Ridge, Lasso, and Elastic Net Regression

Ryan Henderson and Ryan Miller

4/15/22

1 Introduction

Ridge, Lasso, and Elastic Net are regularization methods that can be used in regression analysis in an attempt to improve the accuracy and prediction ability of a regression model. These methods were developed to improve the results of linear regression methods such as ordinary least squares (OLS). This OLS method of obtaining a simple linear model has widely acknowledged problems in prediction and interpretation, so regularization methods using penalties have been used to solve the issues faced by the method. [ZH05]

The main concept behind these regularization methods is to sacrifice some amount of bias in the model to reduce the variance of predicted values. The Ridge and Lasso regularization methods impose their own lambda penalty: L_2 and L_1 respectively. This penalty can be used with a linear regression model generated from an ordinary least squares operation to minimize the residual sum of squares (RSS). The Elastic Net method refers to a regularization method that combines the Ridge and Lasso methods by using both the L_2 and L_1 penalties from each method.

While each regularization method may accomplish the goal of reducing variance in predicted values, the methods have their own drawbacks. These drawbacks inform where each method is most useful in real world scenarios.

Ridge regression is an early regularization method with its roots in works by Andrey N. Tikhonov [TA77], David L. Philips [Phi62], and Arthur Hoerl and Robert Kennard [HK00]. This method is also sometimes known by the names Tikhonov regularization or Tikhonov-Phillips regularization. Ridge regression seeks to bias the original regression model by applying an L_2 penalty to the sum of each squared predictor and adding it to the residual sum of squares to be minimized. This method allows for better parameter estimation in a regression model, but can never remove any parameters. Because of the addition of the squared parameters to the residual sum of squares, mathematically, the value for β can only asymptotically approach zero. This lack of simplicity is the main drawback of the ridge regularization method. [HK00]

Lasso regression was developed by Robert Tibshirani [Tib96a]. It accomplishes the goal of reducing variance of predicted values by adding bias just like Ridge regression, but it also has the ability to produce a simpler model by selecting variables automatically. The Lasso regularization method seeks to bias the original regression model by applying an L_1 penalty to the sum of each predictor's absolute value and adding it to the residual sum of squares to be minimized. This method allows for parameter shrinkage and automatic parameter selection in the model simultaneously [ZH05]. Because of the addition of the absolute values of predictors to the residual sum of squares, mathematically, the value for β can become zero and remove predictors from the model. While this method solves for the problem of interpretability of the model not attained by Ridge regression, it has a few limitations of its own. Lasso regression models are limited by the sample size used generating the model, which limits variable selection when the number of predictors is greater than the sample size [ZH05]. The Lasso method also has trouble selecting between highly correlated predictors, so it generally selects one and does not care which one is selected [ZH05]. On top of these drawbacks, Lasso is also outperformed by Ridge in certain situations. [Tib96a]

The drawbacks from Ridge and Lasso led to the eventual development of the Elastic Net regularization method by Hui Zou and Trevor Hastie [ZH05]. Elastic Net regularization in regression allows the L_1 penalty from Lasso regularization and the L_2 penalty from Ridge regularization to combined linearly by adding them to the residual sum of squares to be minimized. Because of this, the Elastic Net method contains the possibility for both Lasso and Ridge regularization. It is strictly Lasso regularization when the L_2 lambda penalty is 0, and it is strictly Ridge regularization when the L_1 lambda penalty is 0. The combination of these penalties allows for the strengths of Lasso, such as simultaneous automatic variable selection and continuous shrinkage, and the strengths of Ridge, the selection of correlated variables. Elastic Net regularization also permits a data set where the number of predictors is greater than the sample size, unlike Lasso regularization, which was limited in selected predictors by the sample size of the data. According to the creators of the Elastic Net, this method "often outperforms the lasso, while enjoying a similar sparsity of representation" [ZH05].

2 Methods

2.1 Ordinary Least Squares

For the explanation of the Ridge, Lasso, and Elastic Net regularization methods, we will use the Ordinary Least Squares regression method as a starting point to show how these methods change the regression models.

We accept the following equation for a linear regression model: [JWHT13]

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon \tag{1}$$

where Y is the response variable, X are the predictor variables, β are the regression coefficients, ϵ is the error term, and p is the number of predictors.

We can make predictions based on the following formula by determining the estimated regression coefficients. [JWHT13]

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + ... + \hat{\beta}_p x_p \tag{2}$$

These coefficients can be solved by minimizing the residual sum of squares. [JWHT13]

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$RSS = \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - ... - \hat{\beta}_p x_{ip})^2 \tag{3}$$

The calculated values for $\hat{\beta}_0$, $\hat{\beta}_1$, $\hat{\beta}_2$, ..., $\hat{\beta}_p$ that minimize the RSS become the regression coefficients. The residual sum of squares with the regression coefficients shown below will be used in the following methods. [JWHT13]

$$RSS = \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 \tag{4}$$

This can also be shown in the following matrix notation. [ZH05]

$$RSS = (Y - X\beta)'(Y - X\beta)$$

For purposes of comparison, we show the variance-covariance matrix below. The trace of the variance-covariance matrix, the sum of the diagonal elements, yields the total variation of the model.

$$var(\hat{\beta}_{OLS}) = \sigma^2 (X'X)^{-1} \tag{5}$$

2.2 Regularization General Principle

In all of the regularization examples to follow, the general principle is to introduce bias to mitigate variance and improve out-of-sample predictions. The principle is best viewed in the bias-variance tradeoff graph below. By reducing the complexity of the model, such as through predictor elimination, we increase the bias of the model, and decrease our variance across the estimated parameters.

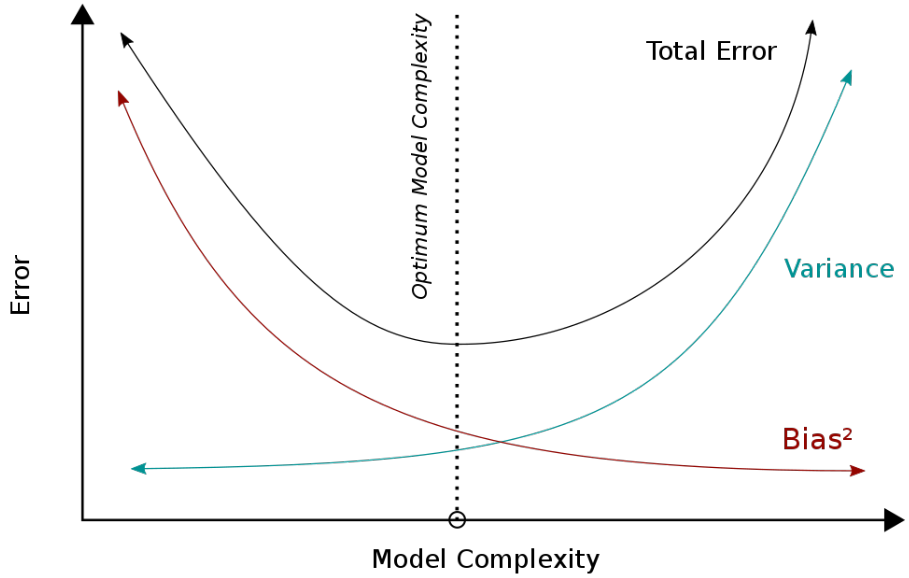

Figure 1: Bias and Variance Trade-off [FR12]

The following plot is not from any dataset, but intended to graphically show what regularized regression does in two dimensions. It's implied that the Regularized Regression line is intended to be fit from the two data points in the training data and bias is introduced to change the y prediction, not altering the slope itself.

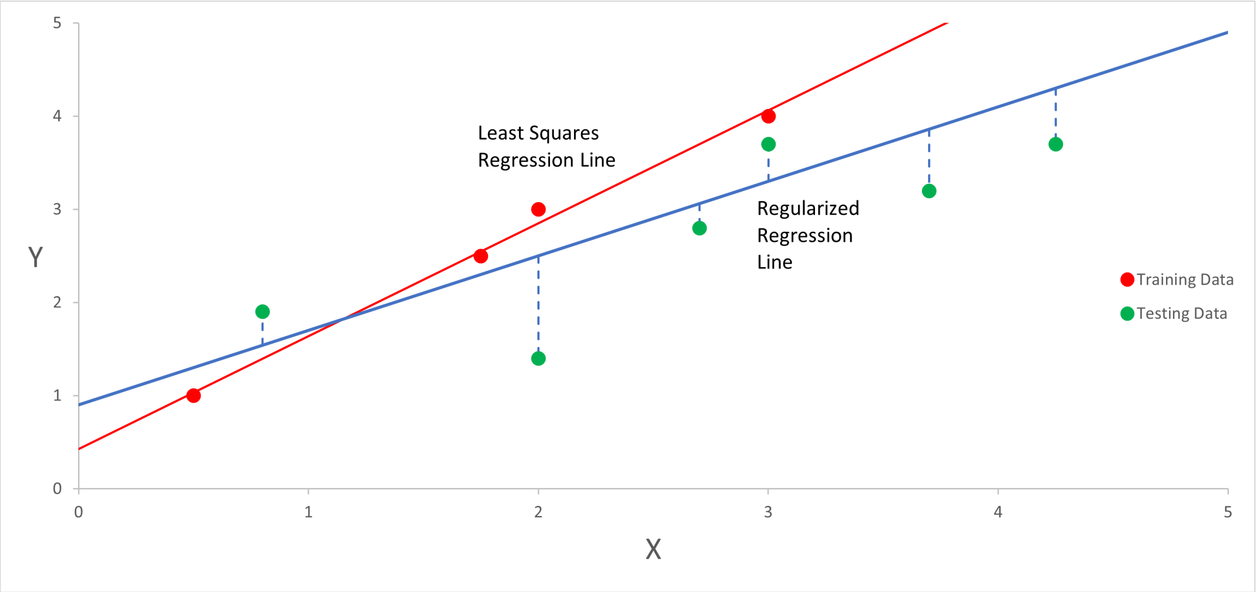

Figure 2: Least Squares and Regularized Regression Lines

2.3 Ridge Regularization

Ridge Regression uses an L_2 lambda penalty regularization to accomplish the bias-variance trade-off to improve the linear regression. The ridge penalty is sometimes known as the shrinkage penalty. It is shown below.[JWHT13]

$$\lambda_2 \sum_{j=1}^{p} \beta_j^2 \tag{6}$$

The λ_2 component is a tuning parameter that determines the impact of the penalty term on the regression. A value of zero results in the penalty having no effect and a value approaching infinity will have the greatest shrinkage effect with regression coefficients approaching zero. Ultimately, the value for λ_2 can be chosen via cross-validation. [JWHT13]

Using the residual sum of squares from Ordinary Least Squares method, we can add the ridge penalty to the argument of the minimum of the RSS to alter the regression coefficients obtained when minimizing. The Ridge Regression coefficient estimates $\hat{\beta}^R$ are found by minimizing the penalized residual sum of squares below.[JWHT13]

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda_2 \sum_{j=1}^{p} \beta_j^2$$

$$RSS + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \tag{7}$$

The penalized RSS to be minimized in Ridge Regression is shown below in matrix notation. [HK00]

$$||Y - X\beta||^2 + \lambda_2 ||\beta||_2^2$$

Solving for one β, our goal is to minimize $(y - x\beta)^2 + \lambda\beta^2$ by taking the derivative w.r.t. β and set equal to zero.

$$-2xy + 2x^2\beta + 2\beta\lambda = 0$$

$$\beta = xy/(x^2 + \lambda) \tag{8}$$

Observing the denominator, we can see that a coefficient can only shrink to zero with Ridge Regression as λ approaches infinity.

$$\lim_{\lambda_{Ridge} \to \infty} \beta = 0 \tag{9}$$

The variance-covariance matrix of the Ridge solution is then easily calculable because the underlying equation is linear and differentiable.[HK00]

$$var(\hat{\beta}_{ridge}) = \sigma^2 (X'X + \lambda I_p)^{-1} X'X (X'X + \lambda I_p)^{-1} \tag{10}$$

This squared shrinkage penalty is also applicable to the family of distributions in Generalized Linear Models.

$$\text{Poisson Ridge MLE: } \min_{\beta_0, \beta} - \left[\frac{1}{N} \sum_{i=1}^{N} \left(y_i(\beta_0 + \beta^T x_i) - e^{\beta_0 + \beta^T x_i} \right) \right] + \lambda ||\beta||^2 \tag{11}$$

$$\text{Binomial Ridge MLE: } \min_{(\beta_0, \beta) \in \mathbb{R}^{p+1}} - \left[\frac{1}{N} \sum_{i=1}^{N} y_i \cdot (\beta_0 + x_i^T \beta) - \log(1 + e^{(\beta_0 + x_i^T \beta)}) \right] + \lambda ||\beta||^2 \tag{12}$$

As we can see, the Ridge estimates utilize the same maximum likelihood estimates as in OLS, just with the penalty parameters.[MS11][CH92] Ridge Regression is useful when all or most covariates are significant. Andrew Ng compared L_1 and L_2 regularization using binomial regression and discovered L_2 regularization rapidly approaches a misclassification rate of 0.5.[Ng04] There is a significantly lower tolerance to the presence of irrelevant features compared to Lasso Regression, the L_1 penalty, which is discussed in the next section.

2.4 Lasso Regularization

Lasso Regression, also known as the L_1 penalty, is a regularization method used to decrease model variance and perform feature selection. Contrasting from the Ridge penalty, the objective function for Lasso uses an absolute value instead of a squared penalty shown below.[JWHT13]

$$\sum_{i=1}^{N}(y_i - \hat{\beta}_0 - \sum_{j=1}^{p}\hat{\beta}_j x_{ij})^2 + \lambda \sum_{j}|\beta_j| \tag{13}$$

$$= (Y - \beta X)'(Y - \beta X) + \lambda ||\beta||_1^{abs} \tag{14}$$

$$\text{With the constraint } \sum_{j=1}^{p}|\beta_j| \leq t$$

The tuning parameter λ is the vector coefficient which controls the amount of regularization and has one-to-one relation to the threshold t. λ is a tuning parameter which controls the level of sparsity in $\hat{\beta}$. For $\lambda = 0$, no regularization occurs and the OLS solution is yielded. Large enough λ values will set some coefficients to zero and perform model selection for the user. Ultimately, the value for λ_1 can be chosen via cross-validation. [JWHT13]

A major difference in calculating Lasso is its non-differentiability and non-linearity due to the absolute value sign. The standard error is still an unresolved component of LASSO regression research due to these properties. Robert Tibshirani discussed the convergence of his method via introducing the inequality constraints sequentially for which there are 2^p different possible signs for β_j coefficients for $j = \{1, 2, ..., p\}$.[Tib96b] As we can see, if p is large, the number of calculations needed to converge is not practically calculable without a computer. Therefore, convergence was defined as

$$||\hat{\beta}^{new} - \hat{\beta}^{old}||^2 \leq 10^{-5} \tag{15}$$

or if the fraction of explained deviance reaches 0.999 [FHT10a]. To better understand this convergence for the LASSO, the orthonormal X case results are shown in the following derivation with a soft-thresholding form (Donoho & Johnstone, 1994)[DJ94]:

$$\hat{\beta}_i^{lasso} = sgn(\hat{\beta}_i^{LS})(|\hat{\beta}_i^{LS}| - \gamma)^+ \tag{16}$$

A study of error variance estimation in LASSO regression was conducted in 2014 and the cross-validation based LASSO residual sum of squares estimator (CVL) was recommended (Tibshirani, Reid, & Friedman, 2016) [RTF16]. The following equation is the CVL variance estimate, which tends to be close to σ^2 in simulations conducted by Tibshirani with credit to Fan, Guo, and Hao (2012)[FGH12].

$$\hat{\sigma}^2_{L,\hat{\lambda}} = \frac{1}{n - \hat{s}_{L,\hat{\lambda}}}\sum_{i=1}^{n}(Y_i - X_i'\hat{\beta}_{\hat{\lambda}})^2 \text{ where } \hat{s}_{L,\hat{\lambda}} \text{ is the number of covariates } \neq 0 \tag{17}$$

In the paper "A significance test for the lasso", the authors propose a covariance test statistic which follows a Exp(1) asymptotic distribution. Typically, when comparing nested models, a chi-squared test would be used to compare the drop in RSS to the chi-squared distribution χ^2_1. That test is inappropriate for the Lasso because additional variables are not fixed, not chosen at random, and produce a large type I error in simulations (Lockhart et al., 2014) [LTTT14].

$$T_k = (\langle y, \mathbf{X}\hat{\beta}(\lambda_{k+1})\rangle - \langle y, \mathbf{X}_A\tilde{\beta}_A(\lambda_{k+1})\rangle)/\sigma^2 \tag{18}$$

The null hypothesis is all truly active variables are contained in the current lasso model. Predictor variables are tested by switching them in and out via a stagewise algorithm (Least Angle Regression) similar to Forward Selection [EHJT04]. As in Ridge regression, adding the penalty term to the maximum likelihood estimator will yield the minimized solution to Lasso.

2.5 Elastic Net Regularization

Elastic Net Regression combines the L_2 lambda penalty from Ridge and the L_1 lambda penalty from Lasso to create a new penalty. This penalty can be seen below. [ZH05]

$$\lambda_2 \sum_{j=1}^{p} \beta_j^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| \tag{19}$$

The λ_2 and λ_1 components are tuning parameter that determine the impact of the penalty term on the regression. Obviously, if $\lambda_2 = 0$, the penalty is just the Lasso regularization penalty, and if $\lambda_1 = 0$, the penalty is the Ridge regularization penalty. The optimal values for λ_2 and λ_1 can be found with cross-validation and training.

This new Elastic Net penalty is combined with the residual sum of squares from the Ordinary Least Squares method. This results in the following penalized residual sum of squares to be minimized to find the Elastic Net regression coefficients $\hat{\beta}^{EN}$. [ZH05]

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda_2 \sum_{j=1}^{p} \beta_j^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j|$$

$$RSS + \lambda_2 \sum_{j=1}^{p} \beta_j^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| \tag{20}$$

The penalized RSS to be minimized in Elastic Net Regression is shown below in matrix notation. [ZH05]

$$||Y - X\beta||^2 + ||\beta||_2^2 + ||\beta||_1^{abs}$$

In the naive Elastic Net definition, the tuning parameters are set by the following equation. [ZH05]

$$\alpha = \lambda_2/(\lambda_1 + \lambda_2)$$

With this α value, the naive Elastic Net penalized residual sum of squares becomes: [ZH05]

$$RSS + \alpha \sum_{j=1}^{p} \beta_j^2 + (1-\alpha) \sum_{j=1}^{p} |\beta_j|$$

The naive Elastic Net coefficients are found by minimizing the penalized residual sum of squares. [ZH05]

$$\hat{\beta}(naive) = \arg\min_{\beta}\{RSS + \alpha \sum_{j=1}^{p} \beta_j^2 + (1-\alpha) \sum_{j=1}^{p} |\beta_j|\} \tag{21}$$

This naive Elastic Net regularization does not have great prediction performance unless it is very close to Ridge Regression or Lasso Regression. Because shrinkage occurs with both penalties, there is double shrinkage. To correct this problem, the naive Elastic Net regression coefficients are rescaled with the following equation: [ZH05]

$$\hat{\beta}(corrected) = (1 + \lambda_2)\hat{\beta}(naive)$$

2.6 Glmnet

For applied use in R, the penalties for Ridge, Lasso, and Elastic Net can be simplified by using a single λ value shown in the following gaussian equation used in the Glmnet function. [SFHT11]

$$\lambda\left(\frac{(1-\alpha)}{2}\sum_{j=1}^{p}\beta_j^2+\alpha\sum_{j=1}^{p}|\beta_j|\right) \tag{22}$$

For Ridge regression, the α value is set to zero to remove the L_1 penalty. For Lasso regression, the α value is set to one to remove the L_2 penalty. The best λ values for Ridge and Lasso regression will be explored through cross validation. The best values for λ and α in this equation for Elastic Net regression will be explored through cross validation and training methods in the Examples with R section.3

3 Examples with R

The R package glmnet gives the option to select the measure of interest in cross-validation (method: cv.glmnet). The default is type.measure="deviance", which uses MSE for Gaussian models. Deviance can be used for logistic and Poisson regression models. Depending on the data, class or AUC (area under the ROC curve) is used for various types of logistic regression models. There are more options for the cv.glmnet method but we will focus on the MSE measure with Gaussian.

Upon fitting the models in cv.glmnet, we receive the corresponding lambda values, mean cross-validated errors for the fitted models ($cvm), non-zero coefficient counts ($nzero), coefficient values ($glmnet.fit$beta), the value of lambda giving the minimum mean cross-validated error ($lambda.min), and the most parsimonious value of lambda such that the model's error is within 1 standard error of the minimum error ($lambda.1se).

3.1 Simplified

	x_1	x_2	y
Training Data	5	4	45
	4	7	23
	3	12	65
	2	8	23
	5	9	76
Testing Data	2	5	40
	6	7	33
	8	3	50

Code:

```
training_data <- data.frame(
  x1 = c(5,4,3,2,5),
  x2 = c(4,7,12,8,9),
  y = c(45,23,65,23,76)
)

testing_data <- data.frame(
  x1 = c(2,6,8),
  x2 = c(5,7,3),
  y = c(40,33,50)
)
```

```
# make into matrices for glmnet compatibility
x.train <- as.matrix(training_data[,c("x1","x2")])
x.test <- as.matrix(testing_data[,c("x1","x2")])

y.train <- as.matrix(training_data[,c("y")])
y.test <- as.matrix(testing_data[,c("y")])

# train our models
linear <-lm(y ~ ., data = training_data)

ridge <- glmnet::cv.glmnet(x = x.train, y = y.train,
                        type.measure = "mse", family="gaussian",
                        alpha = 0, standardize = FALSE)

lasso <- glmnet::cv.glmnet(x = x.train, y = y.train,
                        type.measure = "mse", family="gaussian",
                        alpha = 1, standardize = FALSE)

# Typically we don’t explicitly set alpha for Elastic Net.
# It would find the optimal alpha value on its own.
elasticNet <- glmnet::cv.glmnet(x = x.train, y = y.train,
                             type.measure = "mse", family="gaussian",
                             alpha = 0.5, standardize = FALSE)

results <- data.frame(
  y = testing_data$y,
  ols.pred = as.vector(predict(linear, newdata = testing_data)),
  ridge.pred = as.vector(predict(ridge, newx=x.test, s="lambda.min")),
  lasso.pred = as.vector(predict(lasso, newx=x.test, s="lambda.min")),
  elnet.pred = as.vector(predict(elasticNet, newx=x.test, s="lambda.min"))
)
results$‘(y-ols.pred)^2‘ <- (results$y - results$ols.pred)^2
results$‘(y-ridge.pred)^2‘ <- (results$y - results$ridge.pred)^2
results$‘(y-lasso.pred)^2‘ <- (results$y - results$lasso.pred)^2
results$‘(y-elnet.pred)^2‘ <- (results$y - results$elnet.pred)^2

mapply(results[,6:9],FUN=mean)
```

The resulting models:

$$\text{Ordinary Least Squares}: \ \hat{y_i} = -70.056 + 15.634x_1 + 7.131x_2$$

$$\text{Ridge}: \ \hat{y_i} = -70.056 + 15.634x_1 + 7.131x_2 + 2.66(|15.634| + |7.131|)$$

$$\text{Lasso}: \ \hat{y_i} = -70.056 + 15.634x_1 + 7.131x_2 + 1.966(15.634^2 + 7.131^2)$$

$$\text{Elastic Net}: \ \hat{y_i} = -70.056 + 15.634x_1 + 7.131x_2 + 2.469\left(\frac{(1-0.5)}{2}(15.634^2 + 7.131^2) + 0.5(|15.634| + |7.131|)\right)$$

Notice the intercept of the model is not penalized. This is because penalizing the intercept in the model breaks many of the properties of linear regression. Penalization of the intercept would make the procedure depend on the origin chosen for Y. Adding a constant c to each of the targets y_i would not simply result in a shift of the predictions by the same amount c. [HTF09] In our simple example, the predictions for each model on the testing data are shown below. I showed the Mean Squared Error term underneath to demonstrate the out-of-sample variance reduction from the regularized model performance.

y	$\hat{y}_{OLS}$	$\hat{y}_{Ridge}$	$\hat{y}_{Lasso}$	$\hat{y}_{ElasticNet}$
40	-3.133919	1.476181	3.082157	2.844663
33	73.665203	70.282253	69.553623	69.553623
50	76.411636	71.444607	70.678804	70.678804

$(y - \hat{y}_{OLS})^2$	$(y - \hat{y}_{Ridge})^2$	$(y - \hat{y}_{Lasso})^2$	$(y - \hat{y}_{ElasticNet})^2$
1860.5349	1484.0862	1362.9272	1380.5191
1653.6587	1389.9677	1335.3591	1336.1673
697.5745	459.8724	434.4152	427.6129
$\frac{(y-\hat{y}_{OLS})^2}{n} = 1403.923$	$\frac{(y-\hat{y}_{Ridge})^2}{n} = 1111.309$	$\frac{(y-\hat{y}_{Lasso})^2}{n} = 1044.234$	$\frac{(y-\hat{y}_{EN})^2}{n} = 1048.100$

3.2 Used Car Auction Prices

The following examples will use the data set ”Used car auction prices” [Tun21], which gives historical sales figures from car auctions from 1982 to 2015. With all the factors in the data, each model built will have k-1 dummy variables as predictors in the model. The model contains each dummy variable and each integer and double variable for a total of 172 predictor variables. We will use all the information aside from the selling price as predictor variables to estimate that selling price as the response variable. The data set has been cleaned up for use with the various functions used in the examples. It has also been split up into training and testing sets with a split of 85 percent and 15 percent, respectively.

3.2.1 Example 1: Exploring Regularized Regression Methods with Glmnet

In this example, the whole range of the Elastic Net Regression will be explored. This will include a stepped process that inherently includes both Ridge Regression ($\alpha = 0$) and Lasso Regression ($\alpha = 1$), as well as everything in between.

Using the simplified Elastic Net penalty shown in Formula 22 [HTF01], we will obtain regularized regression models for each value of α in the range of 0 to 1 in increments of 0.1. This will be done using a cross-validated Glmnet function on our training data set, minimizing the model by the MSE measurement [FHT10b, SFHT11]. To accomplish this, the following code is run in R.

```r
## Glmnet Testing of Alpha Values

fits <- list()

for (i in 0:10)
{
  fit.name <- paste0("alpha", i/10)

  ## Cross-validated GLMNET function on training data
  fits[[fit.name]] <- cv.glmnet(x.train,
                                y.train,
                                type.measure="mse",
                                alpha=i/10,
                                family="gaussian")
}
```

The code creates a list of Elastic Net Regression models that can be used to predict the selling price of used cars. Using the predictions of these models, we calculate important metrics to help us evaluate the performance of the models. These metrics include RMSE, R^2, Adjusted R^2, MAPE, and MSPE. The λ value is taken from the Glmnet model where the best model's cross-validated error is within one standard error of the minimum mean cross-validated error. The code to obtain the predictions and metrics is shown below.

```
## Glmnet Model Predictions and Metrics

results <- data.frame()

for (i in 0:10)
{
  fit.name <- paste0("alpha", i/10)

  ## Predictions
  predicted <- predict(fits[[fit.name]],
                       s=fits[[fit.name]]$lambda.1se,
                       newx=x.test)

  ## Evaluation Metrics
  rmse <- RMSE(predicted, y.test)
  r.squared <- R2(predicted, y.test)
  MAPE <- mean(abs((y.test - predicted)/y.test))
  MSPE <- mean(((y.test - predicted)/y.test)^2)
  adj.r.squared <- 1 - ((1 - r.squared)*(dim(baseline.df)[1]-1))
      /(dim(baseline.df)[1]-dim(baseline.df)[2])

  ## Lambda From Glmnet Model
  lambda <- fits[[fit.name]]$lambda.1se

  temp <- data.frame(fit.name=fit.name,
                     alpha=i/10,
                     lambda=lambda,
                     rmse=rmse,
                     r.squared=r.squared,
                     MAPE=MAPE,
                     MSPE=MSPE,
                     adj.r.squared=adj.r.squared,
                     predictors=coef(fits[[fit.name]], s="lambda.1se")p[2])
  results <- rbind(results, temp)
}

results
```

This code produces the following table of models and their corresponding evaluation metrics. An α value of 0 produces the Ridge Regression model, and an α value of 1 produces the Lasso Regression model. Models with an α value closer to 0 will more closely resemble Ridge Regression and models with an α value closer to 1 will more closely resemble Lasso Regression. With that being said, any α value greater than zero gives the model the ability to eliminate predictor variables, as well as shrinkage.

	Lasso Model	Lasso Model	Ridge Model	Ridge Model
	(Minimum MSE)	(1 Standard Error)	(Minimum MSE)	(1 Standard Error)
λ	6.085879	68.36414	946.7656	946.7656
RMSE	\$1,548.93	\$1,584.86	\$1,917.83	\$1,917.83
R^2	97.0322%	95.4262%	84.4%	84.4%
Adj. R^2	97.0276%	95.4257%	84.36%	84.36%
MAPE*	14.07%	14.04%	19.88%	19.88%
MSPE**	19.46%	18.9%	42.38%	42.38%
Predictors	107	8	172	172

*MAPE: Mean Absolute Percentage Error

**MSPE: Mean Squared Prediction Error

Table 1: Comparative Results for Lasso & Ridge Models with training set dimensions as (391,895, 172), testing set dimensions as (69,159, 172), and five-fold cross-validation.

Furthermore, the following Elastic Net results use the same five-fold cross-validation as in the Lasso and Ridge regularization results. We can view fluctuations in the λ_α and resulting $RMSE$ values because the learning algorithm reaches local minima at varying values of α, such as in gradient descent.

Model	α	λ	RMSE	R^2	Adj. R^2	MAPE	MSPE	Predictors
alpha0	0	946.7656	1917.825227	0.844	0.8436	0.1988	0.4238	172
alpha0.1	0.1	34.8256	1548.966574	0.9649	0.9648	0.1414	0.1969	129
alpha0.2	0.2	23.0187	1548.552966	0.967	0.967	0.1411	0.1958	117
alpha0.3	0.3	18.484	1548.833895	0.9678	0.9677	0.1409	0.1951	112
alpha0.4	0.4	13.863	1548.918232	0.9687	0.9686	0.1409	0.1949	112
alpha0.5	0.5	11.0904	1548.848916	0.9693	0.9693	0.1408	0.1950	109
alpha0.6	0.6	10.1431	1548.931175	0.9695	0.9694	0.1408	0.1947	108
alpha0.7	0.7	8.6941	1548.891641	0.9698	0.9697	0.1408	0.1946	107
alpha0.8	0.8	7.6073	1548.782128	0.9700	0.9699	0.1408	0.1946	107
alpha0.9	0.9	6.762	1548.866039	0.9701	0.9701	0.1407	0.1946	107
alpha1	1	6.0858	1548.926828	0.9703	0.9702	0.1407	0.1946	107

Table 2: Elastic Net GLMNET Models and Metrics

Analyzing the table, we would like to choose a model based on reducing the MSE and RMSE values, while maximizing the R^2 and Adjusted R^2 values. While we only have a scale of 0.1 for α, we can see a maximum of R^2 and Adjusted R^2 values in the range $\alpha = 0.7$ to $\alpha = 1$. The RMSE appears to be minimized near $\alpha = 0.2$, and MAPE appears minimized near $\alpha = 0.9$ to $\alpha = 1$.

These models may be among the best models for the data set, but given the discrete nature and scale of the alphas chosen, we are unable to see the full picture of how every alpha affects the model. However, we are able to get a big picture idea of where the better Elastic Net Regression models exist on the scale of Ridge Regression to Lasso Regression.

Out of the model options in the table, any model with an $\alpha > 0.7$ could be used in predictions on our dataset. In this range, the model performed the best according to the evaluation metrics. All of these models selected 107 variables of the available 171 predictor variables in the data set.

This method could be improved by continuing to explore α values at a smaller interval scale, but this would likely be a processing-inefficient solution to finding the best values for α and λ. A more streamlined option is shown in the following example.

3.2.2 Example 2: Best Tuned Model with CARET Training

In this example, the CARET package will be used to train the model [Kuh21]. The model will be trained using 10-fold cross validation, and the regression method is specified as GLMNET.

With this trained model, we can determine the best tuned model according to the function. The best tuned model will be selected by the training function based on the metrics RMSE and Adjusted R^2 by default for regression. [Kuh21]

The code to train the model and access the α and λ values for the best tuned model is shown below.

```
## CARET Training Best Tuned Model

## Training the Model
elasticnetmodel <- train(sellingprice ~ .,
                         data = train,
                         method = "glmnet",
                         trControl = trainControl("cv", number = 10),
                         tuneLength = 20)

## Best Tuned Model Alpha and Lambda
tune <- elasticnetmodel$bestTune
tune
```

The resulting α and λ values for the best tuned model are shown below in Table 3.

α	λ
0.194737	16.93881

Table 3: CARET Training Best Tuned Model α and λ Values

Using this CARET best tuned model, we will predict the selling price of used cars at auction from our testing data set. As in the GLMNET example, the metrics RMSE, R^2, Adjusted R^2, MAPE, and MSPE will be used to evaluate the model. The code for the predictions and the evaluation metrics is shown below.

```
## CARET Training Model Predictions and Metrics

## Predictions
predictions <- predict(elasticnetmodel, test)

## Evaluation Metrics
rmse2 <- RMSE(predictions, y.test)
rsquared2 <- R2(predictions, y.test)
adj.r.squared2 <- 1 - ((1 - r.squared2)*(dim(baseline.df)[1]-1))
    /(dim(baseline.df)[1]-dim(baseline.df)[2])
MAPE <- mean(abs((y.test - predicted)/y.test))
MSPE <- mean(((y.test - predicted)/y.test)^2)

results2 = data.frame(tune=tune,
                      rmse2=rmse2,
                      rsquared2=rsquared2,
                      adj.r.squared2=adj.r.squared2,
                      MAPE=MAPE,
                      MSPE=MSPE)

results2
```

Table 4 summarizes the best tuned model obtained from the CARET function, which optimizes the MSE, and corresponding metrics.

α_{tuned}	λ_{tuned}	**RMSE**	R^2	**Adj** R^2	**MAPE**	**MSPE**	**Predictors**
0.194737	16.93881	1530.166	0.9685902	0.9685889	0.1413	0.1964	132

Table 4: CARET Training Best Tuned Model and Metrics

3.2.3 Conclusion

Compared to the model selected in the Glmnet example, the CARET best tuned model achieves a smaller RMSE value. The CARET best tuned model also achieves similar results on maximizing the R^2 and Adjusted R^2 metrics. These metrics are less than in the Glmnet models where $\alpha > 0.4$, but this difference is very small.

The CARET trained model selected 121 variables of the available 171 predictor variables in the data set, while the chosen Glmnet models had 107 variables. This could possibly make a significant difference in use of the model, depending on the variables and the use of the model.

Between the models produced by the Glmnet and CARET examples, different models may be selected for use depending on the needs of the task using a data set like the Used Car Auction Sales. Processing capabilities and speeds would need to be considered in both the model generation and prediction processes. The CARET tuned model may have some slightly better evaluation metrics, but the Glmnet model selected produces a more parsimonious model with less variables. Because the metrics for the Glmnet models are very close together, different alpha models could be chosen to maximize or minimize a chosen metric. Depending on the use, there could be significant arguments for one model over another.

References

[CH92] S. Le Cessie and J. C. Van Houwelingen. Ridge estimators in logistic regression. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 41(1):191–201, 1992.

[DJ94] David L Donoho and Jain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[EHJT04] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[FGH12] Jianqing Fan, Shaojun Guo, and Ning Hao. Variance estimation using refitted cross-validation in ultrahigh dimensional regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(1):37–65, 2012.

[FHT10a] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[FHT10b] Jerome H. Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[FR12] Scott Fortmann-Roe. Understanding the bias-variance tradeoff. *http://scott.fortmann-roe.com/docs/BiasVariance.html*, 2012.

[HK00] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 42(1):80–86, 2000.

[HTF01] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer New York Inc., 2001.

[HTF09] Trevor Hastie, Robert Tibshirani, and Jerome H Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, volume 2. Springer, 2009.

[JWHT13] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R.* Springer, 2013.

[Kuh21] Max Kuhn. *caret: Classification and Regression Training*, 2021. R package version 6.0-90.

[LTTT14] Richard Lockhart, Jonathan Taylor, Ryan J Tibshirani, and Robert Tibshirani. A significance test for the lasso. *Annals of statistics*, 42(2):413, 2014.

[MS11] Kristofer Månsson and Ghazi Shukur. A poisson ridge regression estimator. *Economic Modelling*, 28(4):1475–1481, 2011.

[Ng04] Andrew Y Ng. Feature selection, l_1 vs. l_2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78, 2004.

[Phi62] David L. Phillips. A technique for the numerical solution of certain integral equations of the first kind. *J. ACM*, 9(1):84–97, 1962.

[RTF16] Stephen Reid, Robert Tibshirani, and Jerome Friedman. A study of error variance estimation in lasso regression. *Statistica Sinica*, 26:35–67, 2016.

[SFHT11] Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13, 2011.

[TA77] Andrey N. Tikhonov and Vasiliy Y. Arsenin. *Solutions of ill-posed problems.* V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York, 1977. Translated from the Russian, Preface by translation editor Fritz John, Scripta Series in Mathematics.

[Tib96a] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[Tib96b] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[Tun21] Bojan Tunguz. Used car auction prices, May 2021.

[ZH05] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.