DEPARTMENT OF MATHEMATICS AND STATISTICS , IIT KANPUR

Penalized Regression with focus on Lasso .

Submitted by :
Arvind(191026)
Rajat(191104)
Deepak(191037)
Shivraj(191136)
Vinay(191171)

September 15, 2020

Abstract

Penalized regression is well known concept and is widely used for high dimensional data (where the number feature's is possibly much larger than the number of observation). Our discussion here mainly move around the of concept of penalized regression with main focus on lasso. So, to have a higher prediction accuracy in case of high dimensional data .here we comparing the considerable performance between various method and tried to highlight the importance of penalized regression .The performance of penalized regression relies crucially on the choice of tuning parameter , which determines amount of regularization and hence the sparsity level of fitted model. The optimal choice of tuning parameter depends on both the structure of the design matrix and the unknown random error distribution (variance, tail behavior, etc.). This article reviews the current literature of tuning parameter selection for high-dimensional regression from both the theoretical and practical perspectives.We discuss various strategies that choose the tuning parameter to achieve prediction accuracy or support recovery. Our review complement existing theory and provide a resource to compare method across range of scenario .

Key words: Penalized regression , Lasso , High dimensional data , sparsity , tuning parameter

CONTENTS

1 History of Penalized Regression(LASSO)

LASSO (Least Absolute Shrinkage and Selection Operator) was introduced to improve the prediction accuracy and interpretability of regression models by altering the model fitting process to select only a subset of the provided regressors for use in the final model rather than using all of them. It was originally introduced in geophysics literature in 1986,and later independently rediscovered and popularized in 1996 by Robert Tibshirani, based on prior work that used the L_1 penalty for both fitting and penalization of the coefficients.

Before LASSO, the most widely used method for choosing which regressors to include in model was stepwise selection, which only improves prediction accuracy in certain cases, such as when only a few regressors have a strong relationship with the response.However in other cases, it can make prediction error worse. Also at the time, ridge regression was the most popular technique for improving prediction accuracy. Ridge regression improves prediction error by shrinking large regression coefficients in order to reduce over-fitting, but it does not perform regressor selection and therefore does not help to make the model more interpretable.

But after discovering, LASSO is able to achieve both of these goals by forcing the sum of the absolute value of the regression coefficients to be less than a fixed value (L_1 penalty), which forces certain coefficients to be set to zero, effectively choosing a simpler model that does not include those coefficients. This idea is similar to ridge regression, in which the sum of the squares of the coefficients is forced to be less than a fixed value(L_2 penalty) , but in the case of ridge regression, this only shrinks the size of the coefficients, it does not set any of them to zero.

2 Introduction

2.1 The Gauss model

Consider the Linear regression setup:
consider the iid samples $(x_i, y_i)\ \epsilon\ R^p XR$, $i = 1(1)n$

$$\mathbf{y}_i = \mathbf{x}_i^{'}\beta + \epsilon_i, i = 1(1)n$$

where $\beta\epsilon R^p$ is the unknown p dimensional unkown parameter vector ,x_i is p dimensional vector of regressors, y_i are response variable and ϵ_i i.i.d. random errors with following assumptions,
1) they have mean zero i.e E(ϵ_i)=0.
2) they are homoscedastic i.e E(ϵ_i^2)=σ^2.
3) errors are uncorrelared and are from gaussian.

The ordinary least square estimate's of β is obtained from equation :

$\min\limits_{\beta\epsilon R^p} \| y - X\beta\|_2^2$.........................(*)

2.2 The failure of least squares in high dimensions

following are the failures of least square solutions in high dimension :

(a.) When the rank of design matrix X is less than p .(eg in high dimensional data when p>n this situation occur) . then we have many solution for β . This type of non uniqueness makes interpretation of solution meaningless.eg in one solution we have $\hat{\beta}_j < 0$ and in other solution for same j we have $\hat{\beta}_j > 0$ for any j=1(1)p.

(b.)It will not generally be the case that for two solution of β we have same prediction .

(c.)So, Both interpretation and actual prediction is impossible when p>n.

(d.)Even when p is moderately close to n . we have unique least square solution it is not advisable to use this solution for prediction as in sample risk will be poor as in sample risk is equal to $\sigma^2 * p/n$

2.3 How can we do better? shrinkage or regularization

A better alternative is the penalized regression allowing to create a linear regression model that is penalized, for having too many variables in the model, by adding a constraint in the equation (*) above . This is also known as shrinkage or regularization methods.

The consequence of imposing this penalty, is to reduce (or shrink) the coefficient values towards zero. This allows the less contributing variables to have a coefficient close to zero or equal zero.

also we Note that, the shrinkage requires the selection of a tuning parameter (λ) that determines the amount of shrinkage.

2.4 Three norms L_0, L_1, L_2

Here we study basic definition about the three norms L_0, L_1, L_2 which is related to different type of penalties(best subset selection,LASSO,Ridge).

For L_0 norm,hey no its actually not a norm (as it does not satisfy positive homogeneity property of norm) , mathematically L_0 is defined as

$$L_0 = ||\beta||_0 = \sum_{j=1}^{p} 1(|\beta_j| > 0)$$

this penalizes the number of non-zero coefficients in the model and the corresponding penalty is called best subset selection.

For L_1 norm, mathematically L_1 is defined as

$$L_1 = ||\beta||_1 = \sum_{j=1}^{p} |\beta_j|$$

this penalty shrinks coefficients towards zero, and can also set many coefficients to be exactly zero and this is associated with LASSO penalty.
For L_2 norm, mathematically L_2 is defined as

$$L_2 = ||\beta||_2 = \sqrt{\sum_{j=1}^{p} \beta_j^2}$$

The shrinkage of the coefficients is achieved by penalizing the regression model with a penalty norm L_2.

The amount of the penalty can be fine-tuned using a constant called lambda (λ). Selecting a optimal value for λ is neccessary.

Now ,There are two reasons why we are often not satisfied with the least squares estimates.

• **The first is prediction accuracy**: the least squares estimates often have low bias but large variance. Prediction accuracy can sometimes be improved by shrinking or setting some coefficients to zero. By doing so we sacrifice a little bit of bias to reduce the variance of the predicted values, and hence may improve the overall prediction accuracy.

• **The second reason is interpretation**: With a large number of predictors, we often would like to determine a smaller subset that exhibit the strongest effects. In order to get the "big picture," we are willing to sacrifice some of the small details.

2.5 What you have learn in next sections

We have discussed up-to now about various problems in high dimensional data while using the least square solution . How inefficient is least square solution is when you are dealing with the high dimensional data and how regularization will help in overcoming these inefficiency of least square solution. further, now we are going briefly into regularization and particularly explaining lasso in broad aspects and comparing with the different methods.

3 Various subset selection

In this section we describe a number of approaches to variable subset selection with linear regression. In later sections we discuss shrinkage and hybrid approaches for controlling variance, as well as other dimension-reduction strategies. These all fall under the general heading model selection

With subset selection we retain only a subset of the variables, and eliminate the rest from the model. Least squares regression is used to estimate the coefficients of the inputs that are retained. There are a number of different strategies for choosing the subset.

3.1 Best subset selection

Best subset regression finds for each $k\epsilon(0,1,2,...,p)$ the subset of size k,that gives smallest residual sum of squares. An efficient algorithm "the leaps and bounds" procedure (Furnival and Wilson, 1974) makes this feasible for p as large as 30 or 40.The lower boundary represents the models that are eligible for selection by the best-subsets approach. Note that the best subset of size 2, for example, need not include the variable that was in the best subset of size 1. The best-subset curve is necessarily decreasing, so cannot be used to select the subset size k. The question of how to choose k involves the trade-off between bias and variance, along with the more subjective desire for parsimony. There are a number of criteria that one may use; typically we choose the smallest model that minimizes an estimate of the expected prediction error. Many of the other approaches are also used ,in that they use the training data to produce a sequence of models varying in complexity and indexed by a single parameter. In the next section we use cross-validation to estimate prediction error and select k; the AIC(Akaike information criterion)is a popular alternative.

3.2 Forward- and Backward-Step-wise Selection

Rather than search through all possible subsets (which becomes infeasible for p much larger than 40), we can seek a good path through them. Forward step-wise selection starts with the intercept, and then sequentially adds into the model the predictor that most improves the fit. With many candidate predictors, this might seem like a lot of computation; however, clever updating algorithms can exploit the QR decomposition for the current fit to rapidly establish the next candidate. Like best-subset regression, forward step-wise produces a sequence of models indexed by k, the subset size, which must be determined.Forward-step-wise selection is a greedy algorithm, producing a nested sequence of models. In this sense it might seem sub-optimal compared to best-subset selection. However, there are several reasons why it might be preferred:

Computational:- for large p we cannot compute the best subset sequence, but we can always compute the forward step-wise sequence (even when $p \geq N$).

Statistical: a price is paid in variance for selecting the best subset of each size; forward step-wise is a more constrained search, and will have lower variance, but perhaps more bias.

Backward-step-wise selection starts with the full model, and sequentially deletes the predictor that has the least impact on the fit. The candidate for dropping is the variable with the smallest Z-score. Backward selection can only be used when $N > p$, while forward step-wise can always be used.

Some software packages implement hybrid step-wise-selection strategies that consider both forward and backward moves at each step, and select the "best" of the two. For example in the R package the step function uses the AIC criterion for weighing the choices, which takes proper account of the number of parameters fit; at each step an add or drop will be performed that minimizes the AIC score. Other more traditional packages base the selection on F-statistics, adding "significant" terms, and dropping "non-significant" terms. These are out of fashion, since they do not take proper account of the multiple testing issues. It is also tempting after a model search to print out a summary of the chosen model, however, the standard errors are not valid, since they do not account for the search process. The bootstrap can be useful in such settings. Finally, we note that often variables come in groups (such as the dummy variables that code a multi-level categorical predictor). Smart stepwise procedures (such as step in R) will add or drop whole groups at a time, taking proper account of their degrees-of-freedom.

3.3 Forward-Stage-wise Regression

Forward-stage-wise regression (FS) is even more constrained than forward step-wise regression. It starts like forward-step-wise regression, with an intercept and centered predictors with coefficients initially all 0. At each step the algorithm identifies the variable most correlated with the current residual. It then computes the simple linear regression coefficient of the residual on this chosen variable, and then adds it to the current coefficient for that variable. This is continued till none of the variables have correlation with the residuals—i.e. the least-squares fit when N > p. Unlike forward-step-wise regression, none of the other variables are adjusted when a term is added to the model. As a consequence, forward stage-wise can take many more than p steps to reach the least squares fit, and historically has been dismissed as being inefficient. It turns out that this "slow fitting" can pay dividends in high-dimensional problems.Also both forward stage-wise and a variant which is slowed down even further are quite competitive, especially in very high-dimensional problems.

4 Shrinkage or regularization methods

By retaining a subset of the predictors and discarding the rest, subset selection produces a model that is interpretable and has possibly lower prediction error than the full model. However, because it is a discrete process— variables are either retained or discarded—it often exhibits high variance, and so doesn't reduce the prediction error of the full model. Shrinkage methods are more continuous, and don't suffer as much from high variability.

A good penalty function should result in an estimator with three properties:-

Unbiasedness: The resulting estimator is nearly unbiased when the true unknown parameter is large to avoid unnecessary modeling bias.

Sparsity: The resulting estimator is a thresholding rule, which automatically sets small estimated coefficients to zero to reduce model complexity.

Continuity: The resulting estimator is continuous in data to avoid instability in model prediction.

4.1 RIDGE REGRESSION

Ridge regression is like ordinary linear regression, The ridge coefficients are defined by solving:

$$\arg\min_{\beta\epsilon R^p} \| y - X\beta\|_2^2 + \lambda\|\beta\|_2^2$$

Ridge regression shrinks the regression coefficients, so that variables, with minor contribution to the outcome, have their coefficients close to zero.

The amount of the penalty can be fine-tuned using a constant called lambda (λ). Selecting an optimal value for λ is neccessary.

Here $\lambda \geq 0$ is a tuning parameter, which controls the strength of the penalty term. Write $\widehat{\beta}^{ridge}$ ridge as the ridge solution. Note that:

(a.)When λ=0, the penalty term has no effect, and ridge regression will produce the classical least square coefficients.Mathematically, $\lambda = 0$, we get the linear regression estimate

(b.)As λ increases to infinite, the impact of the shrinkage penalty grows, and the ridge regression coefficients will get close zero.Mathematically, When $\lambda = \infty$, we get $\widehat{\beta}^{ridge} = 0$

(c.)For λ in between, we are balancing two ideas : fitting a linear model of y on x, and shrinking the coefficients

Also, the penalty term $||\beta||_2^2 = \sum_{j=1}^{p} \beta_j^2$ is unfair is the predictor variables are not on the same scale. Therefore, if we know that the variables are not measured in the same units,we typically scale the columns of x (to have sample variance 1), and then we perform ridge regression.

The standardization of a predictor x, can be achieved using the formula $x^{'} = x/sd(x)$, where sd(x) is the standard deviation of x. The consequence of this is that, all standardized predictors will have a standard deviation of one allowing the final fit to not depend on the scale on which the predictors are measured.

One important **Advantage** of the ridge regression, is that it still performs well, compared to the ordinary least square method, in a situation where you have a large multivariate data with the number of predictors (p) larger than the number of observations (n).

One **Disadvantage** of the ridge regression is that, it will include all the predictors in the final model, unlike the step wise regression methods, which will generally select models that involve a reduced set of variables.

Ridge regression shrinks the coefficients towards zero, but it will not set any of them exactly to zero. The lasso regression is an alternative that overcomes this drawback.

WHEN WE USE RIDGE PENALTY

Ridge regression is a way to create a parsimonious model (simple models with great explanatory predictive power,they explain data with minimum number of parameter or regressors) when the number of predictor variables in a set exceeds the number of observations, or when a data set suffer from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors.It is hoped that the net effect will be to give estimates that are more reliable.

4.2 Lasso regression

Lasso stands for Least Absolute Shrinkage and Selection Operator. The lasso is a shrinkage method like ridge, with subtle but important differences. The lasso estimate is defined by

$$\arg\min_{\beta \epsilon R^p} \|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

The difference between lasso and ridge regression is that in ridge uses a squared L_2 penalty while the former uses an L_1 penalty.Even though these problems look similar,their solutions behave very differently!

In the case of lasso regression, the penalty has the effect of forcing some of the coefficient estimates, with a minor contribution to the model, to be exactly equal to zero. This means that, lasso can be also seen as an alternative to the subset selection methods for performing variable selection in order to reduce the complexity of the model.

As in ridge regression, selecting a optimal value of λ for the lasso is neccessary .

Here $\lambda \geq 0$ is a tuning parameter, which controls the strength of the penalty term. Write $\widehat{\beta}^{LASSO}$ as the LASSO solution. Note that:

(a.)When λ=0, the penalty term has no effect, and ridge regression will produce the classical least square coefficients.Mathematically , $\lambda = 0$, we get the linear regression estimate

(b.)As λ increases to infinite, the impact of the shrinkage penalty grows, and the Lasso regression coefficients will get close zero.Mathematically, when $\lambda = \infty$, we get $\widehat{\beta}^{LASSO} = 0$

(c.)For λ in between these two extremes, we are balancing two ideas: fitting a linear model of y on x, and shrinking the coefficients. But the nature of the l_1 penalty is such that some coefficients are shrunken to zero exactly.

This is what makes the lasso substantially different from ridge regression: it is able to perform variable selection in the linear model. As λ increases, more coefficients are set to zero (less variables are selected), and among the nonzero coefficients, more shrinkage is employed.

One obvious **Advantage** of lasso regression over ridge regression, is that it produces simpler and more interpretable models that incorporate only a reduced set of the predictors. However, neither ridge regression nor the lasso will universally dominate the other.

Generally, lasso might perform better in a situation where some of the predictors have large coefficients, and the remaining predictors have very small coefficients.

WHEN WE USE LASSO PENALTY

The lasso procedure encourages simple, sparse models (models with fewer parameters). This particular type of regression is well-suited for models showing high levels of muticollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

4.3 **SCAD** PENALISED REGRESSION

Variable selection is a fundamental task for high-dimensional statistical modeling. Traditional approaches follow stepwise and subset selection procedures, which are computationally intensive, unstable, and difficult to draw sampling properties from (see, e.g., Breiman 1996). Alternative variable selection methods are sparse penalized approaches, including ridge regression, least absolute shrinkage and selection operator (LASSO; Tibshirani 1996), and the **SMOOTHLY CLIPPED ABSOLUTE DEVIATION** (SCAD) penalty (Fan and Li 2001).

Among these, the SCAD estimator has all of the desirable properties, including unbiasedness, sparsity, and continuity.

SCAD penalty, which is defined by

$$p_\lambda(|\beta|) = \begin{cases} \lambda|\beta| & ,0 \leq |\beta| < \lambda \\ \frac{(a^2-1)\lambda^2-(|\beta|-a\lambda)^2}{2(a-1)} & ,\lambda \leq |\beta| < a\lambda \quad \} \\ \frac{(a+1)\lambda^2}{2} & ,|\beta| \geq a\lambda \end{cases}$$

where a = 3.7 and $\lambda > 0$ is the tuning parameter. The SCAD penalty is continuous and differentiable on (-∞, 0)∪(0,∞), but not differentiable at 0. Its derivative vanishes outside $[-a\lambda, a\lambda]$. Hence, the SCAD penalty can produce continuity, sparsity and unbiasedness estimator for the large coefficients.In this section we denote λ by λ_n to emphasize the dependency of λ on n. The penalized estimator can be obtained by minimizing

$$\|y - X\beta\|_2^2 + \sum_{j=1}^{p_n} P_{\lambda_n}(|\beta_{nj}|)$$

where the function $P_{\lambda_n}(.)$ is the SCAD penalty. Zou and Li (2008) proposed the local linear approximation to the SCAD penalty, which retains the same asymptotic properties. Moreover, this technique significantly improves the computational efficiency of the local quadratic approximation algorithm (Fan and Li, 2001)

when we use SCAD Penalty

The least absolute shrinkage and selection operator (LASSO) has been a popular regression estimator with simultaneous variable selection. However, LASSO does not have the oracle property and its robust version is needed in the case of heavy-tailed errors or serious outliers . We propose a robust penalized regression estimator which provide a simultaneous variable selection and estimator. It is based on the rank regression and the non-convex penalty function, the smoothly clipped absolute deviation (SCAD) function which has the oracle property. The proposed method combines the robustness of the rank regression and the oracle property of the SCAD penalty. We develop an efficient algorithm to compute the proposed estimator that includes a SCAD estimate based on the local linear approximation and the tuning parameter of the penalty function. Our estimate can be obtained by the least absolute deviation method. We used an optimal tuning parameter based on the Bayesian information criterion and the cross validation method. Numerical simulation shows that the proposed estimator is robust and effective to analyze contaminated data.

Asymptotic Properties of the SCAD Estimator with A Diverging Number of Parameters

In this section, we establish several theoretical properties of the LAD-SCAD estimator when the number of predictors diverges with increasing sample size. For simplicity, main assumptions required for our results are presented as follows.

(A1) The error ϵ_i has continuous and positive density f $(\cdot)$ at the origin. Moreover, the density function f $(\cdot)$ has finite derivatives in any neighborhood of 0.

(A2) There exists a positive constant M $< \infty$ such that $max_{1 \le i \le n, 1 \le j \le p_n} |x_{ij}| \le M$.

(A3)$p_n^3 / n --> 0$ as $n --> \infty$.

(A4) There exist constants $0 < \rho_1 < \rho_2 < \infty$ and $0 < \tau_1 < \tau_2 < \infty$ such that $\rho_1 \le \rho_{n_1} \le \rho_{n_2} \le \rho_2$ and $\tau_1 \le \tau_{n_1} \le \tau_{n_2} \le \tau_2$.

(A5) $\lambda_n --> 0$ and $\sqrt{\frac{n}{P_n \lambda_{n]}}} --> \infty$.

Condition (A1) is a very typical condition which is widely employed in the literature (Pollard, 1991; Knight, 1998; Wang et al., 2007).

Condition (A2) imposes the restriction on the covariates and assures the properties of the consistency and asymptotic normality.This condition is also applied by Huang and Xie (2007).

Condition (A3) implies that $P_n = o(n^{1/3})$ and is in line with that of Huber (1973).

Condition (A4) assumes that the matrices$E(xx^{'})$ and $E(ww^{'})$ are positive definite and is identical to the condition given in Huang et al. (2008) and Li et al. (2011).

Condition (A5) is the same as that assumed by Fan and Peng (2004, p. 936) and ensures the estimator with the sparsity property

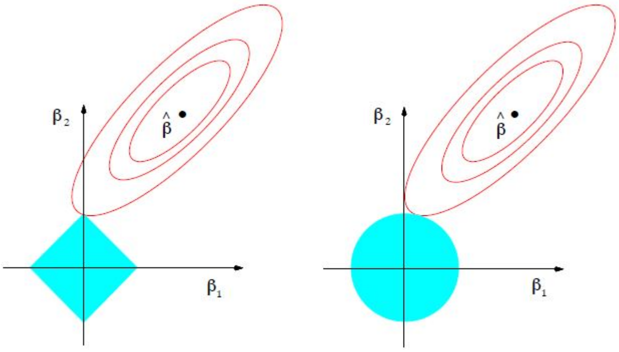
5 Lasso Regression

5.1 Theoretical analysis of the lasso

5.1.1 Geometric interpretation

As discussed above, lasso can set coefficients to zero, while ridge regression, which appears superficially similar, cannot. This is due to the difference in the shape of the constraint boundaries in the two cases. Both lasso and ridge regression can be interpreted as minimizing the same objective function

$\min_{\beta \epsilon R^p} \left\{ \frac{1}{N} \|y - X\beta\|_2^2 \right\}$ but with respect to different constraints: $\|\beta\|_1 \leq t$ for lasso and $\|\beta\|_2^2 \leq t$ for ridge.



From the figure, one can see that the constraint region defined by the L_1 norm is a square rotated so that its corners lie on the axes (in general a cross-polytope), while the region defined by the L_2 norm is a circle (in general an n-sphere), which is rotationally invariant and, therefore, has no corners. As seen in the figure, a convex object that lies tangent to the boundary, such as the line shown, is likely to encounter a corner (or a higher-dimensional equivalent) of a hypercube, for which some components of β are identically zero, while in the case of an n-sphere, the points on the boundary for which some of the components of β are zero are not distinguished from the others and the convex object is no more likely to contact a point at which some components of β are zero than one for which none of them are.

5.1.2 Convex relaxation interpretation

Lasso can also be viewed as a convex relaxation of the best subset selection regression problem, which is to find the subset of $\leq k$ covariates that results in the smallest value of the objective function for some fixed $k \leq n$, where n is the total number of covariates. The "ℓ^0 norm", $\|\cdot\|_0$, which gives the number of nonzero entries of a vector, is the limiting case of "ℓ^p norms", of the form $\|x\|_p = \left(\sum_{i=1}^{n} |x_j|^p\right)^{1/p}$ (where the quotation marks signify that these are not really norms for $p < 1$ since $\|\cdot\|_p$ is not convex for $p < 1$, so the triangle inequality does not hold). Therefore, since p = 1 is the smallest value for which the "ℓ^p norm" is convex (and therefore actually a norm), lasso is, in some sense, the best convex approximation to the best subset selection problem, since the region defined by $\|x\|_1 \leq t$ is the convex hull of the region defined by $\|x\|_p \leq t$ for $p < 1$.

5.2 Generalisation

There are different variants of Ł$_1$ penalty of lasso to overcome its limitation and to make the regularization more efficient for particular problem . They focus on utilizing the dependencies among covariates .Namely the different variants of lasso are elastic net , group lasso , fused lasso , Quasi - norm and bridge regression , adaptive lasso , prior lasso e.t.c. Determining the optimal value of regularization parameter is one of most important part, for the improvement of accuracy of prediction of the model , cross-validation helps us to select tuning or regularization parameter . Here , we are discussing each variants of lasso in brief to understand their need in a particular problem were the Ł$_1$penalty of lasso reaches its limitation and how the particular variants helps us in overcoming these limitations .

5.2.1 Elastic net

Elastic net uses L_1 and L_2 penalty combinedly. There is general problem withL_1 penalty of lasso in high dimensional data ("$p > n$"),as it select atmost n variable out of p and also if there is group of correlated variable it choose one variable from them and ignore the others. To overcome these limitations, the elastic net adds a quadratic part to the penalty ($\|\beta\|^2$), which when used alone is a ridge regression. The estimates of elastic net method are then defined by

$$\widehat{\beta} = \arg\min_{\beta} \|y - X\beta\|_2^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1$$

Now with the quadratic penalty term , we have strongly convex loss function and thus can obtain optimal minimum of estimates. Lasso and ridge regression can be written as a special case of Elastic net method : each of them can be obtained as $\lambda_1 = \lambda, \lambda_2 = 0$ or $\lambda_1 = 0, \lambda_2 = \lambda$. Elastic net method obtain's it's naive version in two stage : first for each fixed λ_2 it finds the ridge regression coefficients, and secondly it does a LASSO type shrinkage. This two stage estimation of elastic net method incurs a double amount of shrinkage which leads to increased bias and poor predictions . To improve the prediction, there is general re-scaling of the coefficients of the naive version of elastic net is done , by multiplying the estimates of coefficient by $(1+\lambda_2)$.

Examples of where the elastic net method has been applied are:
- Support vector machine
- Metric learning
- Portfolio optimization
- Cancer prognosis

5.2.2 GROUP LASSO

Group lasso is introduced by Yuan and Lin in 2006. The main objective of it is to include or exclude predefined groups of covariates in or out of the model together . The penalty function of group lasso is generalization of the L1 penalty of standard lasso and are as follows :

$$\min_{\beta\in\mathbb{R}^p}\left\{\left\|y-\sum_{j=1}^{p}X_j\beta_j\right\|_2^2+\lambda\sum_{j=1}^{p}\|\beta_j\|_{K_j}\right\},\qquad \|z\|_{K_j}=(z^tK_jz)^{1/2}$$

Here , the penalty term is sum over L_2 norm defined by the positive definite matrix K_j. There is special case of reduction of above penalty term to standard lasso when K_j = I. Also , if there is only single group and K_1=I. Then the above penalty term reduces to ridge regression.Since ,as we see from special case if the penalty reduces to L_2 norm on the sub spaces defined by each group then as like ridge regression it cannot select only some of the other covariates from group. Hence , the penalty is the sum over different sub-spaces norm , like the standard lasso, therefore ,coefficients vector corresponding to some sub-spaces is set to zero while others are shrink-ed.

There are particular settings where the group lasso is very useful and the most obvious one is when categorical variables levels are represented as binary covariates .One such settings is in biological studies where grouping is natural. As , pathways of genes and proteins are known and investigator most of the time interested in which pathways are related to an outcome rather than whether particular individual genes are.

5.2.3 FUSED LASSO

In some situation, the object being studied may have important spatial or mutable structure that must be accounted for during analysis, such as time series or image based data. In 2005, **Tibshirani** and colleagues proposed the Fused Lasso to extend the use of lasso to exactly this type of data. The fused lasso objective function is-

$$\widehat{\beta}=\arg\min\left\{\sum_i\left(y_i-\sum_j x_{ij}\beta_j\right)^2\right\}\qquad \text{subject to } \textstyle\sum_{j=1}^{p}|\beta_j|\le s_1 \text{ and } \sum_{j=2}^{p}|\beta_j-\beta_{j-1}|\le s_2.$$

The first constraint is just the typical lasso constraint, but the second directly penalizes large changes with respect to the mutable or spatial structure, which forces the coefficients to vary in a smooth fashion that reflects the underlying logic of the system being studied. Clustered lasso is a generalization to fused lasso that identifies and groups incidental covariates based on their effects (coefficients). The basic idea is to penalized the differences between the coefficients so that nonzero ones make clusters together. This can be modeled using the undermention regularization:

$\sum_{i<j}^{p} |\beta_i - \beta_j| \leq s_2$. In opposite, one can first set variables into highly correlated groups, and then reference a single representative covariate from each set.

various algorithms exist that solve the Fused lasso problem and some generalizations of in a direct form, i.e. there are algorithm that solve it exactly in a finite number of operations.

5.2.4 Quasi-norm and bridge regression

Lasso, elastic net, group and fused lasso construct the penalty functions from the L_1 and L_2 norms (with weights, if necessary). The bridge regression utilizes general ℓ_p norms ($p \geq 1$) and quasi-norms ($0 < p < 1$).For example, for $p = 1/2$ the proportion of lasso objective in the Lagrangian form is to solve

$$\min_{\beta\in\mathbb{R}^p} \left\{ \frac{1}{n} \|y - X\beta\|_2^2 + \lambda\sqrt{\|\beta\|_{1/2}} \right\}, \qquad \text{where } \|\beta\|_{1/2} = \left(\sum_{j=1}^{p} \sqrt{|\beta_j|} \right)^2$$

It is claimed that the fractional quasi-norms ℓ_p ($0 < p < 1$) provide more significant results in data analysis both from the theoretical and empirical perspective. But non-convexity of these quasi-norms causes difficulties in solution of the optimization problem. To solve this problem, an expectation-minimization process is developed and implemented for minimization of function

$$\min_{\beta\in\mathbb{R}^p} \left\{ \frac{1}{n} \|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} \nu(\beta_j^2) \right\},$$

where, $\nu(\gamma)$ is an arbitrary concave monotonically increasing function (for example, $\nu(\gamma) = \sqrt{\gamma}$ gives the lasso penalty and $\nu(\gamma) = \gamma^{1/4}$ gives the $\ell_{1/2}$ penalty).

The masterful algorithm for minimization is based on piece-wise quadratic approximation of sub-quadratic growth (PQSQ)

5.2.5 Adaptive lasso

Definition

We have shown that the lasso cannot be an oracle procedure. However, the asymptotic setup is somewhat unfair, because it forces the coefficients to be equally penalized in the L_1 penalty. We can certainly assign different weights to different coefficients. Let us consider the weighted lasso,

$$\underset{\beta\epsilon R^p}{\operatorname{argmin}} \|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} w_j |\beta_j|$$

where **w** is a known weights vector and and λ is a non negative regularization parameter.We show that if the weights are data-dependent and cleverly chosen, then the weighted lasso can have the oracle properties. The new methodology is called the adaptive lasso.
We now define the adaptive lasso. Suppose that $\widehat{\beta}^{init}$ is a root n– consistent estimator of regression coefficient(which could be given simply, e.g., by least squares).The adaptive lasso estimates are given by

$$\widehat{\beta}^{adapt} = \min_{\beta\epsilon\mathbb{R}^p} \frac{1}{2} \|y - X\beta\|_2^2 + \lambda_n \sum_{j=1}^{p} \widehat{w_j} |\beta_j| \qquad \text{- - - - -}(*)$$

for weights $\widehat{w}_j = 1/|\widehat{\beta}_j^{init}|^\gamma$, $j = 1, 2, ..., p$,where $\widehat{\beta}^{init}$ is some initial estimate (come from (say) ridge regression or the lasso itself) of the regression coefficients, and $\gamma > 0$ is another tuning parameter and $\lambda_n$varies with n .
Note that when the initial estimate $\widehat{\beta}^{init}$has zero components, this makes some weights infinite, which we would formally handle by introducing equality constraints into the problem $(*)$. Hence,since the active set of the adaptive lasso solution $\widehat{\beta}^{adapt}$ is always a subset of that of $\widehat{\beta}^{init}$, we would typically avoid making the initial estimate $\widehat{\beta}^{init}$ super sparse; e.g., if $\widehat{\beta}^{init}$ is fit via the lasso, then we might want to use a bit less regularization in the initial lasso problem

5.2.6 **Prior lasso**

The prior lasso was introduced by **Jiang et al**(2016) for generalized linear models to unified prior information' such as the weightage of certain covariates. In prior lasso, such information is summarized into pseudo responses (are called prior responses) $\hat{y}^{\mathrm{p}}$ and then an extra criterion function is added to the usual objective function of the generalized linear models with a lasso penalty. Without loss of generality, we use linear regression to illustrate prior lasso. In linear regression, the new objective function can be written as

$$\min_{\beta\in\mathbb{R}^p} \left\{ \frac{1}{n} \|y - X\beta\|_2^2 + \frac{1}{n}\eta \|\hat{y}^{\mathrm{p}} - X\beta\|_2^2 + \lambda\|\beta\|_1 \right\},$$

which is equivalent to

$$\min_{\beta\in\mathbb{R}^p} \left\{ \frac{1}{n} \|\tilde{y} - X\beta\|_2^2 + \frac{\lambda}{1+\eta} \|\beta\|_1 \right\},$$

the usual lasso objective function with the responses y being replaced by a weighted average of the observed response and the prior responses $\tilde{y} = (y + \eta\hat{y}^{\mathrm{p}})/(1+\eta)$ (are called the adjusted response values by the prior information).

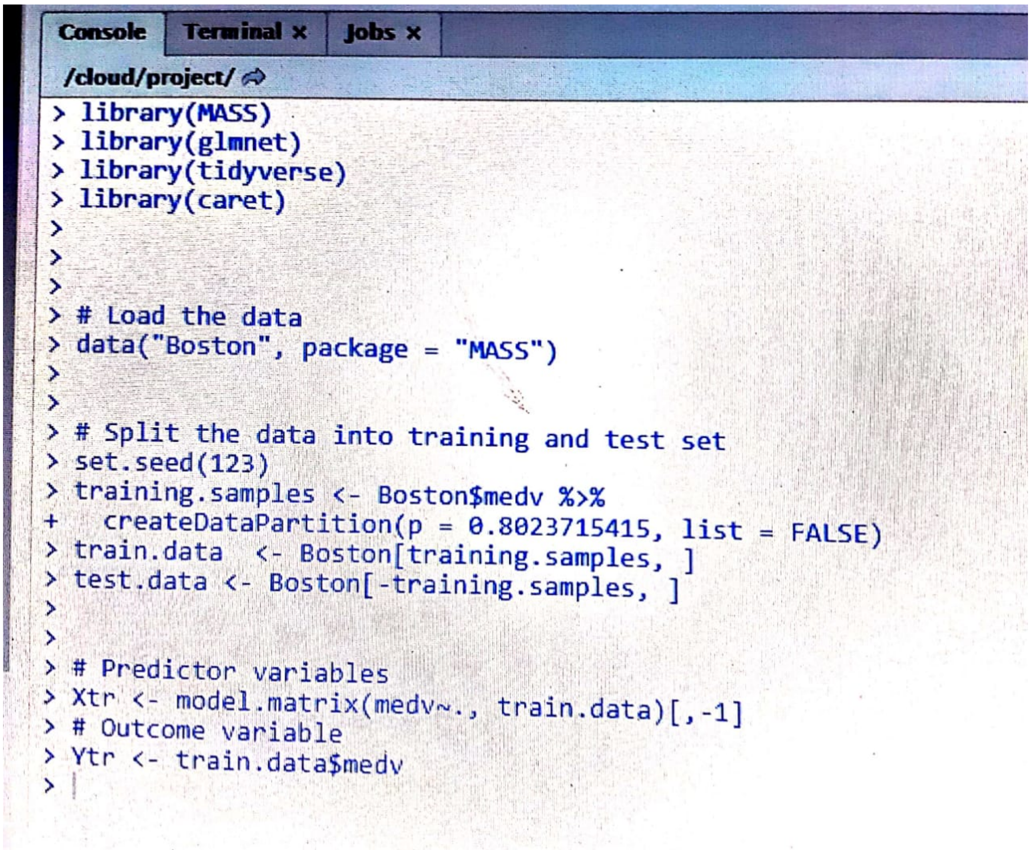
In prior lasso, the parameter η is called a balancing parameter, which balances the relative importance of the data and the prior information. In the extreme case of $\eta = 0$, prior lasso is reduced to lasso. If $\eta = \infty$, prior lasso will just remain the prior information to fit the model. Even, the balancing parameter η has another interesting interpretation: it controls the variance of β in its prior distribution from a Bayesian approach.

Prior lasso is more efficient in parameter estimation and prediction (with a smaller estimation error and prediction error) when the prior information is of high quality, and is durable to the low quality prior information with a good choice of the balancing parameter η.

6 Comparison of different penalised method and in terms of Root mean square error(RMSE) with real life data using R software

We'll use the Boston data set ,for predicting the median house value (mdev), in Boston Suburbs, based on multiple predictor variables. We'll randomly split the data into training set (406 observations for building a predictive model) and test set (rest 100 observations for evaluating the model). Make sure to set seed for reproductibility.

R package required - **"glmnet"** ,**"tidyverse"**,**"caret"**



```
> library(MASS)
> library(glmnet)
> library(tidyverse)
> library(caret)
>
>
>
> # Load the data
> data("Boston", package = "MASS")
>
>
> # Split the data into training and test set
> set.seed(123)
> training.samples <- Boston$medv %>%
+   createDataPartition(p = 0.8023715415, list = FALSE)
> train.data  <- Boston[training.samples, ]
> test.data <- Boston[-training.samples, ]
>
>
> # Predictor variables
> Xtr <- model.matrix(medv~., train.data)[,-1]
> # Outcome variable
> Ytr <- train.data$medv
>
```

here we create two objects:

1) y for storing the outcome variable.
2) x for holding the predictor variables.
This is done using the inbuilt function model.matrix() allowing to automatically transform any qualitative variables (if any) into dummy variables, which is important because glmnet() can only take numerical, quantitative inputs. After creating the model matrix, we remove the intercept component at index = 1.

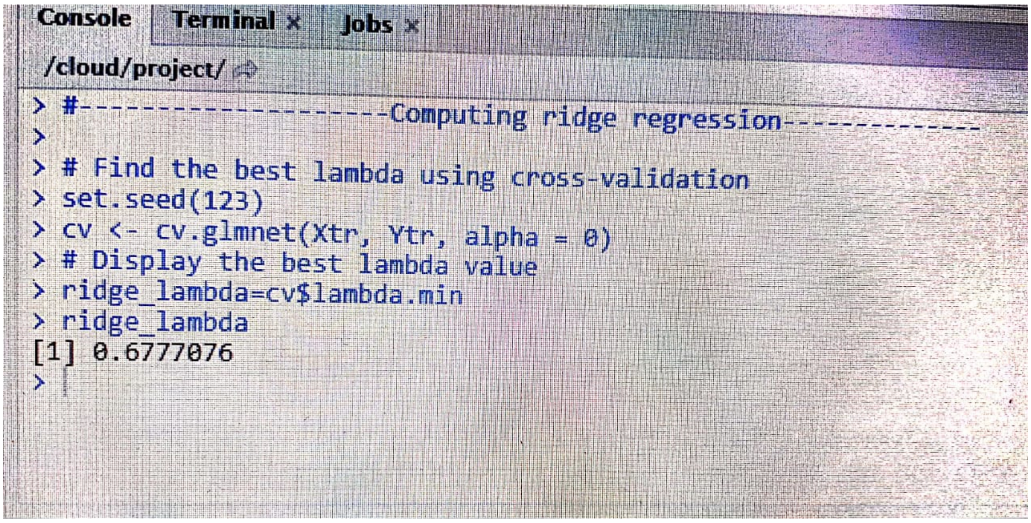
In penalized regression, we need to specify a constant λ to adjust the amount of the coefficient shrinkage. The best λ (also said optimise) for your data, can be defined as the λ that minimize the cross-validation prediction error rate. This can be determined automatically using the inbuilt function cv.glmnet().

The best model is defined as the model that has the lowest prediction error, root mean square error(RMSE).

Setting alpha=1 , R will perform Lasso algorithm.
Setting alpha=0 , R will perform Ridge algorithm
Setting alpha=0.5 , R will perform elastic net algorithm.

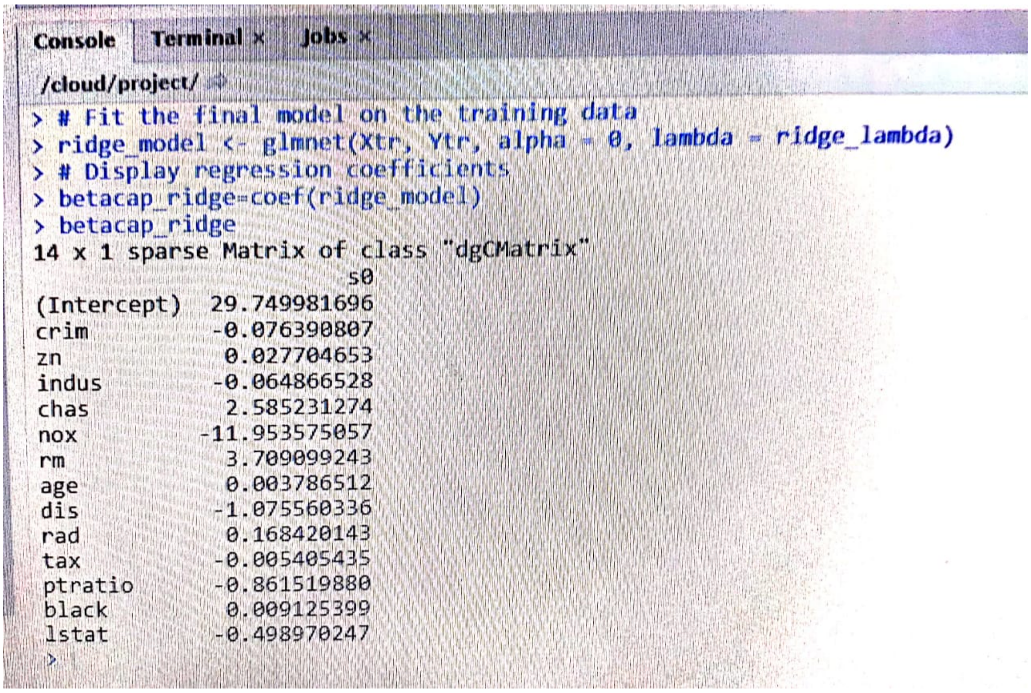
Below are the screenshots of the R script and results for various panelty we use for our data.

Ridge REGRESSION



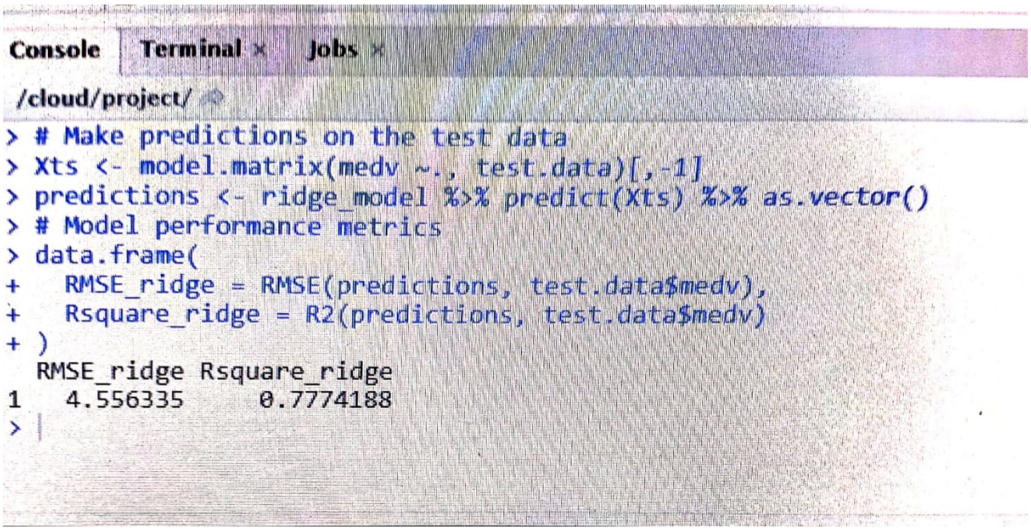
```
> #------------------Computing ridge regression-------------
>
> # Find the best lambda using cross-validation
> set.seed(123)
> cv <- cv.glmnet(Xtr, Ytr, alpha = 0)
> # Display the best lambda value
> ridge_lambda=cv$lambda.min
> ridge_lambda
[1] 0.6777076
>
```

we obtain λ optimum by the given code in R package.



```
> # Fit the final model on the training data
> ridge_model <- glmnet(Xtr, Ytr, alpha = 0, lambda = ridge_lambda)
> # Display regression coefficients
> betacap_ridge=coef(ridge_model)
> betacap_ridge
14 x 1 sparse Matrix of class "dgCMatrix"
                       s0
(Intercept)  29.749981696
crim         -0.076390807
zn            0.027704653
indus        -0.064866528
chas          2.585231274
nox         -11.953575057
rm            3.709099243
age           0.003786512
dis          -1.075560336
rad           0.168420143
tax          -0.005405435
ptratio      -0.861519880
black         0.009125399
lstat        -0.498970247
>
```

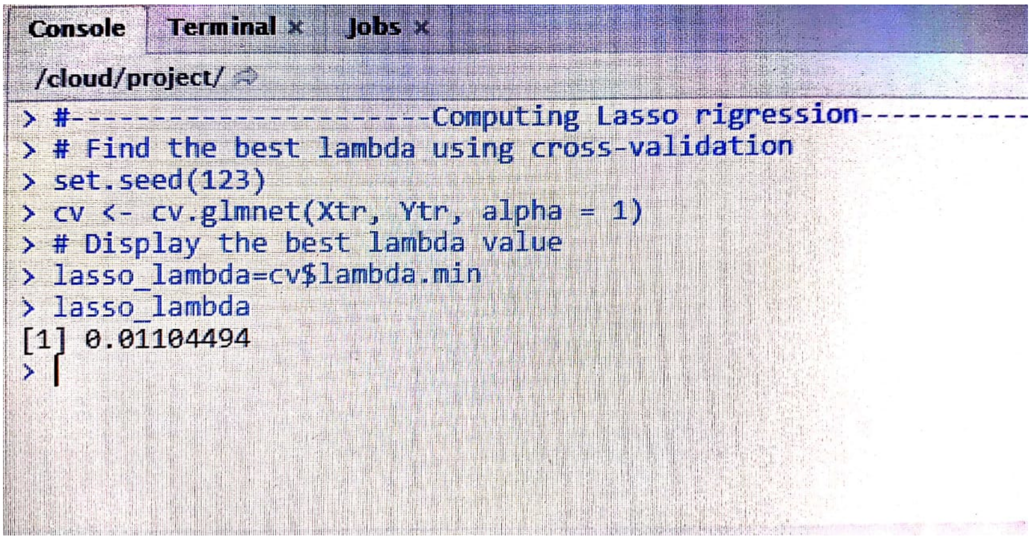
coefficients in given ridge penalty,



```
> # Make predictions on the test data
> Xts <- model.matrix(medv ~., test.data)[,-1]
> predictions <- ridge_model %>% predict(Xts) %>% as.vector()
> # Model performance metrics
> data.frame(
+   RMSE_ridge = RMSE(predictions, test.data$medv),
+   Rsquare_ridge = R2(predictions, test.data$medv)
+ )
  RMSE_ridge Rsquare_ridge
1   4.556335     0.7774188
> |
```

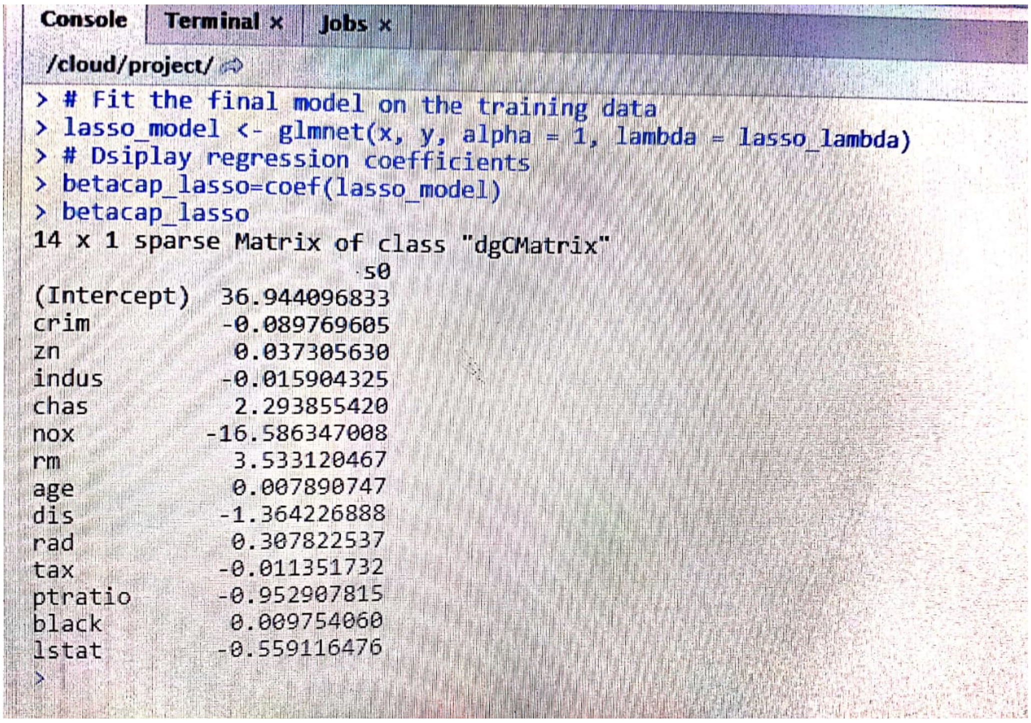
the root mean square error ($RMSE_{ridge}$) and corresponding R^2,

LASSO REGRESSION



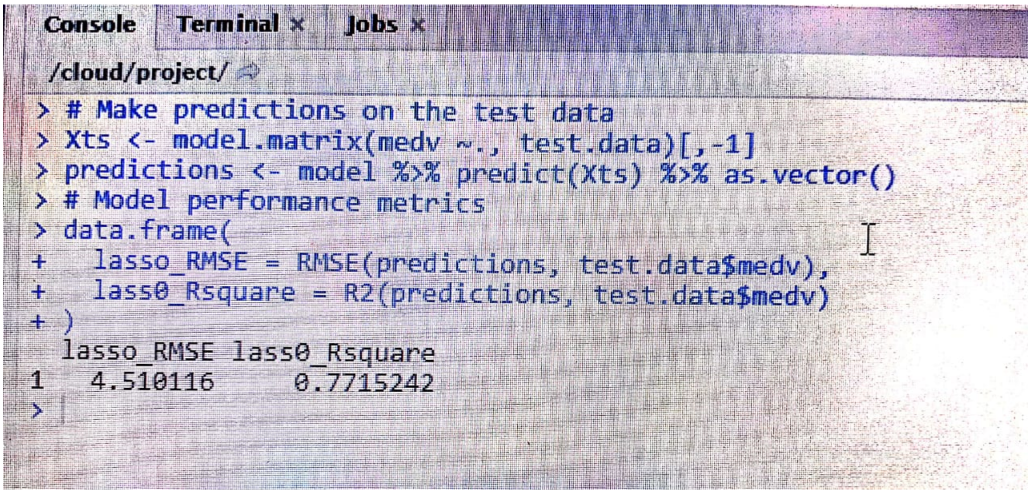
```
> #---------------------Computing Lasso rigression-----------
> # Find the best lambda using cross-validation
> set.seed(123)
> cv <- cv.glmnet(Xtr, Ytr, alpha = 1)
> # Display the best lambda value
> lasso_lambda=cv$lambda.min
> lasso_lambda
[1] 0.01104494
>
```

finding optimum λ in LASSO penalty,



```
> # Fit the final model on the training data
> lasso_model <- glmnet(x, y, alpha = 1, lambda = lasso_lambda)
> # Dsiplay regression coefficients
> betacap_lasso=coef(lasso_model)
> betacap_lasso
14 x 1 sparse Matrix of class "dgCMatrix"
                        s0
(Intercept)  36.944096833
crim         -0.089769605
zn            0.037305630
indus        -0.015904325
chas          2.293855420
nox         -16.586347008
rm            3.533120467
age           0.007890747
dis          -1.364226888
rad           0.307822537
tax          -0.011351732
ptratio      -0.952907815
black         0.009754060
lstat        -0.559116476
>
```

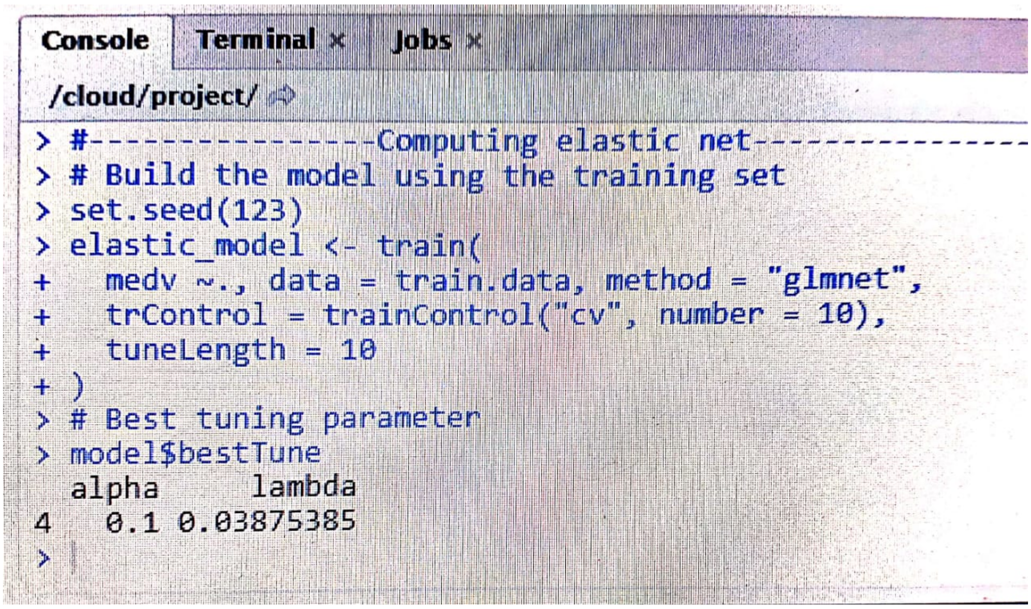
estimated coefficients in LASSO penalty,



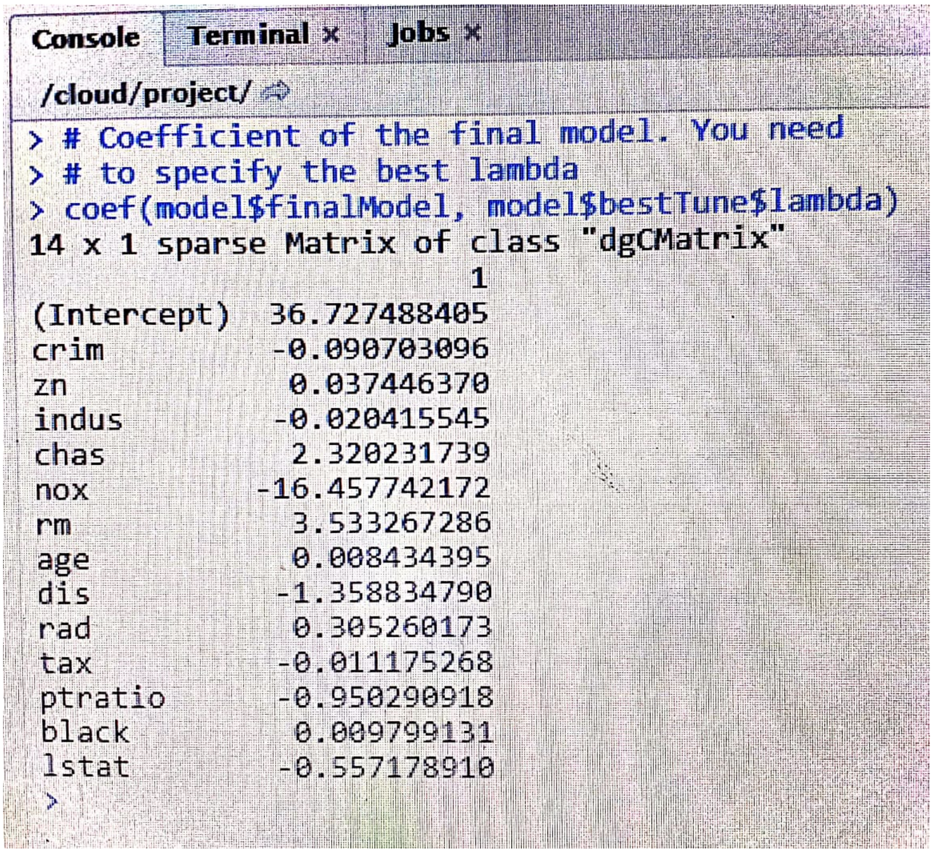
```
> # Make predictions on the test data
> Xts <- model.matrix(medv ~., test.data)[,-1]
> predictions <- model %>% predict(Xts) %>% as.vector()
> # Model performance metrics
> data.frame(
+   lasso_RMSE = RMSE(predictions, test.data$medv),
+   lass0_Rsquare = R2(predictions, test.data$medv)
+ )
  lasso_RMSE lass0_Rsquare
1   4.510116     0.7715242
>
```

the root mean square error ($RMSE_{lasso}$) and corresponding R^2,

ELASTIC NET REGRESSION

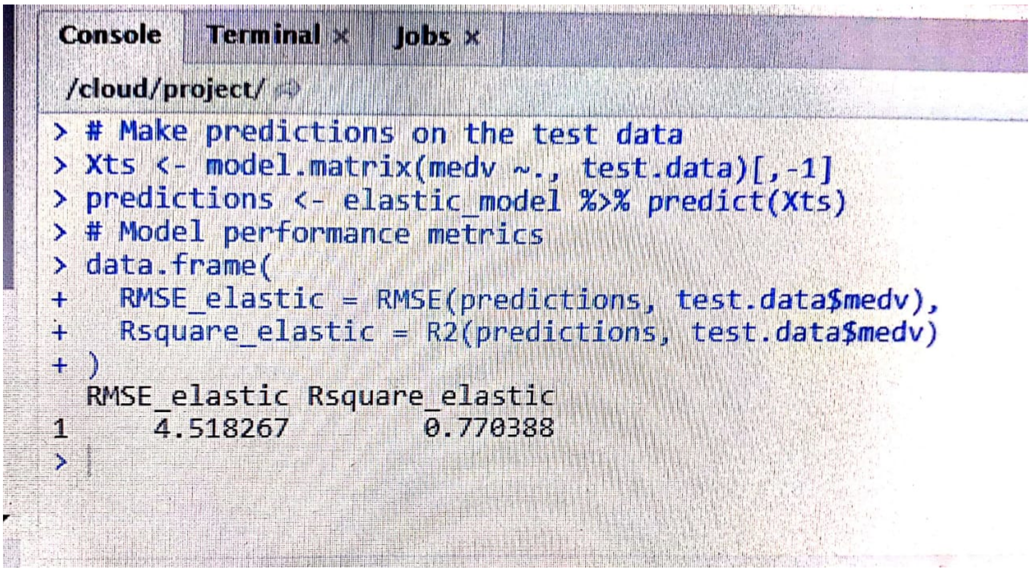


```
> #---------------Computing elastic net---------------
> # Build the model using the training set
> set.seed(123)
> elastic_model <- train(
+   medv ~., data = train.data, method = "glmnet",
+   trControl = trainControl("cv", number = 10),
+   tuneLength = 10
+ )
> # Best tuning parameter
> model$bestTune
  alpha     lambda
4   0.1 0.03875385
>
```



```
> # Coefficient of the final model. You need
> # to specify the best lambda
> coef(model$finalModel, model$bestTune$lambda)
14 x 1 sparse Matrix of class "dgCMatrix"
                        1
(Intercept)  36.727488405
crim         -0.090703096
zn            0.037446370
indus        -0.020415545
chas          2.320231739
nox         -16.457742172
rm            3.533267286
age           0.008434395
dis          -1.358834790
rad           0.305260173
tax          -0.011175268
ptratio      -0.950290918
black         0.009799131
lstat        -0.557178910
>
```

estimated coefficients in elastic panelty,



```
> # Make predictions on the test data
> Xts <- model.matrix(medv ~., test.data)[,-1]
> predictions <- elastic_model %>% predict(Xts)
> # Model performance metrics
> data.frame(
+   RMSE_elastic = RMSE(predictions, test.data$medv),
+   Rsquare_elastic = R2(predictions, test.data$medv)
+ )
  RMSE_elastic Rsquare_elastic
1     4.518267        0.770388
> 
```

the root mean square error ($RMSE_{elastic}$) and corresponding R^2, Now we will compare the RMSE(Root mean square error) for above penalised method .

Table 6.1: COMPARING RMSE OF PENALISED METHODS

Method	RMSE
Lasso	4.510116
Ridge	4.556335
Elastic-net	4.518267

CONCLUSION

The different models performance are comparable. Using ridge ,lasso or elastic net regression set the coefficient of the predictor variable age is nearly shrinks to zero, also from comparison table for RMSE above it is clear that in terms of RMSE LASSO is best from the above three penalty for our given data.

All things equal, we should go for the simpler model. In our example, we can choose the lasso regression model.

7 More about Lasso(research work in related field)

In this section, we introduce some possible applications of the LASSO to recent topics of machine learning research. Most examples are one of penalized estimations which may establish simple guidelines how to use the LASSO for statistical challenges of high-dimensional data analysis.

Unsupervised method of grouping variables and High dimensional clustering

Clustering analysis is widely used in many fields. Traditionally clustering is regarded as unsupervised learning for its lack of a class label or a quantitative response variable, which in contrast is present in supervised learning such as classification and regression.

In medical research

priority-Lasso, an intuitive and practical analysis strategy for building prediction models based on Lasso that takes such block structures into account. It requires the definition of a priority order of blocks of data. Lasso models are calculated successively for every block and the fitted values of every step are included as an offset in the fit of the next step.

8 BIBLIOGRAPHY

REFERENCES

[1] https://www.stat.cmu.edu/ ryantibs/advmethods/notes/highdim.pdf.

[2] http://www.csam.or.kr/journal/view.html?doi=10.29220/CSAM.2017.24.6.673.

[3] `http://www.hpc.unm.edu/~andriese/doc/ref1_fu.pdf`

.

[4] `https://web.stanford.edu/~hastie/Papers/Sparsenet/jasa_MFH_final.pdf`

.

[5] `https://web.stanford.edu/~hastie/StatLearnSparsity-files/SLS_correcte-1.4.16.pdf`

.

[6] https://web.stanford.edu/class/stats202/content/lec13.pdf.

[7] G. M. Furnival and R. W. Wilson. Regressions by leaps and bounds. *Technometrics*, 16(4):499–511, 1974.

[8] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

[9] P. Kumar and F. Kashanchi. On the lp-norm regression models for estimating value-at-risk. *Serdica Journal of Computing*, 8(3):255–268, 2014.

[10] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[11] R. Tibshirani and L. Wasserman. Sparsity, the lasso, and friends, 2017.

[12] W. van Wieringen. Lasso regression.

[13] F. Wang, S. Mukherjee, S. Richardson, and S. M. Hill. High-dimensional regression in practice: an empirical study of finite-sample prediction, variable selection and ranking. *Statistics and computing*, 30(3):697–719, 2020.

[14] Y. Wu and L. Wang. A survey of tuning parameter selection for high-dimensional regression. *Annual Review of Statistics and its Application*, 7:209–226, 2020.

[15] Y. Yang and H. Zou. A fast unified algorithm for solving group-lasso penalize learning problems. *Statistics and Computing*, 25(6):1129–1141, 2015.

[11] [14] [13] [15] [9] [8] [12] [1] [**?**] [2] [3] [4] [5] [6] [**?**] [7] [10]