PIPHER, BRANDON P., December 2019 APPLIED MATHEMATICS

COMPARISON OF REGRESSION METHODS WITH NON-CONVEX PENALTIES (27 pages)

Thesis Advisor: Omar De la Cruz Cabrera

We examine a solution to the problem of sparse selection in linear models. The method used is a mixed norm ℓ_p-ℓ_q algorithm with a focus on non-convex, $q < 1$, penalty parameters. Classical regression, Ordinary Least Squares, has low bias but high variance and prediction accuracy can sometimes be improved by increasing bias to decrease variance. By inducing sparsity we can improve model interpretability, especially in the setting of high-dimensional data. These methods of penalized regression also provide solutions when the Ordinary Least Squares solution is ill-posed under a high-dimensional setting, and have a history of producing accurate and parsimonious models. A simulation study is conducted utilizing another method of penalized regression using non-convex penalties, the SparseNet algorithm, which had previously been compared independently against several other proposed sparsity inducing non-convex solutions. We also include a comparison with other more common penalties such as LASSO, Ridge/Tikhonov, and Elastic Net.

COMPARISON OF REGRESSION METHODS WITH NON-CONVEX PENALTIES

A thesis submitted

to Kent State University

in partial fulfillment of the requirements

for the degree of Master of Science

by

Brandon P. Pipher

December 2019

© Copyright

All rights reserved

Except for previously published materials

Thesis written by

Brandon P. Pipher

B.S., University of Akron, 2017

M.S., Kent State University, 2019

Approved by

Omar De la Cruz Cabrera, Advisor

Andrew M. Tonge, Chair, Department of Mathematical Sciences

James L. Blank, Dean, College of Arts and Sciences

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

My deepest gratitude to my advisor Omar for all of his guidance these past two years. I attribute the skills developed to his wisdom and expertise, and his support of intellectual curiosity has been invaluable.

A special thanks to Dr. Kazim Khan for his time and support with my studies. The insight he offered through all of my questions has been most appreciated.

And most of all a very special thank you to my wife whose constant motivation and relentless support helps me to strive forward, and whose companionship brings a greater meaning to each and every one of my achievements.

CHAPTER 1

Introduction

In this chapter we will introduce the concept of Regression Analysis and related methods of statistical modeling. *Regression Analysis* entails using statistical techniques to examine the relationship between two or more variables. These variables are grouped as being either dependent, or independent variables. *Dependent variables*, often referred to as the response or criterion variable, are those whose outcomes we are interested in studying. *Independent variables*, frequently referred to as either predictors, regressors, covariates, or explanatory variables; are the variables we believe to influence the outcome of our dependent variables. The goal of Regression Analysis is to study variation of the dependent variables through a function consisting of the independent variables, which we refer to as the *regression function*.

1.1 Linear Regression

One of the elementary statistical models of Regression Analysis is the *Simple Linear Regression Model* (SLR). In this model we assume both a single response, Y, and predictor, X, with n observations. We also assume that their relationship is by a linear combination of β's, and so our regression function is $f(x, \beta) = \beta_0 + \beta_1 x$, but with some error referred to as ϵ which is independent from X. In scalar form, representing a single observation, our models equation is as follows: $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$. If we continue using this equation for each observation we can use an equivalent Model Matrix form $Y = X\beta + \epsilon$, which in expanded form is:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Under this structure our X and Y are given, and it leaves β to be estimated. We refer to our

estimation of β by $\hat{\beta}$ where $Y \approx \hat{Y} = X\hat{\beta}$.

We refer to the deviation of our approximation from the true values of Y as the *residual*, which in scalar form for a SLR model is $e_i = y_i - \hat{y}_i = y_i - (\beta_0 + \beta_1 x_i)$. Our residuals represent the data's unexplained variation under our model, and can also be expressed as a vector: $e = Y - \hat{Y} = Y - X\hat{\beta}$. Note that at this point we can proceed to a *Multiple Linear Regression* (MLR) problem, or one where β as a vector contains two or more predictors, without loss of generality by utilizing this Model Matrix form and the subsequent linear algebra. The expanded form of our Model Matrix form $Y = X\beta + \epsilon$ generalized to any number of predictors would then be notated as such:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & \dots & x_{1,m} \\ 1 & x_{2,1} & \dots & x_{2,m} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n,1} & \dots & x_{n,m} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Optionally a model can be fit without an intercept. In the above expanded form β_0 and the column of 1's would create a model with a constant that acts as the intercept. By either using 0's instead of 1's, or omitting the columns of 1's and truncating β_0 from $\hat{\beta}$, we would be fitting a model without an intercept.

1.1.1 Ordinary Least Squares

Classically, our criteria for determining $\hat{\beta}$ is by minimizing our *Sum of Squared Residuals*, which leads to the *Ordinary Least Squares* (OLS) solution to our approximation. Note that we will utilize the notation for norms where if $\vec{w} = (w_1, w_2, \dots, w_n)$ then $||w||_p = \sqrt[p]{\sum_{i=1}^{n} |w_i|^p}$.

$$\text{Scalar Form OLS Solution} : \hat{\beta} = \arg\min_{\beta} \sum (y_i - \hat{y}_i)^2$$

$$\text{Matrix Form OLS Solution} : \hat{\beta} = \arg\min_{\beta} ||Y - X\beta||_2^2$$

Choosing to minimize the Sum of Squared Residuals is not a unique solution to our problem, however. Squaring is beneficial as it forces our terms to be positive, thus allows penalizing both over and undershooting the true model. However, this could also be done by choosing to minimize the sum of the absolute differences, $\sum |y_i - \hat{y}_i|$ or $||Y - X\beta||_1$. One predominant benefit for squaring is that the differentiability allows us to find this minimum through calculus, which under certain

conditions is also a unique solution. Squaring the residuals also has the effect of penalizing outliers compared to taking their absolute. Other motivations arise under the OLS solution through stronger assumptions, such as if our errors, ϵ, being Independent and Identically Distributed (iid) Normal such that $\epsilon \sim N(0, \sigma^2 \mathbb{1})$. Under this normally distributed error assumption we have the OLS solution as both the Maximum Likelihood Estimator (MLE) along with being the Best Linear Unbiased Estimator (BLUE) by the Gauss-Markov theorem. We will explore some of these motivations further.

1.1.2 Normal Equations

Following the OLS solution to our approximation of $\hat{\beta} = \arg\min_\beta ||Y - X\beta||_2^2$ we can find a closed form solution through calculus. Note that in matrix form we can represent the sum of squared residuals as $e^T e = (Y - X\beta)^T (Y - X\beta)$ and thus:

$$\begin{aligned} e^t e = (Y - X\beta)^T (Y - X\beta) &= Y^T Y - Y^T X\beta - \beta^T X^T Y + \beta^T X^T X\beta \\ &= Y^T Y - 2\beta^T X^T Y + \beta^T X^T X\beta \end{aligned} \tag{1.1}$$

The last step followed by making use of $Y^T X\beta$ being a scalar and thus we have the property that $Y^T X\beta = (Y^T X\beta)^T = \beta^T X^T Y$. To find the value of β that minimizes $e^T e$ we will need to take the both the first and second derivative with respect to β. This leads us to the following equations [14, Equations (69) and (81)]:

$$\frac{\partial e^t e}{\partial \beta} = -2X^T Y + 2X^T X\beta = -2X^T (Y - X\beta) \tag{1.2}$$

$$\frac{\partial^2 e^t e}{\partial \beta^2} = 2X^T X \tag{1.3}$$

Therefore if X has full column rank, and so the columns are linearly independent, we have $2X^T X$ as a positive definite matrix as for all $v \neq \vec{0}$ we have $v^T X^T X v = (Xv)^T (Xv) = ||Xv||_2^2 > 0$ as $||Xv||_2^2$ will equal zero if and only if $Xv = 0$ which is true only if $v = \vec{0}$. We thus know that that the solution to our first derivative $-2X^T (Y - X\beta)$ equaling zero will be a minimum by the second partial derivative test as (1.2) is the gradient of $e^t e$ and (1.3) is the hessian of $e^t e$ which is positive definite.

Solving for when the first derivative equation is zero gives us a series of equations referred to as the *Normal Equations*, derived by $-2X^T (Y - X\hat{\beta}) = 0$ implying that $X^T X\hat{\beta} = X^T Y$.

$$\text{(Normal Equations) } X^T Y = X^T X\hat{\beta} \tag{1.4}$$

However, we only have a unique solution if X^TX is full rank. This is the case if we also have more observations than Independent Variables, as we had previously assumed X to be of full column rank. With all of these assumptions we then know that the nullspace of X is trivial and so $Xv = 0$ if and only if $v = 0$, which is a property shared by X^TX, and therefore it is an invertible matrix.

Under these assumptions we can solve for a unique solution of $\hat{\beta}$ by multiplying both sides of the Normal Equations by $(X^TX)^{-1}$. This solution is referred to as the Ordinary Least Squares (OLS) solution. The matrix we multiply Y by to obtain $\hat{Y}$ is referred to colloquially as the *Hat Matrix* as it puts the hat on Y. These equations are as follows:

$$\text{(OLS Solution): } \hat{\beta} = (X^TX)^{-1}X^TY \tag{1.5}$$

$$\text{(Hat Matrix): } H = X(X^TX)^{-1}X^T \tag{1.6}$$

$$\hat{Y} = X\hat{\beta} = X((X^TX)^{-1}X^TY) = (X(X^TX)^{-1}X^T)Y = HY \tag{1.7}$$

1.1.3 Hat Matrix

The Hat Matrix above is actually an orthogonal projection matrix, implying that $HH = H$ and $H^T = H$, as verified below:

$$HH = X(X^TX)^{-1}X^TX(X^TX)^{-1}X^T = X(X^TX)^{-1}X^T = H \tag{1.8}$$

$$H^T = (X(X^TX)^{-1}X^T)^T = (X^T)^T((X^TX)^{-1})^T(X)^T = H \tag{1.9}$$

With this observation we can also note that the OLS solution can be derived from a geometric interpretation as a projection onto the linear space spanned by the regressors. This geometric interpretation is portrayed in the three dimensional, two regressors, case in Figure 1. Under this geometric interpretation our estimate of Y is shown as the vector comprised of linear combinations of regressors, x_1 and x_2, where the vector of residuals $e = Y - X\hat{\beta}$ is at its shortest length. This minimum occurs when the residual vector is orthogonal to the column space of X.

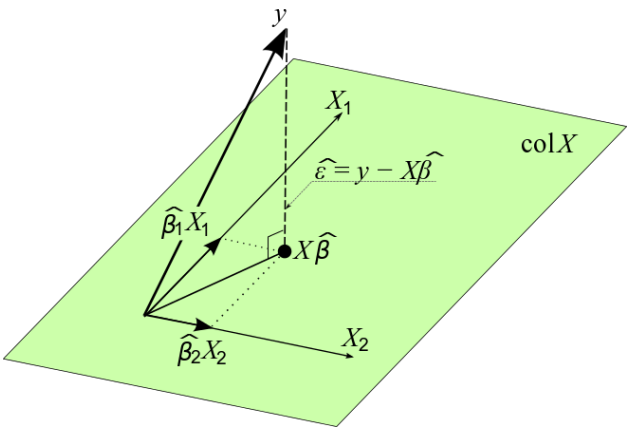


Figure 1: OLS Projection with two regressors (SLR)[1]

1.1.4 OLS as MLE

With an even greater assumption on our original model we can also show that the OLS solution is both the Maximum Likelihood Estimator and the Best Linear Unbiased Estimator. The additional constraint we will add is that our error terms are i.i.d. Normal. That is, $\epsilon \sim N(0, \sigma^2\mathbb{1})$ where $\mathbb{1}$ is the Identity Matrix and noting this is a Multivariate Normal distribution. We then have that the conditional distribution of Y given X is again a Multivariate Normal distribution, such that $Y|X \sim N(X\beta, \sigma^2\mathbb{1})$, which gives us the following likelihood and log-likelihood functions:

$$L(\beta, \sigma^2|X) = \frac{1}{(2\pi)^{n/2}(\sigma^2)^{n/2}} e^{\frac{-1}{2}(Y-X\beta)^T(\sigma^2\mathbb{1})^{-1}(Y-X\beta)} \tag{1.10}$$

$$\begin{aligned}
\mathcal{L}(\beta, \sigma^2|X) &= \log(L(\beta, \sigma^2|X)) \\
&= \frac{-n}{2}\ln(2\pi) - \frac{n}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}(Y - X\beta)^T(Y - X\beta) \\
&= \frac{-n}{2}\ln(2\pi) - \frac{n}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}(Y^T - \beta^T X^T)(Y - X\beta) \\
&= \frac{-n}{2}\ln(2\pi) - \frac{n}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}(Y^TY - Y^TX\beta - \beta^TX^TY + \beta^TX^TX\beta) \\
&= \frac{-n}{2}\ln(2\pi) - \frac{n}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}(Y^TY - 2\beta^TX^TY + \beta^TX^TX\beta)
\end{aligned} \tag{1.11}$$

The final line follows from $Y^TX\beta$ being a scalar, and so equal to its transpose. Note that the log-function is monotonic, and so maximizing the log-likelihood simultaneously maximizes the likelihood function. The derivatives of the log-likelihood is then [14, Equations (69) and (81)]:

$$\frac{\partial\mathcal{L}(\beta, \sigma^2)}{\partial\beta} = -\frac{1}{2\sigma^2}(-2X^TY + 2X^TX\beta) \tag{1.12}$$

[1]Public Domain image uploaded by user Stpasha on WikiMedia.org

$$\frac{\partial^2 \mathcal{L}(\beta, \sigma^2)}{\partial \beta^2} = -\frac{1}{2\sigma^2}(2X^T X) \tag{1.13}$$

By setting (1.12) equal to zero we arrive at the Normal equations as also being the Maximum Likelihood Estimator of $\hat{\beta}$ such that $X^T Y = X^T X\hat{\beta}$, noting that (1.13) being the hessian which is negative definite guarantees this as the maximum of the likelihood function.

1.1.5 OLS as BLUE

To then also show that the OLS solution is BLUE under our iid normally distributed errors we will first show that it is unbiased. That is we wish to prove $\mathrm{E}[\hat{\beta}_{\mathrm{OLS}}] = \beta$, as shown below:

$$\begin{aligned}
\mathrm{E}[\hat{\beta}_{\mathrm{OLS}}|X] &= \mathrm{E}[(X^T X)^{-1} X^T Y|X] \\
&= (X^T X)^{-1} X^T E[Y|X] = (X^T X)^{-1} X^T X\beta = \beta \\
&\Rightarrow \mathrm{E}\left[\,\mathrm{E}[\hat{\beta}_{\mathrm{OLS}}|X]\right] = \mathrm{E}[\hat{\beta}_{\mathrm{OLS}}] = \beta
\end{aligned}$$

We will then compute $\mathrm{Var}[\hat{\beta}_{\mathrm{OLS}}|X]$, noting the following:

$$\begin{aligned}
\hat{\beta} - \beta &= (X^T X)^{-1} X^T Y - \beta \\
&= (X^T X)^{-1} X^T (X\beta + \epsilon) - \beta \\
&= (X^T X)^{-1} X^T \epsilon
\end{aligned}$$

Using the above we can compute the conditional variance:

$$\begin{aligned}
\mathrm{Var}[\hat{\beta}_{\mathrm{OLS}}|X] &= \mathrm{E}[(\hat{\beta} - \beta)(\hat{\beta} - \beta)^T|X] \\
&= \mathrm{E}[(X^T X)^{-1} X^T \epsilon\epsilon^T X (X^T X)^{-1}|X] \quad \text{as } (X^T X)^T = (X)^T (X^T)^T = (X^T X) \\
&= (X^T X)^{-1} X^T \,\mathrm{E}[\epsilon\epsilon^T|X] X (X^T X)^{-1} \\
&= (X^T X)^{-1} X^T \sigma^2 \mathbb{1} X (X^T X)^{-1} \\
&= \sigma^2 (X^T X)^{-1}
\end{aligned}$$

We can now show that the OLS solution has the minimum variance amongst all linear unbiased estimators of β. To do this we will first define $\tilde{\beta} = Cy$ where $C = (X^T X)^{-1} X^T + D$ where D is

any $m \times n$ non-zero matrix. We therefore have that:

$$\begin{aligned}
\text{Var}[\tilde{\beta}|X] &= \text{Var}[Cy|X] = C\,\text{Var}[y|X]C^T = \sigma^2 CC^T \\
&= \sigma^2((X^TX)^{-1}X^T + D)((X^TX)^{-1}X^T + D)^T \\
&= \sigma^2((X^TX)^{-1}X^TX(X^TX)^{-1} + (X^TX)^{-1}X^TD^T + DX(X^TX)^{-1} + DD^T) \\
&= \sigma^2((X^TX)^{-1} + (X^TX)^{-1}X^TD^T + DX(X^TX)^{-1} + DD^T) \\
&= \sigma^2((X^TX)^{-1} + (X^TX)^{-1}(DX)^T + DX(X^TX)^{-1} + DD^T)
\end{aligned}$$

Now note that if $\tilde{\beta} = Cy$ is to be an unbiased estimator we must have the following:

$$\text{E}[\tilde{\beta}|X] = \text{E}[Cy|X] = \text{E}[CX\beta + C\epsilon|X] = \beta$$

This then implies that $CX = \mathbb{1}$ and so $CX = (X^TX)^{-1}X^TX + DX = \mathbb{1} + DX = \mathbb{1}$ and thus $DX = \vec{0}$. We therefore find that:

$$\begin{aligned}
\text{Var}[\tilde{\beta}|X] &= \sigma^2((X^TX)^{-1} + (X^TX)^{-1}(DX)^T + DX(X^TX)^{-1} + DD^T) \\
&= \sigma^2(X^TX)^{-1} + DD^T
\end{aligned}$$

Note that DD^T is a positive semi-definite matrix as for all $a \neq \vec{0}$ we have $a^TDD^Ta = (D^Ta)^T(D^Ta) = ||D^Ta|| \geq 0$. We therefore know that $\text{Var}[\hat{\beta}|X] \leq \text{Var}[\tilde{\beta}|X]$. The variance of each estimator is then computed below, remembering they are unbiased and the variance of a constant is zero:

$$\begin{aligned}
\text{Var}[\hat{\beta}] &= \text{E}_X[\text{Var}[\hat{\beta}|X]] + \text{Var}_X[E[\hat{\beta}|X]] = \text{E}_X[\text{Var}[\hat{\beta}|X]] \\
\text{Var}[\tilde{\beta}] &= \text{E}_X[\text{Var}[\tilde{\beta}|X]] + \text{Var}_X[E[\tilde{\beta}|X]] = \text{E}_X[\text{Var}[\tilde{\beta}|X]]
\end{aligned}$$

Thus we have by the monotonicity property of the Lebesgue integral that since $\text{Var}[\hat{\beta}|X] \leq \text{Var}[\tilde{\beta}|X]$ we then know $\text{E}_X[\text{Var}[\hat{\beta}|X]] \leq \text{E}_X[\text{Var}[\tilde{\beta}|X]]$ and therefore can conclude that $\text{Var}[\hat{\beta}] \leq \text{Var}[\tilde{\beta}]$.

Note that if we were not in the scenario of having our error terms as i.i.d. Normal, but instead had uncorrelated homoscedastic errors independent of X with mean zero under any arbitrary distribution, we would still retain the OLS solution as the Best Linear Unbiased Estimator. Furthermore, note that care was taken on whether we considered X to be stochastic or not when proving the Gauss-Markov theorem and that the result holds regardless.

1.2 Penalized Regression

The ordinary least squares approach to linear regression can perform poorly under situations where there are more variables than samples. Under these scenarios the Normal Equations cannot derive a unique solution due to the matrix X^TX being singular, resulting in an ill-posed problem as there becomes an infinite number of solutions. This can be a very common problem in certain fields such as genetics and with the study of gene micro-array data. We can address this issue by imposing greater constraints onto our model, such as by penalizing the number of variables. Imposing this penalty can allow us to shrink the coefficients of our variables towards zero on the basis of which variables are contributing the least to the model. Three of the more common types of penalized regression are the Ridge, LASSO, and Elastic Net.

1.2.1 Ridge Regression

Ridge regression shrinks the coefficients by creating a penalty on their size. The penalized residual sum of squares for ridge regression is:

$$\hat{\beta}_{ridge} = \arg\min_{\beta} \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \text{ constrained by } \sum_{j=1}^{p}\beta_j^2 \leq t \tag{1.14}$$

for some value of t and p predictors with N samples. It can equivalently be stated in a Lagrangian form as:

$$\hat{\beta}_{ridge} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 \right\} \tag{1.15}$$

where λ constrains the size of the coefficients instead of t. By utilizing the above we can easily rewrite the equation into a matrix form:

$$\hat{\beta}_{ridge} = \arg\min_{\beta} \left\{ (Y - X\beta)^T(Y - X\beta) + \lambda\beta^T\beta \right\} \tag{1.16}$$

We can then solve for the Ridge regression solution using a similar technique as the OLS solution. The solutions are as such:

$$\hat{\beta}_{ridge} = (X^TX + \lambda\mathbb{1})^{-1}X^TY \tag{1.17}$$

which again follows from calculus and the differentiability of the $L2$ norm we're imposing [9][14, Equations (69) and (81)]. The solutions to the Ridge penalty actually follows from solving an equation similar to that of the Normal Equations:

$$\text{(Ridge): } X^TY = X^TX\beta + \lambda\beta \tag{1.18}$$

By utilizing this above equation and factoring we can see the above equation is $X^TY = (X^TX + \lambda\mathbb{1})\beta$. The penalty value λ is a positive number, and this is what allows a Ridge Regression solution to be calculated under circumstances where X^TX is singular unlike the OLS solution. As X^TX is positive semi-definite, and so $v^TX^TXv = (Xv)^T(Xv) = ||Xv||_2^2 \geq 0$, we have that the eigenvalues are always non-negative. We will have X^TX being singular when we have eigenvalues that are zero. By taking the Eigenvalue Decomposition of X^TX, noting it to be symmetric, we have that $X^TX = UVU^T$ where U is an orthogonal matrix whose columns are the eigenvectors of X^TX and V is a diagonal matrix whose entries are the eigenvalues of X^TX. From here we can do the eigendecomposition and find that the λ penalty is actually coercing the eigenvalues to be positive, noting $U\lambda\mathbb{1}U^T = \lambda\mathbb{1}$ by orthgonality:

$$X^TY = (X^TX + \lambda\mathbb{1})\beta = (UVU^T + \lambda\mathbb{1})\beta = (U(V + \lambda\mathbb{1})U^T)\beta \tag{1.19}$$

Penalized Regression and Bias-Variance Tradeoff

Another useful observation about Ridge estimator is that it is a biased estimator as shown below, assuming $\lambda \neq 0$:

$$\mathrm{E}[\hat{\beta}_{ridge}|X] = (X^TX + \lambda\mathbb{1})^{-1}X^T\,\mathrm{E}[Y|X] = (X^TX + \lambda\mathbb{1})^{-1}X^TX\beta$$

It can actually be shown that the Ridge estimator can always achieve a lower MSE than the OLS solution. That is, even while under iid Normal errors of mean zero and the OLS solution being of minimum variance amongst all linear unbiased estimators and thus having the minimum MSE amongst all linear unbiased estimators, as if δ were an estimator then $\mathrm{MSE}(\delta) = \mathrm{Var}(\delta) + \mathrm{Bias}(\delta)^2$, it can be shown that there always exists a λ such that the ridge estimator has lower MSE than the OLS estimator.[3][15] This note introduces the Bias-Variance Tradeoff, and it should be noted that penalized regression methods in general introduce bias in coefficient estimation with the hopes of reducing MSE.

While Ridge regression allows us to construct a regression solution when the OLS solution is not unique, and can actually achieve a lower MSE than the OLS solution, it is a penalty that only constrains the size of the coefficients and does not set them to zero. As such, the final model under a ridge penalty will still contain all predictors and so in situations involving a large number of predictors this can hurt the interpretability of the model, and leads us to the next penalty common penalty term; the LASSO.

1.2.2 LASSO Regression

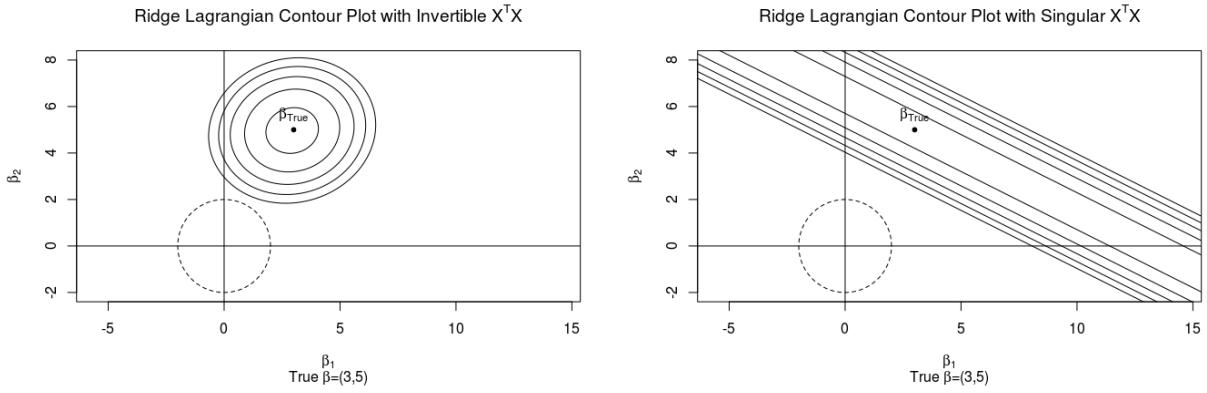
Another way of stating the Ridge penalty is by emphasizing that we are choosing a beta that satisfies minimizing the $L2$ norm of both the residuals and the size of the coefficients, and would be written as such:

$$\hat{\beta}_{ridge} = \arg\min_{\beta} ||Y - X\beta||_2^2 + \lambda||\beta||_2^2 \tag{1.20}$$

The *LASSO*, or Least Absolute Shrinkage and Selection Operator, penalty works similarly but by instead minimizing with an $L1$ penalty on the betas:

$$\hat{\beta}_{LASSO} = \arg\min_{\beta} ||Y - X\beta||_2^2 + \lambda||\beta||_1 \tag{1.21}$$

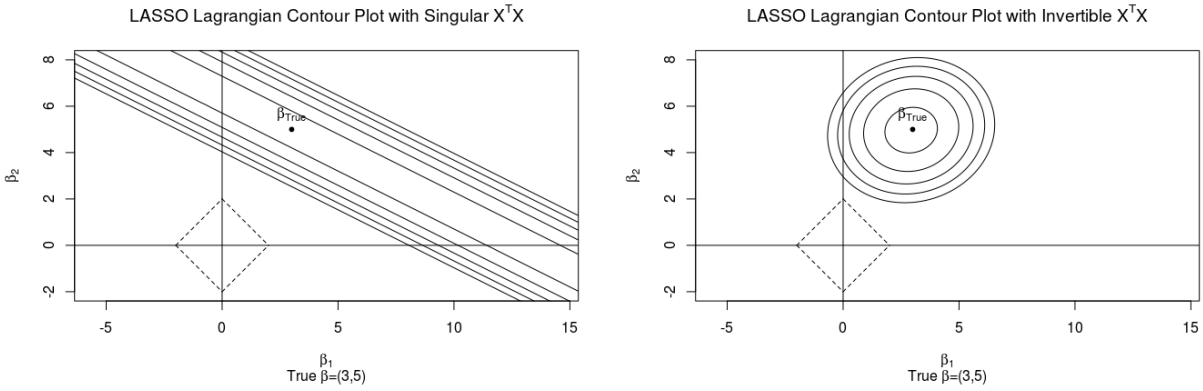
The LASSO method of penalizing does not have a closed form solution like Ridge Regression, and is instead a quadratic programming problem. There are, however, known methods of computing the LASSO solution path with similar efficiency to that of the Ridge regression solution and so the LASSO remains as a popular alternative to Ridge Regression. [9]



(a) Ridge Contour under invertibility

(b) Ridge Contour under singularity

Figure 2: Ridge Regression Penalty Contour Plots



(a) Lasso Contour under singularity

(b) LASSO Contour under invertibility

Figure 3: LASSO Regression Penalty Contour Plots

In the figure above we can see a graphical representation of how utilizing the L_1 norm (LASSO) gives us a constraint region that can set some of the least contributing coefficients on a regression model to zero, whereas this is less probable with Ridge. The figures represent the constraint regions on a two predictor model under X^TX being both invertible and singular. Note that the non-differentiability of the corners, their sharpness, is what is allowing LASSO to find solutions that can reduce the number of covariates by setting their coefficients to zero.

1.2.3 Elastic Net Regression

Elastic Net regression combines the L_1 and L_2 penalties and the coefficient estimates are calculated by the following equation:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(\|Y - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1) \tag{1.22}$$

Most common methods of computing this utilize another hyperparameter α to represent the mix between the norms, while still utilizing λ to control the level of penalization. Under this type of representation our equation would look like the following:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(\|Y - X\beta\|^2 + \lambda(\alpha\|\beta\|^2 + (1-\alpha)\|\beta\|_1)) \tag{1.23}$$

Elastic Net acts as a method of calculating various intermediary convex penalties between the Ridge and LASSO regression penalties, as best demonstrated in the figure below. By combining both methods of penalization it allows us to still favor a sparse model while removing the limitation

on the number of selected variables, and also utilize the Ridge penalty's ability to group highly correlated parameters and shrink their coefficients.

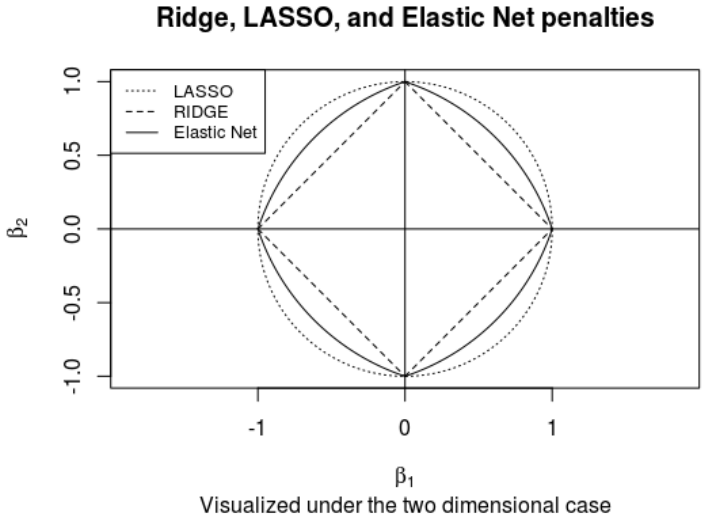


Figure 4: Example of a Lasso, Ridge, and Elastic Net ($\alpha = 0.5$) penalty

1.3 Coordinate Descent

Coordinate descent is an optimization algorithm where given some k-dimensional function f we select a single parameter, fixing all other $k-1$ parameters in the function, and then minimize f. We then continue doing this one by one with each parameter until we either reach convergence or a maximum number of iterations is made.

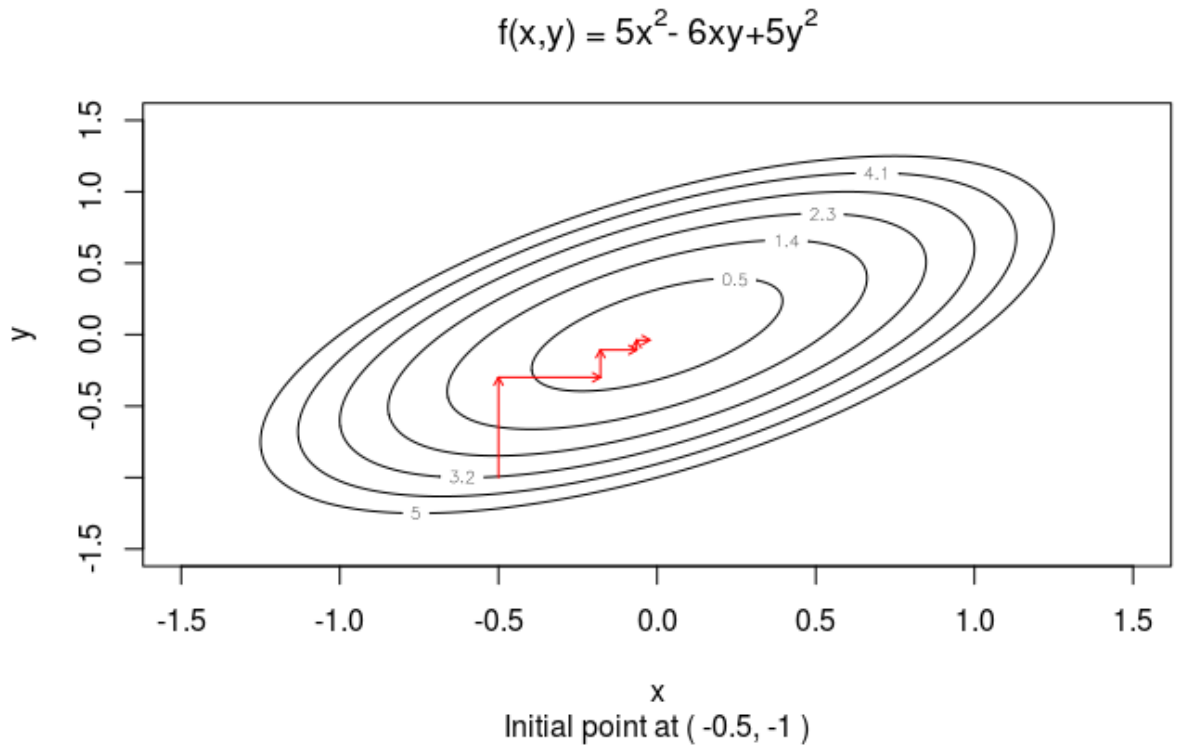


Figure 5: Example of a Coordinate Descent Path

The above figure represents a two-dimensional polynomial function f showing the individual paths taken to reach convergence. Coordinate Descent as a method can offer poor performance when the parameters are highly correlated, and so alternative methods can be preferred. [17] While computationally thrifty, Coordinate Descent is not guaranteed to converge at the minimal value. The R Packages GLMNET and SPARSENET both utilize Coordinate Descent to compute their models for LASSO, Ridge, and non-convex penalties. [5][13]

1.4 LpLq

1.4.1 Generalized Krylov Subspace Methods

The algorithm for the model fitting method ℓ_p-ℓ_q utilizes a *Generalized Krylov Subspace method* for computational efficiency. In linear algebra the order-r *Krylov Subspace* generated by a matrix $A \in \mathbb{R}^{n\times n}$ and a vector $b \in \mathbb{R}^n$ is the linear subspace spanned by the images of b under the first r powers of A starting from $A^0 = \mathbb{1}$. More simply written:

$$K_r(A, b) = span\{b, Ab, A^2b, \ldots, A^{r-1}b\} \tag{1.24}$$

Methods that use Krylov Subspaces are referred to as *Krylov Subspace Methods*, and a Generalized Krylov Subspace Method is if it uses a subspace generated by more than one matrix.[1][10]

1.4.2 LpLq

The model fitting method of ℓ_p-ℓ_q is that of varying the norms used on the residuals penalty and the regularization penalty:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \|Y - X\beta\|_p^p + \lambda\|\beta\|_q^q \tag{1.25}$$

When $q = 2$ we have the Ridge penalty, when $q = 1$ we have the LASSO penalty, and when $q \in (1, 2)$ we have something similar to the Elastic Net penalty where a compromise penalty between Ridge and LASSO is formed. However, when $q > 1$ the penalty function $||\beta||^q$ is differentiable at the corners and does not share the sparsity property of LASSO. This is in contrast to the Elastic Net penalties which can visually appear similar, but are non-differentiable at the corners. For the purposes of this paper we examined the non-convex penalities which occur when $q < 1$, as these can more heavily favor sparsity, and reduce shrinkage, than convex penalties like LASSO, Ridge, and Elastic Net.

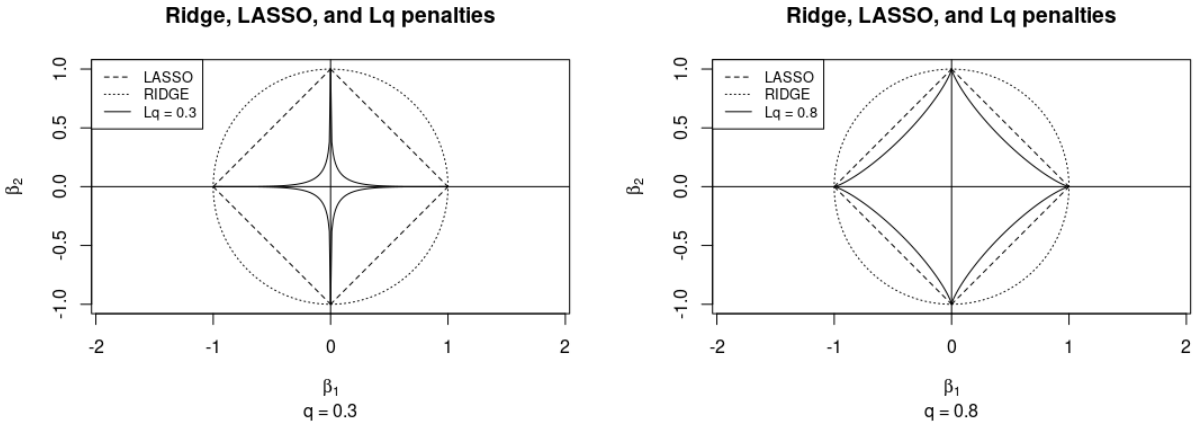


Figure 6: Convex vs Non-convex penalties with ℓ_p-ℓ_q

As previously discussed when $p = 2$ we have several nice properties under normality. However, if this assumption were to be violated then varying the ℓ_p norm could be more optimal and can be done utilizing the ℓ_p-ℓ_q algorithm. In the case of our simulation study since the data is Normal we focus on a fixed $p = 2$.

The ℓ_p-ℓ_q algorithm is an *Adaptive MM Algorithm*. An *MM algorithm* operates by creating a surrogate function that minorizes or majorizes the objective function. When the surrogate function is optimized, the objective function is driven uphill or downhill as needed. In the case of ℓ_p-ℓ_q it utilizes a majorizer to seek minimums, with the majorizer adapted in size to hopefully avoid skipping any possible steeper descents. Another more common example of a MM Algorithm would be the EM (Expectation-Maximization) Algorithm.

For values of $0 \leq q < 1$ the non-convexity of the penalty results in combinatorial computational complexity, and non-convex optimization itself is at least NP-hard. In the simpler case of $q = 0$ the solution can only be solved exactly when the number of covariates are, approximately, at most forty.[8]

1.5 Sparsenet

Sparsenet is an algorithm for computing regression models with non-convex penalties, similar to that of ℓ_p-ℓ_q with $p = 2$ and $q < 1$. The motivation for these non-convex penalties comes from scenarios where the LASSO fails as a variable selector. In these situations to acheive the full effect of a relevant variable the λ penalty must be weakened to the point of allowing other redundant but possibly correlated variables into the model. A predominant difference from Sparsenet to ℓ_p-ℓ_q is that Sparsenet utilizes Coordinate Descent to compute the solution path, and has the residual penalty norm fixed as the L_2-norm.

CHAPTER 2

Methodology

The simulation data was generated following the correlation structures used by the paper *Sparsenet: Coordinate Descent with Non-Convex penalties* by Mazumder, Friedman and Hastie.[13] There are 5 different simulations utilized to make comparisons between the computational algorithms Sparsenet, LpLq, and GLMNET with alpha = {0, 1, 0.5} which corresponds to Ridge, LASSO, and an example of a Elastic Net regression respectively. Each of these algorithms were used as implementations of R packages, with the Sparsenet and GLMNET packages being hosted on the CRAN R Repository.

Under each simulation a training sample was generated that was 10-times larger than the size of the testing sample. For each tuning parameter we had utilized k-fold cross-validation with k = 10 to allow the individual algorithms to find optimal tuning parameters according to a minimum Mean Squared Prediction Error averaged across the 10 folds, noting the method of assigning the folds were uniform across each algorithm. The choice of k could be varied from $2 \leq k \leq n$, but the value of k = 10 is a common recommendation by heuristic experts. [9] Once this optimal parameter was found each model is fit with these parameters to the entire validation set to generate their corresponding coefficient values, noting the number of non-zero coefficients, and how many coefficient were correctly classified as being zero or non-zero. The mean squared error with the training set was then computed using these coefficients.

2.0.1 Data Generation

Generating the data followed techniques from a Multivariate Normal Distribution using the affine transformation technique. We look to generate the data under the following normality structure, being given some Variance-Covariance matrix Σ , m-coefficients β, the number of n-samples, and a standard deviation σ:

$$Y_{n\times 1} \sim X_{n\times m}\beta_{m\times 1} + \epsilon_{n\times 1} \text{ such that } X_{n\times m} \sim \mathrm{N}_m(0_{m\times 1}, \Sigma_{m\times m}) \text{ and } \epsilon \sim \mathrm{N}(0, \sigma^2 \mathbb{1}) \tag{2.1}$$

We are utilizing a Signal-to-Noise Ratio (SNR) of 3 in each simulation where:

$$(\text{SNR}) \ = \frac{\sqrt{\beta^T \Sigma \beta}}{\sigma} \Rightarrow \sigma = \frac{\sqrt{\beta^T \Sigma \beta}}{\text{SNR}}; \ \text{SNR} = 3 \tag{2.2}$$

Utilizing SNR = 3 provides us with the standard deviation, σ, used the generate our error terms ϵ once we are given β, n, and Σ.

To start generating the data we create $X \sim N(0, \mathbb{1})$ by generating $n \times m$ independent samples from the standard Normal distribution and form them into the $X_{n \times m}$ matrix. To then give this matrix the desired variance-covariance structure we take the given Σ and undergo an eigenvalue decomposition of the matrix. Note that Σ, and all variance-covariance matrices by definition, are symmetric and so $\Sigma = UVU^T$ where U is an orthogonal matrix of eigenvectors so $UU^T = \mathbb{1}$, and V is a diagonal matrix of eigenvalues.

Given that the variance-covariance matrix is always positive semi-definite, and so the eigenvalues are non-negative, we can assign $B = U\sqrt{V}$ noting $BB^T = \Sigma$ and $\sqrt{V}\sqrt{V} = V$. The affine transformation then works by utilizing the following:

$$\text{Given } X \sim \text{N}(\mu, \Sigma) \text{ and } Y \sim c + BX$$

$$\text{Then } Y \sim \text{N}(c + B\mu, B\Sigma B^T)$$

Therefore when $B = U\sqrt{V}$ and $Z \sim \text{N}(0, \mathbb{1})$ we have $X = BZ \sim \text{N}(0, B\mathbb{1}B^T = \Sigma)$. It is worth noting that the two equations above are generating column vectors Y, X, but in a regression setup we utilize row vectors with each row corresponding to a sample when referring to our matrix X. This means that our actual algebra for the calculation works as $X_{n \times m} = Z_{n \times m} B^T$ to have the correct result of a matrix of n-samples with the given distribution $X \sim \text{N}(0, \Sigma)$ when $Z \sim \text{N}(0, \mathbb{1})$. To then generate Y we then computed the following $Y_{n \times 1} = X_{n \times m}\beta_{m \times 1} + \epsilon_{n \times 1}$ where ϵ is n-samples from a $\text{N}(0, \sigma)$ distribution.

2.0.2 Simulation Study

There are then six distinct simulations utilized, whose parameters are listed below. Note the use of the notation $\Sigma(\rho; m)$ which denotes a $m \times m$ matrix with 1's on the diagonal, and ρ's on the

off-diagonal.

S1: $n = 35, \quad p = 30, \quad \Sigma^{S1} = \Sigma(0.4; p)$ and $\beta^{S1} = (0.03, 0.07, 0.1, 0.9, 0.93, 0.97, \mathbf{0}_{1\times 24})$

M1: $n = 100, \quad p = 200, \quad \Sigma^{M1} = \{0.7^{|i-j|}\}_{1\leq i,\ j\leq p}$ and β^{M1} has 10 non-zeros such that $\beta^{M1}_{20i+1} = 1, i = 0, 1, \ldots, 9;$ and $\beta^{M1}_i = 0$ otherwise.

M1.5: $n = 500, \quad p = 1000, \quad \Sigma^{M1.5} = \text{blockkdiag}(\Sigma^{M1}, \ldots, \Sigma^{M1})$ and $\beta^{M1.5} = (\beta^{M1}, \ldots, \beta^{M1})$ (five blocks)

M1.10: $n = 500, \quad p = 2000$ (and is like M1.5 but with ten blocks instead of five)

M2.5: $n = 500, \quad p = 1000, \quad \Sigma^{M2.5} = \text{blockdiag}(\Sigma(0.5, 200), \ldots, \Sigma(0.5, 200))$ and $\beta^{M2.5} = (\beta^{M2}, \ldots, \beta^{M2})$ (five blocks). Here $\beta^{M2} = (\beta_1, \ldots, \beta_{10}, \mathbf{0}_{1\times 190})$ is such that the first ten coefficients form an equi-spaced grid on $[0, 0.5]$.

M2.10: $n = 500, \quad p = 2000,$ and is like M2.5 but with ten blocks instead of five.

CHAPTER 3

Results

Table 1: S1 Results

Model	# of Nonzero (6)	MSPE	# of False Zeros	# of Missed Zeros
ridge	30	0.7453	0	24
elasticnet	15	0.6596	0	9
lasso	15	0.6526	0	9
lplq	11	0.6463	0	5
sparsenet	13	0.6302	0	7

Table 2: M1 Results

Model	# of Nonzero (10)	MSPE	# of False Zeros	# of Missed Zeros
ridge	200	1.7391	0	190
elasticnet	50	1.3703	0	40
lasso	22	1.3447	0	12
lplq	10	1.2418	0	0
sparsenet	10	1.2375	0	0

Table 3: M1.5 Results

Model	# of Nonzero (50)	MSPE	# of False Zeros	# of Missed Zeros
ridge	1000	7.4046	0	950
elasticnet	184	6.1731	0	134
lasso	156	6.1406	0	106
lplq	50	6.0298	0	0
sparsenet	50	6.0326	0	0

Table 4: M1.10 Results

Model	# of Nonzero (100)	MSPE	# of False Zeros	# of Missed Zeros
ridge	2000	17.2603	0	1900
elasticnet	409	13.3988	0	309
lasso	366	13.3215	0	266
lplq	124	12.6251	0	24
sparsenet	100	12.6791	0	0

Table 5: M2.5 Results

Model	# of Nonzero (45)	MSPE	# of False Zeros	# of Missed Zeros
ridge	1000	2.3221	0	955
elasticnet	168	1.9323	1	124
lasso	154	1.9283	1	110
lplq	42	1.8940	5	2
sparsenet	40	1.8809	5	0

Table 6: M2.10 Results

Model	# of Nonzero (90)	MSPE	# of False Zeros	# of Missed Zeros
ridge	2000	4.9749	0	1910
elasticnet	320	3.9650	5	235
lasso	311	3.9557	5	226
lplq	77	3.9661	15	2
sparsenet	74	3.9603	16	0

Table 7: Number of Penalty values (λ)

	ridge	elasticnet	lasso	lplq	sparsenet
S1	100	74	73	10	100
M1	100	83	83	10	100
M1.5	100	82	83	10	100
M1.10	100	89	86	10	100
M2.5	100	88	88	10	100
M2.10	100	92	92	10	100

Table 8: Number of Non-convex Family Parameters (q and γ)

	ℓ_p-ℓ_q	Sparsenet
S1	11	74
M1	11	83
M1.5	11	82
M1.10	11	89
M2.5	11	88
M2.10	11	92

CHAPTER 4

Discussion

Starting with the first simulation S1 we find that ℓ_p-ℓ_q had computed the sparsest model, with the fewest misclassified zero coefficients, along with having the second smallest MSPE and overall achieves the best performance in this scenario. With M1 and M1.5 we find a relative tie in the performance of ℓ_p-ℓ_q and Sparsenet, both of which are found to outperform the classical penalties, although Sparsenet does achieve a smaller MSPE than ℓ_p-ℓ_q in both instances. With M2.10 despite LASSO achieving the lowest MSPE the convex penalties of ℓ_p-ℓ_q and Sparsenet achieve much sparser models with vastly better coefficient classification rates. An interesting result with M2.10 is that ℓ_p-ℓ_q is actually the closest in having the correct number of non-zero terms, albeit this is the result of 17 misclassified coefficients compared to Sparsenets 16. Both non-convex penalties achieve a very parsimonious model under M2.10 compared to the more common penalties and with similar MSPE's, all of this despite LASSO actually have the lowest MSPE in this simulation. Similar results are shown for M2.5 as with M2.10 but the non-convex penalties do achieve a lower MSPE here with ℓ_p-ℓ_q again being the second lowest MSPE. With M2.5 we again find ℓ_p-ℓ_q to be the closest to the correct number of non-zero coefficients, though with having 7 misclassified coefficients compared to Sparsenet's 5 miscalssified coefficients. Arguably the worst performance by ℓ_p-ℓ_q regarding coefficient classification was with M1.10, although it does achieve the lowest MSPE out of all methods presented in this situation. In this M2.10 simulation Sparsenet had correctly classified all of the coefficients whereas ℓ_p-ℓ_q incorrectly classified 24 true-zero coefficients as non-zero.

The biggest hurdles with this simulation study were the computing resources needed to fit the models for ℓ_p-ℓ_q. Various techniques are still being developed to further the algorithms computational efficiency and there had been several revisions to the algorithm since the start of our work. Due to these constraints the number of models fit under the ℓ_p-ℓ_q algorithm are at a much

smaller magnitude compared to the other methods of penalized regression, as seen in Tables 7 and 8. Furthermore, the algorithms utilized in GLMNET and SPARSENET calculate the sequence of parameters in a much more finely tuned and better studied method than the sequence of parameters for the solution path of ℓ_p-ℓ_q, which was unknown at the time of this study. Despite these shortcomings there are some promising results shown as while ℓ_p-ℓ_q was constrained by computing power it still remained a viable contender in each simulation against Sparsenet where both were generally vastly outperforming the LASSO and other convex penalty regression models with regards to parsimony.

BIBLIOGRAPHY

[1] A. Buccini and L. Reichel. An $\ell_2-\ell_q$ regularization method for large discrete ill-posed problems. *J. Sci. Comput.*, 78(3):1526–1549, Mar. 2019.

[2] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 04 2004.

[3] R. W. Farebrother. Further results on the mean square error of ridge regression. *Journal of the Royal Statistical Society. Series B (Methodological)*, 38(3):248–250, 1976.

[4] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[5] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[6] W. H. Greene. *Econometric Analysis*. Prentice Hall, 2002.

[7] D. A. Harville. *Matrix Algebra From a Statistician's Perspective*. Springer, 2000.

[8] T. Hastie. *Statistical learning with Sparsity : the lasso and generalizations*. CRC Press LLC, Boca Raton, 2015.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, 2009.

[10] G. Huang, A. Lanza, S. Morigi, L. Reichel, and F. Sgallari. Majorization-minimization generalized krylov subspace methods for $\ell_p - \ell_q$ optimization applied to image restoration. *BIT Numerical Mathematics*, 57, 01 2017.

[11] A. Lanza, S. Morigi, L. Reichel, and F. Sgallari. A generalized krylov subspace method for p-q minimization. *SIAM J. Scientific Computing*, 37, 2015.

[12] J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics, 2nd Edition*. Wiley, 1999.

[13] R. Mazumder, J. Friedman, and T. Hastie. Sparsenet: Coordinate descent with nonconvex penalties. *Journal of American Statistical Association*, 106(495):1125–1138, 2011.

[14] K. B. Petersen and M. S. Pedersen. The matrix cookbook, nov 2012. Version 20121115.

[15] C. M. Theobald. Generalizations of mean square error applied to ridge regression. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(1):103–106, 1974.

[16] R. Tibshirani. Regression shrinkage and selection via the lasso. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 58:267–288, 1994.

[17] R. J. Tibshirani. Dykstras algorithm, admm, and coordinate descent: Connections, insights, and extensions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 517–528. Curran Associates, Inc., 2017.

APPENDIX A

Proof of Variance-Covariance matrix being positive semi-definite.

By definition we want $a^t \Sigma a \geq 0$ for all $a \neq \vec{0}$. As Σ is a symmetric matrix by definition we know $a^T \Sigma a$ is in quadratic form and thus:

$$a^t \Sigma a = \sum_i \sum_j \sigma_{ij} a_i a_j$$

$$\text{Where } \sigma_{ij} = \mathrm{E}[Z_i Z_j], \quad Z_i = X_i - \mathrm{E}[X_i]$$

Therefore,

$$\sum_i \sum_j \sigma_{ij} a_i a_j = \mathrm{E}\left[\sum_i \sum_j a_i Z_i Z_j a_j \right] \text{ by Linearity}$$

$$= \mathrm{E}[(\sum_i a_i Z_i)^2] \text{ by Induction}$$

As $(\sum_i a_i Z_i)^2 \geq 0 \Rightarrow \mathrm{E}[(\sum_i a_i Z_i)^2] = a^T \Sigma a \geq 0 \ \square$

APPENDIX B

MSE(Estimator) = Variance(Estimator) + Bias(Estimator)^2

$$
\begin{aligned}
\mathrm{MSE}(\hat{\theta}) &= \mathrm{E}_{\hat{\theta}}\left[(\hat{\theta}-\theta)^2\right] \\
&= \mathrm{E}\left[\left(\hat{\theta}-\mathrm{E}[\hat{\theta}]+\mathrm{E}[\hat{\theta}]-\theta\right)^2\right] && \text{Add and subtract } \mathrm{E}[\hat{\theta}] \\
&= \mathrm{E}\left[\left(\hat{\theta}-\mathrm{E}[\hat{\theta}]\right)^2+2\left(\hat{\theta}-\mathrm{E}[\hat{\theta}]\right)\left(\mathrm{E}[\hat{\theta}]-\theta\right)+\left(\mathrm{E}[\hat{\theta}]-\theta\right)^2\right] && \text{Group and expand} \\
&= \mathrm{E}\left[\left(\hat{\theta}-\mathrm{E}[\hat{\theta}]\right)^2\right]+\mathrm{E}\left[2\left(\hat{\theta}-\mathrm{E}[\hat{\theta}]\right)\left(\mathrm{E}[\hat{\theta}]-\theta\right)\right]+\mathrm{E}\left[\left(\mathrm{E}[\hat{\theta}]-\theta\right)^2\right] \\
&= \mathrm{E}\left[\left(\hat{\theta}-\mathrm{E}[\hat{\theta}]\right)^2\right]+2\left(\mathrm{E}[\hat{\theta}]-\theta\right)\mathrm{E}\left[\hat{\theta}-\mathrm{E}[\hat{\theta}]\right]+\left(\mathrm{E}[\hat{\theta}]-\theta\right)^2 && \mathrm{E}[\hat{\theta}]-\theta = \text{constant} \\
&= \mathrm{E}\left[\left(\hat{\theta}-\mathrm{E}[\hat{\theta}]\right)^2\right]+2\left(\mathrm{E}[\hat{\theta}]-\theta\right)\left(\mathrm{E}[\hat{\theta}]-\mathrm{E}[\hat{\theta}]\right)+\left(\mathrm{E}[\hat{\theta}]-\theta\right)^2 && \mathrm{E}[\hat{\theta}] = \text{constant} \\
&= \mathrm{E}\left[\left(\hat{\theta}-\mathrm{E}[\hat{\theta}]\right)^2\right]+\left(\mathrm{E}[\hat{\theta}]-\theta\right)^2 \\
&= \mathrm{Var}(\hat{\theta})+\mathrm{Bias}(\hat{\theta})^2
\end{aligned}
$$

APPENDIX C

MSE(prederr) = Variance(prederr)+Bias(prederr)^2+σ^2

Let $y = f(x;\beta) + \epsilon$ where $\mathrm{E}[\epsilon] = 0$, $\mathrm{Var}[\epsilon] = \sigma^2$, ϵ is independent of X, and $\mathrm{Var}[y] = \sigma^2$.

$$\begin{aligned}
\mathrm{MSE} = \mathrm{E}\left[(y-\hat{f})^2\right] &= \mathrm{E}\left[(f+\varepsilon-\hat{f})^2\right] \\
&= \mathrm{E}\left[(f+\varepsilon-\hat{f}+\mathrm{E}[\hat{f}]-\mathrm{E}[\hat{f}])^2\right] \\
&= \mathrm{E}\left[(f-\mathrm{E}[\hat{f}])^2\right] + \mathrm{E}[\varepsilon^2] + \mathrm{E}\left[(\mathrm{E}[\hat{f}]-\hat{f})^2\right] + 2\,\mathrm{E}\left[(f-\mathrm{E}[\hat{f}])\varepsilon\right] + 2\,\mathrm{E}\left[\varepsilon(\mathrm{E}[\hat{f}]-\hat{f})\right] \\
&\quad + 2\,\mathrm{E}\left[(\mathrm{E}[\hat{f}]-\hat{f})(f-\mathrm{E}[\hat{f}])\right] \\
&= (f-\mathrm{E}[\hat{f}])^2 + \mathrm{E}[\varepsilon^2] + \mathrm{E}\left[(\mathrm{E}[\hat{f}]-\hat{f})^2\right] + 2(f-\mathrm{E}[\hat{f}])\,\mathrm{E}[\varepsilon] \\
&\quad + 2\,\mathrm{E}[\varepsilon]\,\mathrm{E}\left[\mathrm{E}[\hat{f}]-\hat{f}\right] + 2\,\mathrm{E}\left[\mathrm{E}[\hat{f}]-\hat{f}\right](f-\mathrm{E}[\hat{f}]) \\
&= (f-\mathrm{E}[\hat{f}])^2 + \mathrm{E}[\varepsilon^2] + \mathrm{E}\left[(\mathrm{E}[\hat{f}]-\hat{f})^2\right] \\
&= (f-\mathrm{E}[\hat{f}])^2 + \mathrm{Var}[y] + \mathrm{Var}\left[\hat{f}\right] \\
&= \mathrm{Bias}[\hat{f}]^2 + \mathrm{Var}[y] + \mathrm{Var}\left[\hat{f}\right] \\
&= \mathrm{Bias}[\hat{f}]^2 + \sigma^2 + \mathrm{Var}\left[\hat{f}\right]
\end{aligned}$$