

The result for the prostate cancer data (see [16]) with added multicollinearity is reported in Table (3).

3.4 Result of Simulated Data

An artificial simulated data is generated using R programming language (see Dalgaard, 2002). The boosted versions of the estimators mentioned above are computed for this data. In this data, there are eight explanatory variables. The true model is $Y = X\beta + \sigma \in$, and explanatory variables are generated using

$$x_{ij} = (1-\rho^2)^{\frac{1}{2}} w_{ij} + \rho w_{ip},$$
$$i = 1, 2, \cdots, n;\ j = 1, 2, \cdots, p,$$

where w_{ij} are independent standard normal deviates and ρ^2 is the correlation between x_{ij} and $x_{ij'}$ for $j, j' < p$ and $j \neq j'$, $j, j' = 1, 2, \cdots, p$. When j or $j' = p$,the correlation is ρ.

We consider We consider four different values for ρ= 0.9, 0.99, 0.999 and 0.9999. These variables are then standardized, so that $X'X$ and $X'Y$ are in the correlation form. We simulate the data with sample size $n = 97$.

The variances of the error terms are taken as σ^2= 0.0001, 0.01, 1, 25, and 100. The optimal k, given by penalty, is taken as Penalty = 0, 0.01, 0.1, 1, 10, 100, and 1000 (see Zou and Hastie, 2005). The results of these computations are reported in Tables (4 – 10).

Table 1: Table of SMSE for HK Data

Penalty	ORR	OJR	MOJR	(r,k) class
	Before Boosting			
0	1.4282	1.4282	1.4280	1.8070
0.01	1.4881	1.4381	1.5180	1.8110
0.1	1.9191	1.6432	2.0791	2.0333
1	6.1393	2.6420	7.5000	6.1552
10	51.0771	25.7171	73.9411	51.0791
100	96.2130	88.252	104.1580	96.2131
1000	103.994	103.078	104.91	103.994
	After Boosting			
0	1.4281	1.4280	1.4280	1.8072
0.01	1.4281	1.4280	1.4280	1.8072
0.1	1.4281	1.4280	1.4280	1.8072
1	1.4293	1.4280	1.4560	1.8072
10	1.6117	1.5296	2.2790	1.8078
100	2.4852	2.0735	75.8310	2.517
1000	19.419	5.7028	104.543	19.4224

Table 2: Table of SMSE for Prostate Data

Penalty	ORR	OJR	MOJR	(r,k) class
	Before Boosting			
0	0.4962	0.4962	0.4962	0.4962
0.01	0.4963	0.4962	0.4964	0.4963
0.1	0.5061	0.4969	0.5099	0.5061
1	0.6405	0.5489	0.7029	0.6405
10	1.1138	0.9212	1.2962	1.1138
100	1.3916	1.3486	1.4345	1.3916
1000	1.4325	1.4277	1.4372	1.4325
	After Boosting			
0	0.4962	0.4962	0.4962	0.4962
0.01	0.4962	0.4962	0.4962	0.4962
0.1	0.4962	0.4962	0.4962	0.4962
1	0.4962	0.4962	0.4962	0.4962
10	0.4962	0.4962	0.5153	0.4962
100	0.5341	0.5085	1.2747	0.5341
1000	0.8831	0.7060	1.4353	0.8831

Table 3: Table of SMSE for Prostate Data With Multicollinearity

Penalty	ORR	OJR	MOJR	(r,k) class
	Before Boosting			
0	0.5076	0.5076	0.5076	0.5076
0.01	0.5077	0.5076	0.5077	0.5077
0.1	0.5131	0.5078	0.5150	0.5131
1	0.6024	0.5409	0.6426	0.6024
10	1.0242	0.8111	1.2170	1.0242
100	1.3982	1.3323	1.4639	1.3982
1000	1.4626	1.4550	1.4702	1.4626
	After Boosting			
0	0.4952	0.4952	0.4952	0.5076
0.01	0.5076	0.5076	0.5076	0.5076
0.1	0.5076	0.5076	0.5076	0.5076
1	0.5076	0.5076	0.5076	0.5076
10	0.5076	0.5076	0.5195	0.5076
100	0.5315	0.5131	1.2271	0.5315
1000	0.7606	0.6316	1.4672	0.7606

Table 4: Table of SMSE for Simulated Data with Penalty = 0

σ	ρ	*ORR*	*OJR*	*MOJR*	(r,k) class	*ORR*	*OJR*	*MOJR*	(r,k) class
		Before Boosting				After Boosting			
0.01	0.9	0.0000136	0.0000136	0.0000136	0.0000136	0.0000136	0.0000136	0.0000136	0.0000136
	0.99	0.0000117	0.0000117	0.0000117	0.0000125	0.0000117	0.0000117	0.0000117	0.0000125
	0.999	0.0000115	0.0000115	0.0000115	0.0000123	0.0000115	0.0000115	0.0000115	0.0000123
	0.9999	0.0000115	0.0000115	0.0000115	0.0000123	0.0000115	0.0000115	0.0000115	0.0000123
0.1	0.9	0.0014744	0.0014744	0.0014744	0.0014744	0.0014744	0.0014744	0.0014744	0.0014744
	0.99	0.0012734	0.0012734	0.0012734	0.0014002	0.0012734	0.0012734	0.0012734	0.0014002
	0.999	0.0012545	0.0012545	0.0012545	0.0013799	0.0012545	0.0012545	0.0012545	0.0013799
	0.9999	0.0012523	0.0012523	0.0012523	0.0013776	0.0012523	0.0012523	0.0012523	0.0013776
1	0.9	0.009589	0.009589	0.009589	0.009589	0.009589	0.009589	0.009589	0.009589
	0.99	0.0094516	0.0094516	0.0094516	0.0103938	0.0094516	0.0094516	0.0094516	0.0103938
	0.999	0.0094293	0.0094293	0.0094293	0.0103693	0.0094293	0.0094293	0.0094293	0.0103693
	0.9999	0.0094245	0.0094245	0.0094245	0.010364	0.0094245	0.0094245	0.0094245	0.010364
5	0.9	0.0102746	0.0102746	0.0102746	0.0102746	0.0102746	0.0102746	0.0102746	0.0102746
	0.99	0.0102789	0.0102789	0.0102789	0.0112214	0.0102789	0.0102789	0.0102789	0.0112214
	0.999	0.0102794	0.0102794	0.0102794	0.0112219	0.0102794	0.0102794	0.0102794	0.0112219
	0.9999	0.0102794	0.0102794	0.0102794	0.011222	0.0102794	0.0102794	0.0102794	0.011222
10	0.9	0.0108822	0.0108822	0.0108822	0.0108822	0.0108822	0.0108822	0.0108822	0.0108822
	0.99	0.0108849	0.0108849	0.0108849	0.0112332	0.0108849	0.0108849	0.0108849	0.0112332
	0.999	0.0108848	0.0108848	0.0108848	0.0112344	0.0108848	0.0108848	0.0108848	0.0112344
	0.9999	0.0108846	0.0108846	0.0108846	0.0112348	0.0108846	0.0108846	0.0108846	0.0112348

Table 5: Table of SMSE for Simulated Data with Penalty = 0.01

σ	ρ	*ORR*	*OJR*	*MOJR*	(r,k) class	*ORR*	*OJR*	*MOJR*	(r,k) class
		Before Boosting				After Boosting			
0.01	0.9	0.0000137	0.0000136	0.0000137	0.0000137	0.0000136	0.0000136	0.0000136	0.0000136
	0.99	0.0000118	0.0000117	0.0000119	0.0000125	0.0000117	0.0000117	0.0000117	0.0000125
	0.999	0.0000121	0.0000119	0.0000122	0.0000123	0.0000115	0.0000115	0.0000115	0.0000123
	0.9999	0.0000123	0.0000122	0.0000123	0.0000123	0.0000115	0.0000115	0.0000116	0.0000123
0.1	0.9	0.0014753	0.0014744	0.0014754	0.0014753	0.0014744	0.0014744	0.0014744	0.0014744
	0.99	0.0012981	0.0012785	0.0013117	0.0014002	0.0012734	0.0012734	0.0012734	0.0014002
	0.999	0.0013521	0.0013308	0.001373	0.0013799	0.0012545	0.0012545	0.0012545	0.0013799
	0.9999	0.0013743	0.0013710	0.0013775	0.0013776	0.0012533	0.0012523	0.0012835	0.0013776
1	0.9	0.0095922	0.0095891	0.0095926	0.0095922	0.009589	0.009589	0.009589	0.009589
	0.99	0.0095736	0.0094711	0.0096365	0.0103938	0.0094516	0.0094516	0.0094516	0.0103938
	0.999	0.0100851	0.0098935	0.0102704	0.0103693	0.0094293	0.0094293	0.0094293	0.0103693
	0.9999	0.010327	0.0102916	0.0103624	0.010364	0.0094277	0.0094245	0.0095608	0.010364
5	0.9	0.0102771	0.0102747	0.0102774	0.0102771	0.0102746	0.0102746	0.0102746	0.0102746
	0.99	0.0103771	0.0102913	0.010425	0.0112214	0.0102789	0.0102789	0.0102789	0.0112214
	0.999	0.0109035	0.0106966	0.0111026	0.0112219	0.0102794	0.0102794	0.0102794	0.0112219
	0.9999	0.0111794	0.0111388	0.0112199	0.011222	0.0102807	0.0102794	0.0103712	0.011222
10	0.9	0.0108837	0.0108822	0.0108839	0.0108837	0.0108822	0.0108822	0.0108822	0.0108822
	0.99	0.0109412	0.0108954	0.0109715	0.0112332	0.0108849	0.0108849	0.0108849	0.0112332
	0.999	0.0111428	0.0110777	0.0112063	0.0112344	0.0108848	0.0108848	0.0108848	0.0112344
	0.9999	0.0112232	0.011212	0.0112343	0.0112348	0.0108866	0.0108846	0.0109546	0.0112348

Table 6: Table of SMSE for Simulated Data with Penalty = 0.1

σ	ρ	*ORR*	*OJR*	*MOJR*	(r,k) class	*ORR*	*OJR*	*MOJR*	(r,k) class
		Before Boosting				After Boosting			
0.01	0.9	0.0000161	0.0000136	0.0000163	0.0000161	0.0000136	0.0000136	0.0000136	0.0000136
	0.99	0.000014	0.0000121	0.0000142	0.0000142	0.0000117	0.0000117	0.0000117	0.0000125
	0.999	0.000014	0.0000122	0.0000141	0.000014	0.0000115	0.0000115	0.0000116	0.0000123
	0.9999	0.000014	0.0000123	0.000014	0.000014	0.0000119	0.0000117	0.0000121	0.0000123
0.1	0.9	0.0015054	0.0014804	0.0015215	0.0015054	0.0014744	0.0014744	0.0014744	0.0014744
	0.99	0.0013738	0.0013509	0.001395	0.0014017	0.0012734	0.0012734	0.0012734	0.0014002
	0.999	0.001378	0.0013733	0.0013813	0.0013814	0.0012555	0.0012545	0.0012863	0.0013799
	0.9999	0.0013788	0.0013769	0.0013791	0.0013791	0.0013259	0.0012965	0.0013618	0.0013776
1	0.9	0.0097157	0.0096093	0.0097812	0.0097157	0.009589	0.009589	0.009589	0.009589
	0.99	0.0101093	0.0099168	0.0102954	0.0103939	0.0094516	0.0094516	0.0094516	0.0103938
	0.999	0.0103324	0.0102969	0.0103678	0.0103694	0.0094325	0.0094293	0.0095674	0.0103693
	0.9999	0.0103603	0.0103564	0.0103641	0.0103642	0.009857	0.0096404	0.0101931	0.010364
5	0.9	0.0103833	0.0102892	0.0104376	0.0103833	0.0102746	0.0102746	0.0102746	0.0102746
	0.99	0.0109083	0.010703	0.0111062	0.0112214	0.0102789	0.0102789	0.0102789	0.0112214
	0.999	0.0111796	0.0111393	0.0112199	0.0112219	0.0102808	0.0102795	0.0103745	0.0112219
	0.9999	0.0112176	0.0112132	0.011222	0.011222	0.0106571	0.0104403	0.0110266	0.011222
10	0.9	0.0109387	0.0108927	0.0109691	0.0109387	0.0108822	0.0108822	0.0108822	0.0108822
	0.99	0.0111422	0.0110772	0.0112055	0.0112332	0.0108849	0.0108849	0.0108849	0.0112332
	0.999	0.0112229	0.0112118	0.011234	0.0112344	0.0108867	0.0108848	0.0109556	0.0112344
	0.9999	0.0112336	0.0112324	0.0112348	0.0112348	0.0110681	0.0109868	0.0111808	0.0112348

Table 7: Table of SMSE for Simulated Data with Penalty = 1

σ	ρ	*ORR*	*OJR*	*MOJR*	(r,k) class	*ORR*	*OJR*	*MOJR*	(r,k) class
		Before Boosting				After Boosting			
0.01	0.9	0.000201	0.0000172	0.0002455	0.000201	0.0000136	0.0000136	0.0000136	0.0000136
	0.99	0.0001551	0.0000142	0.0001851	0.0001551	0.0000117	0.0000117	0.0000119	0.0000125
	0.999	0.0001513	0.000014	0.0001801	0.0001513	0.0000119	0.0000117	0.0000122	0.0000123
	0.9999	0.0001509	0.000014	0.0001796	0.0001509	0.0000122	0.0000122	0.0000123	0.0000123
0.1	0.9	0.0017486	0.0015671	0.0018103	0.0017486	0.0014744	0.0014744	0.0014744	0.0014744
	0.99	0.0015219	0.0013952	0.0015514	0.0015252	0.0012744	0.0012734	0.0013101	0.0014002
	0.999	0.0015015	0.0013807	0.0015272	0.0015019	0.0013282	0.0012988	0.0013656	0.0013799
	0.9999	0.0014993	0.001379	0.0015246	0.0014993	0.0013709	0.0013647	0.0013761	0.0013776
1	0.9	0.010275	0.0100702	0.0104655	0.010275	0.009589	0.009589	0.009589	0.009589
	0.99	0.0103675	0.0103215	0.0104051	0.0104045	0.0094548	0.0094517	0.0096116	0.0103938
	0.999	0.0103762	0.0103618	0.0103822	0.01038	0.0098617	0.0096448	0.0102136	0.0103693
	0.9999	0.0103744	0.0103634	0.010377	0.0103748	0.0102905	0.0102234	0.0103471	0.010364
5	0.9	0.0109242	0.0107246	0.0111173	0.0109242	0.0102746	0.0102746	0.0102746	0.0102746
	0.99	0.0111802	0.0111406	0.0112197	0.0112216	0.0102804	0.010279	0.0103986	0.0112214
	0.999	0.0112177	0.0112132	0.0112221	0.0112221	0.0106594	0.0104421	0.0110449	0.0112219
	0.9999	0.0112217	0.0112211	0.0112222	0.0112222	0.0111374	0.0110609	0.0112026	0.011222
10	0.9	0.0111385	0.0110746	0.0112009	0.0111385	0.0108822	0.0108822	0.0108822	0.0108822
	0.99	0.0112218	0.0112107	0.0112328	0.0112332	0.0108869	0.010885	0.0109651	0.0112332
	0.999	0.0112333	0.0112321	0.0112344	0.0112345	0.0110681	0.0109868	0.0111855	0.0112344
	0.9999	0.0112347	0.0112345	0.0112348	0.0112348	0.0112117	0.0111905	0.0112295	0.0112348

Table 8: Table of SMSE for Simulated Data with Penalty = 10

σ	ρ	*ORR*	*OJR*	*MOJR*	(r,k) class	*ORR*	*OJR*	*MOJR*	(r,k) class
		Before Boosting				After Boosting			
0.01	0.9	0.0040129	0.0014392	0.0061685	0.0040129	0.0000136	0.0000136	0.0000142	0.0000136
	0.99	0.003527	0.001113	0.0054728	0.003527	0.0000121	0.0000119	0.0000124	0.0000125
	0.999	0.0034814	0.0010846	0.0054057	0.0034814	0.0000122	0.0000122	0.0000123	0.0000123
	0.9999	0.0034769	0.0010818	0.005399	0.0034769	0.0000123	0.0000123	0.0000123	0.0000123
0.1	0.9	0.0050424	0.0028333	0.0068932	0.0050424	0.0014757	0.0014745	0.0015524	0.0014757
	0.99	0.0044798	0.002364	0.0061853	0.0044802	0.0013482	0.0013185	0.0013929	0.0014002
	0.999	0.0044262	0.0023214	0.0061161	0.0044263	0.0013732	0.001367	0.0013791	0.0013799
	0.9999	0.0044207	0.002317	0.006109	0.0044207	0.0013769	0.0013762	0.0013775	0.0013776
1	0.9	0.0107551	0.0105619	0.0109224	0.0107551	0.0095924	0.0095891	0.0099591	0.0095924
	0.99	0.0106537	0.0104688	0.0108035	0.0106575	0.009885	0.0096666	0.0103117	0.0103938
	0.999	0.0106368	0.0104513	0.0107858	0.0106372	0.0102957	0.0102286	0.0103608	0.0103693
	0.9999	0.0106331	0.010447	0.0107825	0.0106332	0.0103564	0.0103488	0.0103632	0.010364
5	0.9	0.0111862	0.0111457	0.0112261	0.0111862	0.0102763	0.0102747	0.0106127	0.0102763
	0.99	0.0112217	0.0112143	0.0112285	0.011226	0.0106666	0.0104476	0.0111296	0.0112214
	0.999	0.0112258	0.0112224	0.0112287	0.0112263	0.0111379	0.0110618	0.0112122	0.0112219
	0.9999	0.0112263	0.0112232	0.0112287	0.0112263	0.0112132	0.0112044	0.011221	0.011222
10	0.9	0.011219	0.0112063	0.0112314	0.011219	0.0108841	0.0108822	0.0110387	0.0108841
	0.99	0.0112329	0.0112311	0.0112345	0.0112341	0.0110677	0.0109866	0.0112077	0.0112332
	0.999	0.0112348	0.0112343	0.0112352	0.0112349	0.0112114	0.0111902	0.0112318	0.0112344
	0.9999	0.0112351	0.0112348	0.0112353	0.0112351	0.0112324	0.01123	0.0112345	0.0112348

Table 9: Table of SMSE for Simulated Data with Penalty = 100

σ	ρ	*ORR*	*OJR*	*MOJR*	(r,k) class	*ORR*	*OJR*	*MOJR*	(r,k) class
		Before Boosting				After Boosting			
0.01	0.9	0.0098613	0.008655	0.0110627	0.0098613	0.0000141	0.0000138	0.0069448	0.0000141
	0.99	0.009658	0.0083018	0.0110069	0.009658	0.0000124	0.0000124	0.0065312	0.0000125
	0.999	0.0096371	0.008266	0.0110007	0.0096371	0.0000123	0.0000123	0.0064923	0.0000123
	0.9999	0.009635	0.0082624	0.0110001	0.009635	0.0000123	0.0000123	0.0064884	0.0000123
0.1	0.9	0.0100575	0.0090234	0.0110875	0.0100575	0.0015616	0.0015274	0.0075573	0.0015616
	0.99	0.009853	0.0086645	0.0110352	0.0098531	0.0013935	0.0013873	0.0071128	0.0014002
	0.999	0.0098319	0.0086279	0.0110294	0.0098319	0.0013792	0.0013785	0.0070703	0.0013799
	0.9999	0.0098297	0.0086241	0.0110288	0.0098297	0.0013775	0.0013775	0.0070659	0.0013776
1	0.9	0.0111476	0.0110696	0.0112253	0.0111476	0.0100387	0.0098136	0.0109578	0.0100387
	0.99	0.0111172	0.011015	0.0112188	0.0111175	0.0103203	0.0102531	0.0108815	0.0103938
	0.999	0.0111125	0.0110066	0.0112178	0.0111125	0.0103616	0.010354	0.0108695	0.0103693
	0.9999	0.0111116	0.0110049	0.0112176	0.0111116	0.0103633	0.0103625	0.0108671	0.010364
5	0.9	0.0112298	0.011224	0.0112357	0.0112298	0.0106913	0.0104664	0.0112143	0.0106913
	0.99	0.0112335	0.0112313	0.0112357	0.0112339	0.0111393	0.0110647	0.0112283	0.0112214
	0.999	0.0112339	0.0112322	0.0112357	0.011234	0.0112131	0.0112044	0.0112299	0.0112219
	0.9999	0.011234	0.0112323	0.0112357	0.011234	0.0112211	0.0112202	0.01123	0.011222
10	0.9	0.0112337	0.0112316	0.0112358	0.0112337	0.0110654	0.0109844	0.0112282	0.0110654
	0.99	0.0112355	0.011235	0.0112359	0.0112356	0.0112104	0.0111893	0.0112344	0.0112332
	0.999	0.0112357	0.0112355	0.0112359	0.0112357	0.011232	0.0112297	0.0112353	0.0112344
	0.9999	0.0112358	0.0112356	0.0112359	0.0112358	0.0112345	0.0112343	0.0112354	0.0112348

Table 10: Table of SMSE for Simulated Data with Penalty = 1000

σ	ρ	*ORR*	*OJR*	*MOJR*	(r,k) class	*ORR*	*OJR*	*MOJR*	(r,k) class
		Before Boosting				After Boosting			
0.01	0.9	0.0110859	0.0109379	0.011234	0.0110859	0.0007646	0.0000645	0.0111744	0.0007646
	0.99	0.0110614	0.0108895	0.0112332	0.0110614	0.0004927	0.000033	0.0111641	0.0004927
	0.999	0.0110588	0.0108844	0.0112332	0.0110588	0.0004707	0.000031	0.0111631	0.0004707
	0.9999	0.0110585	0.0108839	0.0112331	0.0110585	0.0004686	0.0000308	0.011163	0.0004686
0.1	0.9	0.0111074	0.0109805	0.0112342	0.0111074	0.0022551	0.0016483	0.0111832	0.0022551
	0.99	0.0110829	0.0109323	0.0112336	0.0110829	0.0018204	0.0014168	0.011173	0.0018211
	0.999	0.0110804	0.0109273	0.0112335	0.0110804	0.0017823	0.0013962	0.011172	0.0017824
	0.9999	0.0110801	0.0109267	0.0112335	0.0110801	0.0017784	0.0013939	0.0111718	0.0017784
1	0.9	0.0112263	0.0112169	0.0112358	0.0112263	0.0105194	0.0104103	0.011232	0.0105194
	0.99	0.0112228	0.0112099	0.0112358	0.0112229	0.0104222	0.0103801	0.0112306	0.0104298
	0.999	0.0112223	0.0112088	0.0112357	0.0112223	0.0104039	0.0103692	0.0112303	0.0104047
	0.9999	0.0112222	0.0112086	0.0112357	0.0112222	0.0103994	0.0103653	0.0112303	0.0103995
5	0.9	0.0112353	0.0112347	0.0112359	0.0112353	0.0111435	0.0110731	0.0112357	0.0111435
	0.99	0.0112357	0.0112354	0.011236	0.0112357	0.0112135	0.0112044	0.0112358	0.011222
	0.999	0.0112357	0.0112355	0.011236	0.0112357	0.0112216	0.0112202	0.0112359	0.0112225
	0.9999	0.0112357	0.0112355	0.011236	0.0112357	0.0112225	0.0112218	0.0112359	0.0112226
10	0.9	0.0112357	0.0112355	0.0112359	0.0112357	0.0112054	0.0111845	0.0112359	0.0112054
	0.99	0.0112359	0.0112359	0.011236	0.0112359	0.011231	0.0112285	0.0112359	0.0112333
	0.999	0.0112359	0.0112359	0.011236	0.0112359	0.0112343	0.011234	0.0112359	0.0112345
	0.9999	0.0112359	0.0112359	0.011236	0.0112359	0.0112348	0.0112347	0.011236	0.0112348

Conclusions

Boosting of ORR, OJR, MJR and (r,k) class estimators are used to estimate regression coefficients iteratively using residuals, yielding an alternative to usual estimators under the problem of multicollinearity. We show that these are useful and work better than the estimators before boosting by using the SMSE criterion.

References

[1] F. Batah, T. Ramanathan and S. Gore. (2008). The Efficiency of Modified Jackknife and Ridge Type Regression Estimators: A Comparison. *Surveys in Mathematics and its Applications* 3: 111-122.

[2] M. Baye and D. Parker. (1984). Combining ridge and principal component regression: a money demand illustration, *Communication in Statistics - Theory and Methods* 13: 197 - 205.

[3] P. Buhlmann. (2006). Boosting for high-dimensional data, *The Annals of Statistics* 34:559-583.

[4] P. Buhlmann and B. Yu. (2003). Boosting with the L2 loss: Regression and classification, *Journal of the American Statistical Association* 98:324 – 339.

[5] P, Dalgaard, Introductory Statistics with R. New York, Berlin, Heidelberg: Springer-Verlag, 2002.

[6] N. Duffy and D. Helmbold. (2002). Boosting Methods for Regression, *Machine Learning* 47:153 200.

[7] B. Efron, T. Hastie, I. Johnstone and R. Tibshirani. (2004). Least Angle Regression. *Annals of Statistics (with discussion)* 32:407 - 499.

[8] Y. Freund. (1995). Boosting a Weak Learning Algorithm by Majority, *Information and Computation* 121:256 - 285.

[9] J. Friedman, T. Hastie, S. Rosset, R. Tibshirani and J. Zhu. (2004). Discussion of Boosting Papers, *Annals of Statistics* 32:102 - 107.

[19]

[10] A. E. Hoerl, and R. W. Kennard. (1970). Ridge Regression: Biased Estimation for Non orthogonal Problems, *Technometrics* 12:55-67.

[11] A. Hoerl, and R. Kennard. (1981). Ridge regression: 1980 advances. algorithms, and applications. *American Journal of Mathematical and Management Sciences* 1:5 - 83.

[12] W. Jiang. (2004). Process consistency for Ada-boost, *Annals of Statistics* 32:13 29.

[13] R. Schapire. (1990).The Strenght of Weak Learnability, *Machine Learning* 5 :197 - 227.

[14] B. Singh, Y.Chaubey, and T. Dwivedi. (1986). An Almost Unbiased Ridge Estimator, *Sankhya Ser.B* 48: 342 - 346.

[15] R. Tibshirani. (1996). Regression Shrinkage and Selection via the Lasso, *Journal of the Royal Statistical Society B* 58:267 - 288.

[16] G. Tutz and H. Binder. (2007). Boosting Ridge regression, *Computational Statistics and Data Analysis* 51:6044 - 6059.

[17] T. Zhang and B. Yu. (2005). Boosting with early stopping: convergence and consistency, *The Annals of Statistics* 33:1538 1579.

[18] H. Zou and T. Hastie. (2005). Regularization and Variable Selection Via the Elastic Net, *Journal of the Royal Statistical Society B* 67:301 - 320.

P- ISSN 1991-8941 E-ISSN 2706-6703 *Journal of University of Anbar for Pure Science (JUAPS)* Open Access
2013,(7), (1) :178-186

تطبيق أسلوب الدعم لانحدار الحرف

فراس شاكر محمود الغريري
E.mail: ferashaker2001@yahoo.com

الخلاصة:

ان الهدف من هذا البحث هو تطبيق طريقة او اسلوب الدعم لمختلف طرق التقدير في نموذج الانحدار الخطي المتعدد عند وجود مشكلة التداخل الخطي في النموذج. وقد اقترحنا وطورنا خوارزمية لهذه الطرق مع توضيح لاداء كل طريقة وقارنا بينهم باستخدام اسلوب المحاكاة وكذلك بعض البيانات العددية مثل بيانات (Hoerl and Kennard) و بيانات سرطان البروستات (Tibshirani). أسلوب الدعم يحتاج الى التوقف لذلك استخدمنا معيار AICC . بعد المقارنة بين هذه الطرق باستخدام معيار مجموع مربعات الخطأ (SMSS) ، لوحظ ان طرق التقدير قيد الدراسة مع أسلوب الدعم لها أفضلية واستخداما في العمل من نفس الطرق بدون أسلوب الدعم.