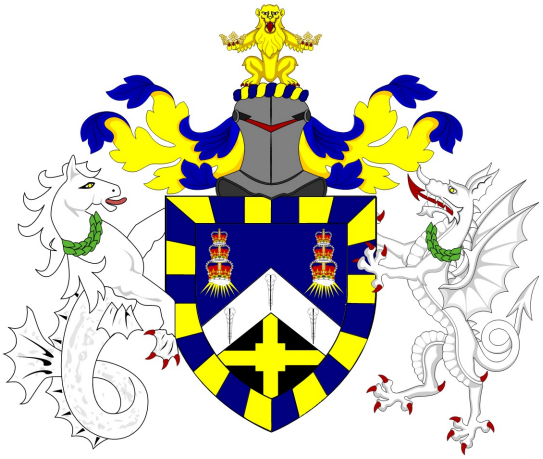
Mathematics MSc Dissertation MTHM038, 2019/20

Lasso

A study of Lasso Regression algorithm

Stuti Malik, ID 150047032

Supervisor: Prof. Dr Hugo Maruri-Aguilar



A thesis presented for the degree of

Master of Science in *Mathematics*

School of Mathematical Sciences

Queen Mary University of London

Declaration of original work

This declaration is made on November 16, 2020.

Student's Declaration: I Stuti Malik hereby declare that the work in this thesis is my original work. I have not copied from any other students' work, work of mine submitted elsewhere, or from any other sources except where due reference or acknowledgement is made explicitly in the text, nor has any part been written for me by another person.

Abstract

The least absolute shrinkage and selection operator, Lasso, simplifies the regression model by shrinking the insignificant coefficients to zero and thus making the model more interpretable, by selecting only the most relevant coefficients. It is very useful for a complicated model with large number of coefficients, as too many features in a model can cause over-fitting, affecting the predicting power of the model. Lasso methodology improves the prediction accuracy of a model by avoiding over-fitting. We will see how such an optimal model is chosen and study the lasso steps, and gain an understanding of how lasso method is performed and used to improve the model prediction.

We will gain an understanding of the calculations involved in the lasso process, understand the methodology and try to formulate a collection of algorithms that helps us to perform the lasso steps from scratch.

An applied example of performing the lasso algorithm steps on a synthetic data-set is discussed, showing all the intermediate steps in detail, to reproduce the results, as given by using the `lars` package in R.

Contents

Chapter 1

Introduction

1.1 Motivation for this work

Lasso is a methodology which shrinks the coefficients in a regression model so that only the most significant coefficients are retained in the model. It simplifies the regression model and also improves the prediction quality by reducing the variance of the predictions and making the fit more general and less reliant on the given data, so that the model fits the unseen data better.
We will build upon the Regression Shrinkage and Selection via Lasso, by Robert Tibshirirani, 1994 paper's Lasso approach [12] and provide an explanation and analysis of the process and build an explanation of the algorithm which performs the Lasso steps. The aim of this project is to provide the alternate explanation of the process and understand the lasso operations more deeply.

1.2 Regression analysis

Consider a simple linear regression model:

$$Y = X\beta + \epsilon$$

Where ϵ is the error term.
The expected value of the response variable, Y, is

$$\hat{Y} = X\hat{\beta}$$

The sum of squared errors, which we aim to minimise, in a simple linear regression is:

$$\epsilon = \sum_{i=1}^{N}(y_i - \alpha - \sum_j \beta_j x_{ij})^2$$

Where α is the constant or intercept, and β_j are the coefficients or slope in the linear regression. This equation is the sum of squared error terms, which is the actual value of the response variable, y, minus its predicted value, $\hat{y}$, as given by linear regression model, $\hat{\beta}X$.

The least squares regression equation, which aims to minimise the sum of error squares is written as:

$$\arg\min_{\beta} \sum_{i=1}^{N}(y_i - \alpha - \sum_j \beta_j x_{ij})^2$$

The OLS (ordinary least squares) estimate for the coefficients can easily be shown to be $\beta = (X^T X)^{-1} X^T Y$, by solving the optimisation equation above, for the $X^T X$ being a non-singular matrix. Some useful explanations of the derivation of the least square solution are covered in the notes [10] and the book [15] (chapter 3, page 45), which uses the concept of minimising the root mean square error to find the optimal parameters.

1.2.1 The problem

The problem with the least squares estimate method is that it depends too much on the data, and fits a model which is a good fit just for the given data-set, however, it might not perform as well on a new independent data which we have not seen. This is especially the case, and a more apparent problem when fitting a model on a small data-set, or the one with a lot of dependent variables, and having correlations among the predictors, as some variables might not be as relevant or significant to affect the predictions, and other variables might have a stronger influence on the response variable.

1.2.2 The lasso approach

In the LASSO methodology (least absolute shrinkage and selection operator), we add a L1 norm penalty (the L1 norm of the model parameter vector, β, is the sum of absolute values

of all the parameters, $|\beta|_1 = \sum_j |\beta_j|$), written as $|\beta|_1 \leq t$, to the least squares equation, as a regularisation constrain, that is, the L1 norm of the parameters being less than a constant, so that some variables shrink to zero and we only include the most relevant parameters, in the model in order to reduce the prediction error. This added constraint cause the shrinkage of the insignificant features/parameters/coefficients to zero. This keeps the model simple and it detects and prevents the over-fitting problem, thus improving the prediction ability.

In lasso methodology, we change the optimisation equation for model coefficients, $\hat{\beta}$, in order to minimise the error terms from

$$\hat{\beta} = \arg\min_{\beta}\{\frac{1}{2}||Y - X\beta||_2^2\}$$

to

$$\hat{\beta} = \arg\min_{\beta}\{\frac{1}{2}||Y - X\beta||_2^2\} : |\beta|_1 \leq t$$

or

$$\hat{\beta} = \arg\min_{\beta}\{\frac{1}{2}||Y - X\beta||_2^2 + \lambda|\beta|_1\}$$

That is, adding a linear constraint on the quadratic optimisation problem.

Considering $\hat{\beta} = \arg\min_{\beta}\{\frac{1}{2}||Y - X\beta||_2^2\} : |\beta|_1 \leq t$, the tuning parameter, t, has to be positive and not more than the ordinary least squares estimate of β, $|\beta^{OLS}|_1$, as otherwise the condition $|\beta^{OLS}|_1 \leq t$ is already satisfied. If $t < |\beta^{OLS}|_1$, the parameters perform shrinkage to satisfy the condition, and some parameters are shrunk to exactly zero due to the nature of the constraint. Therefore, the condition on t is $0 \leq t \leq |\beta^{OLS}|_1$.
Adding the constraint $|\beta|_1 \leq t$, where t is the regularisation parameter, ensures that the sum of absolute value of coefficients is constrained and consequently, some coefficients shrink to zero and others to a smaller value. Later in the chapter, we will see how to select the regularisation parameter, t, to perform the model selection, which has a better prediction quality than the least squares method and has the optimal selection of model coefficients, making the model simpler and more easy to interpret.
Now considering the different version of the lasso regression model, $\hat{\beta} = \arg\min_{\beta}\{\frac{1}{2}||Y - X\beta||_2^2 + \lambda|\beta|_1\}$, and comparing the conditions on t with λ, we see that the $\lambda = 0$, corresponds to when $t = |\beta^{OLS}|_1$, as in this case, there is no shrinkage applied and we just have the simple linear regression model, with ordinary least squares estimate,$\hat{\beta} = |\beta^{OLS}|_1$. The condition on λ is $0 \leq \lambda \leq ?$, which we need to find. Note that as t decreases, λ increases, applying more shrinkage to the model. As an interesting exercise, we need to find what

value of λ (in equation 2.2) does $t = 0$ (in equation 2.1) corresponds to. Note that $t = 0$ leads to complete shrinkage, that is all coefficients shrink to zero, as the condition $|\beta|_1 \leq 0$ is satisfied only if $|\beta|_1 = 0$ and all coefficients, $\beta_1, \beta_2, ..., \beta_n$, are zero. The value of λ where all the coefficients shrink to zero is different for every model and data. We need to do the lasso path analysis to find that, and to find the optimal λ which suggests which coefficients to pick and their values, giving a model with a lower L1 norm of coefficients, to prevent over-fitting and improve the prediction accuracy.

The aim of lasso regression is to find the optimal model coefficients, $\hat{\beta}$, such that it optimises the quadratic least squares equation, $|||Y - X\beta||_2^2$, which aims to minimise the squared difference between the predicted values and the actual values of the response variable, Y, subject to the linear constraint on the absolute values of coefficients, $|\beta|_1 \leq t$.

1.2.3 A typical linear model

After finding the optimal model parameters, using one of the methods such as least squares regression, ridge regression, lasso regression, etc., we use them to make predictions on a new data-set.

The predicted or expected value of the response variable, $\hat{Y} = \begin{pmatrix} \hat{y_1} \\ \hat{y_2} \\ . \\ . \\ \hat{y_n} \end{pmatrix} = X\hat{B}$, where X is the data matrix and $\hat{\beta}$ is the vector of optimal parameters selected.

$X = \begin{pmatrix} 1 & x_{11} & x_{12} & . & . & . & x_{1p} \\ 1 & x_{21} & x_{22} & . & . & . & x_{2p} \\ . & & & & & & \\ . & & & & & & \\ . & & & & & & \\ 1 & x_{n1} & x_{n2} & . & . & . & x_{np} \end{pmatrix}$ and $\hat{B} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ . \\ . \\ . \\ \hat{\beta}_p \end{pmatrix}$ if the intercept is non-zero, and

$$X = \begin{pmatrix} x_{11} & x_{12} & . & . & . & x_{1p} \\ x_{21} & x_{22} & . & . & . & x_{2p} \\ . & & & & & \\ . & & & & & \\ . & & & & & \\ x_{n1} & x_{n2} & . & . & . & x_{np} \end{pmatrix} \text{ and } \hat{B} = \begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ . \\ . \\ . \\ \hat{\beta}_p \end{pmatrix} \text{ if we set the intercept to be zero.}$$

There is more about different variations of the model in the appendix section, with an example showing different lasso paths for normalised and unnormalised data.

1.2.4 Judging a model performance

We usually start by splitting the data-set into a training and testing set, and fit our model only using the data in the training set. We judge the model performance by its root mean squared error or mean squared error fit on the test set. We prefer that the root mean square or the mean square prediction error is similar on both training and test set, for model to have a good generalisation ability. The sign of poor generalisation ability is when the mean squared error on the test set is much larger than on the training set, and this indicates that the model is an over-fit. We can avoid problem of over-fitting, which is detected by validating a model on test set, by keeping the model simple, hence using regularisation. We also want the model to make good predictions on both the training and test data-set. We seek a most simple model, with lowest parameters possible, to have a reasonably high prediction quality, as too many parameters can lead to an over-fit and larger error on the test set. As we increase the number of parameters, the training set prediction error decreases or stays the same, but the test set prediction error usually decreases at first and then starts to increase again after a certain number of parameters. The model makes poor predictions if the number of parameters is too small or too large, so we need to find the optimal selection of parameters. The lasso method helps find such a model, which is simple and has a good prediction ability.

1.3 Literature Review

Lasso methodology was studied in detail and proposed by Robert Tibshirani, in his original paper [12], and he also contributed in the paper [3] about LARS, least angle regression

methodology, and the paper [8] about lasso for hierarchical interactions, the book [15] about data mining, inference and predictions, and the book [16] about the lasso and generalisations. He also wrote a paper about an extension of lasso to variable selection in the Cox's proportional hazards model [13].

The original lasso paper, [12], proposes a new method for estimation of coefficients in linear regression, and discusses how adding a special constraint on the coefficients gives more interpretable models, showing the geometrical illustration of the constraint. This paper also looks at the limitations of subset and ridge regression regularisation methods, and discusses how lasso enjoys the favourable properties of both. The paper explains the general idea of lasso, and briefly discusses the extensions to generalised regression models and tree-based models. This paper describes the lasso algorithm, which we will try to analyse, build an alternate explanation of the algorithm and show the calculation steps involved.

As discussed in the lasso paper [12], to visualise the lasso model, we can plot $|\beta|_1$, which is a tilted square, to see that it has sharp edges where the parameter completely shrinks to zero when the elliptic curve touches the edge, as opposed to the ridge regression, which has the L2 norm penalty, $||\beta||_2$, giving a circular space plot, where the parameter goes close to zero but does not actually touch it. Thus, the lasso regression model has an advantage of giving more sparse model than ridge regression. It also has an advantage over subset selection, which is a discrete process of dropping or adding variables in the model, as the lasso regression also provides an interpret-able model but have a higher prediction accuracy because in subset regression the small changes in data can give very different models.

The uniqueness of lasso is discussed in the "The Lasso Problem and Uniqueness" paper [14], which discusses the question of when is the lasso solution well defined and unique, and how to manage the case when the lasso solutions are not unique, while making a progress towards understanding some aspects of non-uniqueness in lasso solutions. This paper reviews the unifying properties of lasso solutions, and point out particular forms of solutions that have distinctive properties. The uniqueness of lasso solutions is also discussed in the book [16], chapter 2, page 19.

An useful introduction to Lasso Regression is provided by the "Intro to Lasso Regression" lecture slide [2], which discusses the comparison between ridge and lasso regression, explains the concept of sub-differential and a solution to lasso equation, and also shows an applied example with R code, to do the lasso path analysis.

An useful insight on different regularisation method algorithms is provided by the "Regularization Paths" slide, [7], which compares the different regularisation methods and their paths.

The Lasso Page [11] has the main collection of links for useful research and relevant work

about lasso. The lasso methodology has been extended into more advanced models, such as polynomial regression, logistic regression, hierarchical models, where there is an interaction between predictors and Cox's proportional hazards model [8], and many more.
An useful method for extending Lasso for hierarchical polynomial models is provided by the paper [5], which discusses the development of parameter constraints that enforce hierarchy for hierarchical polynomial models, using the divisibility conditions of model terms in polynomial hierarchy. This paper provides an useful explanation for estimating the parameters in constrained lasso using standard quadratic programming techniques, using the hierarchy developed and a matrix for constrained optimization.

1.4 An overview of the contents

The chapter 2, "Lasso Analysis", covers the proofs of the main equations used in the lasso methodology, in order to gain a deeper understanding of the lasso equations, that are used to fit a model, having the L1 norm constraint. It also covers the how the tuning parameter, λ, is selected in the lasso process, with an example, using cross-validation and interpolation to select the optimal model coefficients of a complicated model with eight parameters.
The chapter 3, "Lasso Algorithm", develops an understanding of the lasso methodology and the steps used in lasso process, from scratch. It suggests a potential algorithm to find the generation of moves between quadrants and step by step process to perform the lasso calculation by hand and detailed understanding of the process, with small examples.
The chapter 4, covers a detailed example of performing the lasso steps by hand, to retrieve all the coefficients at each step in the lasso process and the corresponding lambdas given using the `lars` function in R. It also covers an analysis of the lasso path plot and its relation with the calculations giving model coefficients in different quadrants.
The chapter 5, covers the conclusion and summary of the methods learnt about the selection of optimal model using the lasso methodology, and possible further developments and applications of the process.
Notation used in this document:
Unless stated otherwise, the β refers to the vector of coefficients, the X refers to the data matrix, and the Y refers to the vector of response variable. The $\hat{\beta}$ refers to the optimal model coefficients selected. The intercept is assumed to be zero in most of the proofs and lasso analysis examples, unless stated otherwise. The $|\beta|_1$ refers to the L1 norm of the coefficients, and $\beta_1, \beta_2, ..\beta_p$ refers to the individual coefficients, in a p-parameter model.

Chapter 2

Lasso Analysis

2.1 Two forms of defining the lasso estimate

As an introduction to lasso equations, we will show that the two forms of writing lasso estimate for the model coefficients, $\hat{\beta}$, are equivalent:

The equation 2.1 and 2.2 are equivalent

$$\hat{\beta} = \arg\min_{\beta}\{\frac{1}{2}||Y - X\beta||_2^2\} : |\beta|_1 \leq t \tag{2.1}$$

$$\hat{\beta} = \arg\min_{\beta}\{\frac{1}{2}||Y - X\beta||_2^2 + \lambda|\beta|_1\} \tag{2.2}$$

For the proof, consider the method of Lagrange multipliers.
For more details on Lagrange multipliers, [4] has a a very good explanation of why it works.

Let us start with the equation 2.1, $\arg\min_{\beta}\{\frac{1}{2}||Y - X\beta||_2^2\}$ subject to $|\beta|_1 \leq t$. We can ignore the factor of positive constant, $\frac{1}{2}$, in minimising $||Y - X\beta||_2^2$ with respect to β and write $\arg\min_{\beta}\{||Y - X\beta||_2^2\}$ subject to $|\beta|_1 \leq t$.

According to Lagrange multipliers, at the boundary point, $|\beta|_1 = t$, the gradients of $||Y - X\beta||_2^2$ and $|\beta|_1$ are parallel, and can be writte as:

$$\frac{\partial||Y - X\beta||_2^2}{\partial\beta} = \lambda_1 \frac{\partial|\beta|_1}{\partial\beta}$$

These are the gradient vector with respect to the coefficients.
Now rearranging by bringing all the terms to the left-hand side of the equation gives:

$$\frac{\partial(||Y - X\beta||_2^2 - \lambda_1|\beta|_1)}{\partial\beta} = 0$$

Now let $-\lambda_1 = \lambda$:

$$\frac{\partial(||Y - X\beta||_2^2 + \lambda|\beta|_1)}{\partial\beta} = 0$$

Hence, it is easy to see now that we are optimising the term, $||Y - X\beta||_2^2 + \lambda|\beta|_1$, with respect to the coefficients, β, and we can check by computing the second derivative that this is in fact the minimum. We can relate this to the equation 2.2, $\arg\min_{\beta}\{||Y - X\beta||_2^2 + \lambda|\beta|_1\}$, since to find the β which results in a minimum of the term, $||Y - X\beta||_2^2 + \lambda|\beta|_1$, we equate its first derivative with respect to β to zero.

2.2 Lasso solution proof

In this section, we will show that the closed form solution for $\hat{\beta}$ to the lasso equation, as described in the previous section, $\arg\min_{\beta}\{||Y - X\beta||_2^2 + \lambda|\beta|_1\}$, can be written as:

$$\hat{\beta} = sign(\hat{\beta}_{OLS})(|\hat{\beta}_{OLS}| - \lambda)^+ \tag{2.3}$$

where $\hat{\beta}_{OLS}$ is the ordinary least square (OLS) solution, or in other words, where $\lambda = 0$, in the linear model, $||Y - X\beta||_2^2 + \lambda|\beta|_1$. The OLS estimate, $\hat{\beta}_{OLS}$, already has a closed form solution to the linear model, being $(X^TX)^{-1}X^TY$.

First, we will start by showing the following identity:

$$||Y - X\beta||_2^2 = Y^TY - 2\beta^TX^TY + \beta^TX^TX\beta \tag{2.4}$$

Writing the L2 norm, $||Y - X\beta||_2$, as a matrix multiplication:

$$||Y - X\beta||_2^2 = (Y - X\beta)^T(Y - X\beta)$$

Using the properties of transpose of a matrix:

$$= (Y^T - \beta^T X^T)(Y - X\beta)$$

Multiplying out the brackets:

$$= Y^T Y - Y^T X\beta - \beta^T X^T Y + \beta^T X^T X\beta$$

Here, note an important fact that all the terms are scalars, as they all are 1×1 matrix. (Y being a $n \times 1$ vector, X being a $n \times p$ matrix and β being a $p \times 1$ vector, if intercept=FALSE, or X being a $n \times (p+1)$ matrix and β being a $(p+1) \times 1$ vector, if intercept=TRUE). Using the fact that the terms $Y^T X\beta$ and $\beta^T X^T Y$ are scalars and taking a transpose of a scalar is the equal to the scalar itself, we get:

$$Y^T X\beta = (Y^T X\beta)^T = \beta^T X^T Y$$

Hence, the middle two terms are equal. Let us choose to write both terms in the form $\beta^T X^T Y$

$$= Y^T Y - \beta^T X^T Y - \beta^T X^T Y + \beta^T X^T X\beta$$

And now we see the identity as stated in equation 2.4:

$$||Y - X\beta||_2^2 = Y^T Y - 2\beta^T X^T Y + \beta^T X^T X\beta$$

As the summary of the steps involved to show the identity (2.4):
$||Y - X\beta||_2^2 = (Y - X\beta)^T(Y - X\beta) = Y^T Y - Y^T X\beta - \beta^T X^T Y + \beta^T X^T X\beta = Y^T Y - 2\beta^T X^T Y + \beta^T X^T X\beta$

For the orthogonal design case $X^TX = I$,

$$||Y - X\beta||_2^2 = Y^TY - 2\beta^TX^TY + \beta^T\beta$$

$$\frac{1}{2}||Y - X\beta||_2^2 = \frac{1}{2}Y^TY - \beta^TX^TY + \frac{1}{2}\beta^T\beta$$

The orthogonal data set, X, where $X^TX = I$, means the data set is uncorrelated, and there is no correlation between the different features and each feature is independent of the others. In other words, all independent variables in the model are uncorrelated.

Differentiating $\frac{1}{2}||Y - X\beta||_2^2$ with respect to β gives:

$$\frac{\partial(\frac{1}{2}||Y - X\beta||_2^2)}{\partial\beta} = -X^TY + \beta$$

Recall the expression to find the optimal model parameters/coefficients, $\hat{\beta}$:

$$\hat{\beta} = \arg\min_{\beta}\{\frac{1}{2}||Y - X\beta||_2^2 + \lambda|\beta|_1\}$$

We have to differentiate this expression to find the minimum. We haves already differentiated $\frac{1}{2}||Y - X\beta||_2^2$ in the previous step, but note that the regularisation term, $|\beta|_1$, is non-differentiable with respect to β at zero, because $|\beta|_1 = \sum_j |\beta|_j$ and the $\lim_{\beta\to 0^+} |\beta| = 1$ is different from the $\lim_{\beta\to 0^-} |\beta| = -1$, and hence the $\lim_{\beta\to 0} |\beta|$ does not exist.
We can use the sub-differential method. There is more information on the concept of sub-differential method in the slide [2].

Note:

$$|\beta_i| = \begin{cases} \beta_i, & \text{if } \beta_i > 0, \\ -\beta_i, & \text{if } \beta_i < 0, \\ 0 & \text{if } \beta_i = 0. \end{cases} \tag{2.5}$$

and differentiating $|\beta|$, we obtain:

$$\frac{\partial|\beta|}{\partial\beta} = sign(\beta) = \begin{pmatrix} sign(\beta_1) \\ sign(\beta_2) \\ . \\ . \\ . \\ sign(\beta_p) \end{pmatrix} \tag{2.6}$$

Where,

$$sign(\beta_i) = \begin{cases} 1, & \text{if } \beta_i > 0, \\ -1, & \text{if } \beta_i < 0, \\ 0 & \text{if } \beta_i = 0. \end{cases} \tag{2.7}$$

Thus, putting it together, differentiating $\frac{1}{2}||Y - X\beta||_2^2 + \lambda|\beta|_1$ with respect to β gives:

$$\frac{\partial(\frac{1}{2}||Y - X\beta||_2^2 + \lambda|\beta|_1)}{\partial\beta} = -X^TY + \beta + \lambda sign(\beta) \tag{2.8}$$

as the derivative of the sum is equal to the sum of the derivatives.

From now, for simplicity of the proof, consider a model with only one parameter, β (and the intercept set to zero). For the single predictor model, as considered in this proof, we will show the Soft Thresholding solution. For a model with multiple predictors, we have to use the method of cyclical coordinate descent, for which there is more information on the book [16], chapter 2, page 15.
Let us consider this derivative case by case:

$$\frac{\partial(\frac{1}{2}||Y - X\beta||_2^2 + \lambda|\beta|_1)}{\partial\beta} = \begin{cases} -X^TY + \beta + \lambda & \text{if } sign(\beta) > 0, \\ -X^TY + \beta - \lambda & \text{if } sign(\beta) < 0, \\ -X^TY + \beta & \text{if } sign(\beta) = 0. \end{cases} \tag{2.9}$$

For the case $sign(\beta) > 0$:

$$\frac{\partial(\frac{1}{2}||Y - X\beta||_2^2 + \lambda|\beta|_1)}{\partial\beta} = -X^TY + \beta + \lambda$$

Equating this expression to zero gives the optimal β, denoted as $\hat{\beta}$:

$$-X^T Y + \hat{\beta} + \lambda = 0$$

$$\hat{\beta} = X^T Y - \lambda$$

For the case $sign(\beta) < 0$:

$$\frac{\partial(\frac{1}{2}||Y - X\beta||_2^2 + \lambda|\beta|_1)}{\partial\beta} = -X^T Y + \beta + -\lambda$$

Equating this expression to zero gives the optimal β, denoted as $\hat{\beta}$:

$$-X^T Y + \hat{\beta} - \lambda = 0$$

$$\hat{\beta} = X^T Y + \lambda$$

For the case $sign(\beta) = 0$:

$$-X^T Y + \hat{\beta} = 0$$

$$\hat{\beta} = X^T Y$$

The optimal β for the ordinary least squares equation, denoted a $\hat{\beta}_{OLS}$, for the orthogonal setting, where $X^T X = I$, is:

$$\hat{\beta}_{OLS} = X^T Y$$

The Ordinary least squares estimate, $\hat{\beta}_{OLS}$, corresponds to λ= 0 in the lasso model, that is the solution for $\arg\min_{\beta}\{||Y - X\beta||_2^2\}$.

Lasso solution (after replacing X^TY by $\hat{\beta}_{OLS}$):

$$\hat{\beta} = \begin{cases} \hat{\beta}_{OLS} - \lambda & \text{if } sign(\beta) > 0, \\ -\hat{\beta}_{OLS} + \lambda & \text{if } sign(\beta) < 0. \end{cases} \tag{2.10}$$

Also note:
$|\hat{\beta}_{OLS}| = \hat{\beta}_{OLS}$ for $\hat{\beta}_{OLS} > 0$
$|\hat{\beta}_{OLS}| = -\hat{\beta}_{OLS}$ for $\hat{\beta}_{OLS} < 0$

The Lasso solution can be written as:

$$\hat{\beta} = \begin{cases} |\hat{\beta}_{OLS}| - \lambda & \text{if } \hat{\beta}_{OLS} > 0, \\ -|\hat{\beta}_{OLS}| + \lambda & \text{if } \hat{\beta}_{OLS} < 0. \end{cases} \tag{2.11}$$

The nature of above equations imply that we bring the coefficients closer to zero, by subtracting a constant from positive coefficients and adding a constant to negative coefficients, thus bringing it closer to zero.

In lasso methodology, we add an additional requirement, that for $|\hat{\beta}_{OLS}|$ being less than a constant, a, we shrink the coefficient exactly to zero. Otherwise, bring the coefficient closer to zero by adding or subtracting the constant, a.
That is, we add the condition: $\hat{\beta} = 0$ for $|\hat{\beta}_{OLS}| < a$

We now see that the lasso solution, for $\arg\min_{\beta}\{\frac{1}{2}||Y - X\beta||_2^2 + \lambda|\beta|_1\}$ is:

$$\hat{\beta} = \begin{cases} |\hat{\beta}_{OLS}| - a & \text{if } \hat{\beta}_{OLS} > 0, \\ -|\hat{\beta}_{OLS}| + a & \text{if } \hat{\beta}_{OLS} < 0, \\ 0 & \text{if } |\hat{\beta}_{OLS}| < a \end{cases} \tag{2.12}$$

In this case, the constant $a = \lambda$, as shown in the working above.
This is called Soft Thresholding.

The Soft Shrinkage Lasso solution can be written as:

$$\hat{\beta} = sign(\hat{\beta}_{OLS})(|\hat{\beta}_{OLS}| - a)^+$$

Where $(|\hat{\beta}_{OLS}| - a)^+$ is the maximum between $(|\hat{\beta}_{OLS}| - a)$ and 0, and

$$sign(\hat{\beta}_{OLS}) = \begin{cases} 1 & \text{for } \hat{\beta}_{OLS} > 0, \\ -1 & \text{for } \hat{\beta}_{OLS} < 0, \\ 0 & \text{for } \hat{\beta}_{OLS} = 0 \end{cases}$$

For example, if $(|\hat{\beta}_{OLS}| - a) < 0$, that is, $|\hat{\beta}_{OLS}| < a$, then $\hat{\beta} = 0$.
For $\hat{\beta}_{OLS} > a > 0$, $\hat{\beta} = 1(|\hat{\beta}_{OLS}| - a) = \hat{\beta}_{OLS} - a$.
For $\hat{\beta}_{OLS} < -a < 0$, $\hat{\beta} = -1(|\hat{\beta}_{OLS}| - a) = -|\hat{\beta}_{OLS}| + a = -\hat{\beta}_{OLS} + a$.
This implies the lasso solution 2.3, where $a = \lambda$.

2.3 Lasso equations

Let the loss function, L be $\frac{1}{2}||Y - X\beta||_2^2 + \lambda|\beta|_1$
Using the identity in equation 2.4, $\frac{1}{2}||Y - X\beta||_2^2 = \frac{1}{2}Y^TY - \beta^TX^TY + \frac{1}{2}\beta^TX^TX\beta$
Hence,

$$L = \frac{1}{2}Y^TY - \beta^TX^TY + \frac{1}{2}\beta^TX^TX\beta + \lambda|\beta|_1$$

Differentiate with respect to β, we get:

$$L' = -X^TY + X^TX\beta + \lambda sign(\beta)$$

Equating L'=0 to find the optimal solution and rearranging terms, we get:

$$X^TX\beta = X^TY - \lambda sign(\beta)$$

Given the matric X^TX is non-singular, multiplying bother sides on the left by $(X^TX)^{-1}$ gives:

$$\beta = (X^TX)^{-1}X^TY - \lambda(X^TX)^{-1}sign(\beta)$$

This is a vector, where $(X^TX)^{-1}X^TY$ is the OLS estimate, β_{OLS}, and $(X^TX)^{-1}sign(\beta)$ is a vector that directs a line.

At $\beta_i = 0$ (shrinking the coefficient β_i, for $1 \leq i \leq p$, in a model with p parameters),

$$((X^TX)^{-1}X^TY)_i = \lambda((X^TX)^{-1}sign(\beta))_i$$

$$\lambda_i = \frac{((X^TX)^{-1}X^TY)_i}{((X^TX)^{-1}sign(\beta))_i}$$

During the lasso process, as we change quadrants, the dimensions of coefficients changes and we adjust the matrix, X^TX, by replacing the rows and columns of the lost dimension by zeros, as a result when some coefficients disappear. Let us denote the adjusted X^TX matrix, depending on the quadrant, by $(X^TX)_Q$. We then work with the generalised inverse, $(X^TX)_Q^-$, of the adjusted relevant matrix $(X^TX)_Q$, after replacing the relevant rows and columns of lost coefficients by zero. In that case, the equations are modified to:

$$\beta = (X^TX)_Q^-X^TY - \lambda(X^TX)_Q^-sign(\beta)$$

and

$$\lambda_i = \frac{((X^TX)_Q^-X^TY)_i}{((X^TX)_Q^-sign(\beta))_i}$$

The main formulas to use in lasso
The two formulas are used recursively in the lasso process:

2.4 Selection of lambda (cross-validation)

As a way to find the optimal model, we first do the lasso path, and then pick a model in the path.
In the lasso model, $\hat{\beta} = \arg\min_{\beta}\{\frac{1}{2}||Y - X\beta||_2^2 + \lambda|\beta|_1\}$, we want to find a way to select the optimal tuning parameter, λ, which is the amount of shrinkage, also known as the hyper-parameter, as its value controls the learning process. There are lots of ways to estimate the λ

in lasso regression, including Grid search, Random search, Bayesian optimisation, Gradient based optimisation (gradient descent), Evolutionary optimisation, Population Based Training, Early Stopping, and many more. More discussion on selecting the lasso parameter, λ, is included in the paper [9]. We describe how Cross-validation can be used to estimate the optimal λ.

To use the cross-validation method for estimating the optimal λ, the graph of the fraction of L1 norm against the cross-validated mean squared error is plotted in R, using the function, `cv.lars`. We then look for the fraction of L1 norm which gives the minimum mean squared error, thus fitting the model best.

The fraction of L1 norm is referred to as $\frac{|\beta(\lambda)|_1}{|\beta_{OLS}|_1}$, which is the L1 norm of β at λ, as a fraction of the maximum L1 norm of β, at $\lambda = 0$ or the OLS estimate. This is the same scale as used in the x-axis of the lasso plot against the coefficient values. The fraction of L1 norm has values between 0 and 1, 0 when all the coefficients shrink to zero at some λ, and 1 when no shrinkage of parameters is applied, at the OLS estimate.

At the optimal fraction of the L1 norm, given by cross-validation, we can use interpolation to then estimate the corresponding coefficients, using the set of values of lambda and corresponding beta, found from the lasso analysis.

After performing the cross validation to find the mean squared error at each value of the fraction of L1 norm, from 0 to 1, and finding the optimal fraction of L1 norm or percentage shrinkage, which gives the minimum mean squared error, we can retrieve the optimal coefficients and the corresponding optimal lambda, by using interpolation on two sets of coefficients corresponding to the two values of percentage shrinkage, s, between which the optimal percentage shrinkage, $\hat{s}$ lies.

2.4.1 An example for selecting coefficients using the cross-validation result

We will show an example of how the selection of coefficients works, given a cross-validation result for the optimal fraction of L1 norm or percentage shrinkage.

The lambdas and corresponding coefficients at each lasso step can be found using the `lars$beta`) and `lars$lambda` in R, or by doing the calculations by hand using the lasso equations.

Suppose that we are given the optimal percentage shrinkage, $\hat{s} = 0.4444444$, from the cross-validation result. We will now use interpolation to find the optimal coefficients, at

$\hat{s} = 0.4444444$. For that, we need to find the percentage shrinkage, $s = \frac{|\beta(\lambda)|_1}{|\beta_{OLS}|_1}$, at every lasso step, and find the two set of coefficients, corresponding to two values of s, between which the optimal s, $\hat{s} = 0.4444444$, lies, and use interpolation for each coefficient to find its optimal value.

Consider the following coefficients at each step in lasso process:

index	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8
1	0.0000000	0.0000000	0.000000000	0.000000000	0.0000000	0.0000000	0.00000000	0.000000000
2	0.3573072	0.0000000	0.000000000	0.000000000	0.0000000	0.0000000	0.00000000	0.000000000
3	0.4257216	0.0000000	0.000000000	0.000000000	0.1947727	0.0000000	0.00000000	0.000000000
4	0.4881371	0.2520752	0.000000000	0.000000000	0.4284329	0.0000000	0.00000000	0.000000000
5	0.4905858	0.2576043	0.000000000	0.003589065	0.4397189	0.0000000	0.00000000	0.000000000
6	0.5161130	0.3456454	0.000000000	0.050958515	0.5673425	0.0000000	0.00000000	0.001507652
7	0.5293721	0.3920518	-0.007810432	0.075456866	0.6005709	0.0000000	0.00000000	0.002410744
8	0.5333568	0.4126015	-0.011058591	0.085222284	0.6160884	0.0000000	0.01253390	0.002552477
9	0.5870229	0.4544606	-0.019637208	0.107054351	0.7661559	-0.1054736	0.04513596	0.004525324

The final L1 norm is:
$0.5870229 + 0.4544606 + 0.019637208 + 0.107054351 + 0.7661559 + 0.1054736 + 0.04513596 + 0.004525324 = 2.089466$

We now use interpolation, noting that the optimal percentage shrinkage, $\hat{s} = 0.4444444$, lies between the percentage shrinkage, s, observed at index 3 and 4 (only showing the relevant calculations for simplicity).

Standardised lambda for index 4:

$$s(\lambda) = \frac{(0.4881371 + 0.2520752 + 0.4284329)}{2.089466} = 0.5593032$$

Standardised lambda for index 3:

$$s(\lambda) = \frac{(0.4257216 + 0.1947727)}{2.089466} = 0.2969631$$

Interpolation for β_1:

$$\frac{(0.4444444 - 0.2969631)}{(0.5593032 - 0.2969631)} = \frac{(\beta_1 - 0.4257216)}{(0.4881371 - 0.4257216)}$$

$$\beta_1 = 0.4608101$$

Interpolation for β_2:

$$\frac{(0.4444444 - 0.2969631)}{(0.5593032 - 0.2969631)} = \frac{(\beta_2 - 0)}{(0.2520752 - 0)}$$

$$\beta_2 = 0.1417106$$

Interpolation for β_5:

$$\frac{(0.4444444 - 0.2969631)}{(0.5593032 - 0.2969631)} = \frac{(\beta_5 - 0.1947727)}{(0.4284329 - 0.1947727)}$$

$$\beta_5 = 0.3261309$$

The other coefficients, $\beta_3, \beta_4, \beta_6, \beta_7, \beta_8$ are equal to zero. We can also use a similar interpolation to find the optimal lambda, given the corresponding lambdas at index 3 and 4.

Chapter 3

Lasso Algorithm

3.1 Step by step process in lasso

We will explain the lasso paths and the trajectory of all variables/parameters shrinking to zero.
At lambda=0, we have the OLS estimate of the data (ordinary least squares). As we increase lambda, the sum of magnitude of coefficients shrink overall, that is, $|\beta|_1$ goes to zero, and it eventually hits zero for lambda large enough. We will see how to find the lambda, for example, where all the coefficients shrink to zero.
The sum of coefficients always shrink at every step, i.e. as λ increases the sum of absolute value of coefficients, $|\beta|_1$ decreases, and after finite steps all coefficients shrink to zero.
In each step of the lasso path, the coefficients either shrink towards zero or reactivate. When they reactivate, they enter into a new quadrant and when they shrink, they leave a particular quadrant, and the dimension of coefficient vector is decreased.

Let us see how lasso path jumps from quadrants.
The number of quadrants for a data set with m categories/features is 3^m At each step in the lasso process, the sign vector of coefficients move between the neighboring quadrants.

The data with two features has 9 possible quadrants:

$$\begin{array}{ccc} (0,0), & (1,0), & (-1,0), \\ (0,1), & (1,1), & (-1,1), \\ (0,-1), & (1,-1), & (-1,-1). \end{array}$$

The data with three features has 27 possible quadrants:

(0,0,0), (0,1,0), (0,-1,0),	(1,0,0), (1,1,0), (1,-1,0),	(-1,0,0), (-1,1,0), (-1,-1,0),
(0,0,1), (0,1,1), (0,-1,1),	(1,0,1), (1,1,1), (1,-1,1),	(-1,0,1), (-1,1,1), (-1,-1,1),
(0,0,-1), (0,1,-1), (0,-1,-1),	(1,0,-1), (1,1,-1), (1,-1,-1),	(-1,0,-1), (-1,1,-1), (-1,-1,-1)

3.1.1 Insights about L1 norm and quadrants

The L1 norm of coefficients is written as $|\beta|_1 = \sum_{i=1}^{p} \beta_i = |\beta_1| + |\beta_2| + |\beta_3| + ... + |\beta_p|$

The magnitude of a one-dimensional variable, the coefficient, $|\beta_i|$, is defined as:

$$|\beta_i| = \begin{cases} \beta_i, & \text{if } \beta_i > 0 \rightarrow sign(\beta) = +1, \\ 0 & \text{if } \beta_i = 0 \rightarrow sign(\beta) = 0, \\ -\beta_i, & \text{if } \beta_i < 0 \rightarrow sign(\beta) = -1 \end{cases} \tag{3.1}$$

For a simple example, let us consider a two parameter model, where $|\beta|_1 = \sum_{i=1}^{2} \beta_i = |\beta_1| + |\beta_2|$.

Consider all the possible signs of (β_1, β_2), given below:

(-1 , +1)	(0 , +1)	(+1 , +1)
(-1 , 0)	(0 , 0)	(+1 , 0)
(-1 , -1)	(0 , -1)	(+1 , -1)

The corresponding L1 norm of coefficients, $|\beta|_1$=

$-\beta_1 + \beta_2$	β_2	$\beta_1 + \beta_2$
$-\beta_1$	0	β_1
$-\beta_1 - \beta_2$	$-\beta_2$	$\beta_1 - \beta_2$

For example, in the quadrant, where $\beta_1 < 0$ and $\beta_2 > 0$, $|\beta_1| = -\beta_1$ and $|\beta_2| = \beta_2$, thus $|\beta|_1 = |\beta_1| + |\beta_2| = -\beta_1 + \beta_2$.

Hence, we have 3×3 possible quadrants, and 3×3 possible values for the L1 norm of coefficients, $|\beta|_1$, for a two parameter model, as each coefficients has three possibles signs. In general, we have 3^p possible quadrants, and 3^p possible values for the L1 norm of coefficients, written as $|\beta|_1$, for a model with p parameters, as each coefficients has three possibles signs. The possible moves to the different quadrants correspond to the possible different values for the L1 norms of the coefficients.

3.2 Lasso Methodology

Main rules about the lasso path process

1. The lambda has to be positive
2. The lambda always increases at every step in the lasso path
3. Pick the quadrant with the next smallest positive lambda, larger than the lambda at previous step
4. The quadrants, given by $sign(\beta)$, should not be repeated
5. The lasso trajectory only moves in neighbouring quadrants at each step
6. The sum of coefficients always shrink at every step

Steps in Lasso:

1. Compute the ordinary least squares estimate, $\beta_{OLS} = (X^T X)^{-1} X^T Y$, and start from the initial quadrant, given by $sign(\beta_{OLS})$.
2. Pick a quadrant, given by $sign(\beta)$.
3. Given the initial quadrant, $sign(\beta)$, find all the next possible quadrants it can jump to, which are the neighboring quadrants, considering the shrinkage and reactivation of each coefficient.
4. For a shrinkage move, pick an initial quadrant, given by $sign(\beta)$, and solve for the shrinking coefficient, β_i, by equating it to zero, in the vector equation $\beta = (X^T X)_Q^- (X^T Y)_Q - \lambda (X^T X)_Q^- sign(\beta)$, component-wise, using the matrix $(X^T X)_Q$ and the vector $(X^T Y)_Q$ from the previous step.
5. For a reactivation move, adjust the matrix $(X^T X)_Q$ and the vector $(X^T Y)_Q$ again, and solve for the reactivating coefficient only, by using the new quadrant, updated $sign(\beta)$, in the equation $\beta = (X^T X)_Q^- (X^T Y)_Q - \lambda (X^T X)_Q^- sign(\beta)$.

6. Adjust the matrix X^TX and the vector X^TY by replacing the nth rows and columns with zeros, wherever the nth row of $sign(\beta)$ is zero, $(sign(\beta))_n = 0$. Denote this by $(X^TX)_Q$ and $(X^TY)_Q$ respectively, representing the dependency on current quadrant, Q.

7. Compute the generalised inverse of the square matrix, $(X^TX)_Q$, denoted as $(X^TX)_Q^-$, and then compute the vectors $(X^TX)_Q^-(X^TY)_Q$ and $(X^TX)_Q^-sign(\beta)$.
(We need to calculate the generalised inverse of the matrix $(X^TX)_Q$ because the inverse in not defined if the matrix has some zero rows and columns. The generalised inverse of a matrix S has the following properties: $S^-SS^- = S^-$ and $SS^-S = S$. More about generalised inverse can be found in [1]).

8. Solve for each λ_i componentwise:

$$\lambda_i = \frac{((X^TX)_Q^-(X^TY)_Q)_i}{((X^TX)_Q^-sign(\beta))_i}$$

9. Go back to step two and try the next quadrant.

10. Pick the smallest lambda which satisfies all the following conditions:
It is positive, it is larger than the one chosen in previous step.

11. Pick the quadrant with smallest lambda satisfying the conditions in step 7.

12. Determine the coefficient vector, β, using the chosen λ, and the chosen quadrant, using the equation:

$$\beta = (X^TX)_Q^-(X^TY)_Q - \lambda(X^TX)_Q^-sign(\beta)$$

For example:
If the current quadrant is $(1, 0, -1, 0)^T$,

Potential Moves	Action	Using the quadrant/sign(β)
$(0,0,-1,0)^T$(shrinkage)	solve for β_1	$(1,0,-1,0)^T$
$(1,0,0,0)^T$(shrinkage)	solve for β_3	$(1,0,-1,0)^T$
$(1,-1,-1,0)^T$(reactivate)	solve for β_2	$(1,-1,-1,0)^T$
$(1,1,-1,0)^T$(reactivate)	solve for β_2	$(1,1,-1,0)^T$
$(1,0,-1,-1)^T$(reactivate)	solve for β_4	$(1,0,-1,-1)^T$
$(1,0,-1,1)^T$(reactivate)	solve for β_4	$(1,0,-1,1)^T$

3.3 Suggested algorithm/develop methodology

Implementation of quadrant lasso:

There are two possible moves for each coefficient:
Shrinkage- where a coefficient leaves a quadrant
Reactivation- where a coefficient enters a quadrant

The coefficients can only jump to the neighboring quadrants, i.e. the - sign and the + sign of a coefficient can only leave that quadrant to a zero sign, and the zero sign of a coefficient can only enter its neighboring quadrants to have either + sign or - sign.

Considering the three possible signs of the coefficient, '+', '-', and '0', the possible moves for shrinkage and reactivation are:

Shrinkage:	Reactivation:
$+1 \longrightarrow 0$	$0 \longrightarrow +1$
$-1 \longrightarrow 0$	$0 \longrightarrow -1$

For the possible moves, if sign=0, reactivate to +1 or -1, if sign= +1 or -1, shrink to 0. In general, the number of reactivation moves are number of zeros in the sign vector $\times$ two, as only zero is reactivated to two possible signs, + and -, and the number of shrinkage moves the number of non zero signs or the number of + signs plus the number of - signs, as only the non zero signs can be shrunk to zero.

If sign=0, reactivate $\rightarrow$ +1,-1, but only solve for the system of the lambdas corresponding to the coefficients being reactivated, using the new quadrant/sign vector, the one leading to reactivation.
If sign= +1 or -1 $\rightarrow$ shrink +1 or -1 to 0, solve the system of equations for lambdas corresponding to each coefficient, using the initial quadrant/sign vector, and shrink the coefficient which gives the smallest lambda.

We try to formulate the following algorithms for the lasso process:

Algorithm 1: Finding all possible potential moves to other quadrants

Input:

A string of p symbols, with each symbol having three possible values: 1, 0 or -1.

Output: A collection of strings, one for shrinkage and other for reactivation.

Result: Possible moves for a coefficient being reactivated or shrunk

. initialization;

For all the symbols, starting from the first one;

if *the symbol is 0* **then**

give a string with its value replaced by 1, other values unchanged (reactivation) ;

give a string with its value replaced by -1, other values unchanged (reactivation) ;

else

give a string with the non-zero value replaced by 0, other values unchanged (shrinkage) ;

end

List the reactivation and shrinkage moves separately, and write them as sign vectors.

Algorithm 2: Picking a quadrant and lambda to use in the calculation for coefficients

Input:

Sign vector of the initial quadrant,
and the collection of sign vectors for the reactivation steps.

Output:

Select the sign vector and lambda to use in the calculation for coefficients

Initialisation: Ignore the repeated quadrants in the reactivation steps and the shrinkage of coefficients which will give a repeated quadrant.

We solve the following component-wise, for λ_i:

$$\lambda_i = \frac{((X^TX)_Q^-(X^TY)_Q)_i}{((X^TX)_Q^- sign(\beta))_i}$$

Given an initial sign vector:

For shrinkage: Consider only the ith entries which are non-zero, in the initial sign vector, and solve for each λ_i using the sign vector of initial quadrant.

For reactivation: Consider only the ith entries which are zero, in the initial sign vector, and solve for each λ_i, using the two possible sign vectors of each ith reactivated coefficient.

For all the calculations of λs, using different sign vectors, pick the one giving the smallest positive λ_i, either from the shrinkage step or the reactivation step.

Result: Record the next smallest positive λ_i and its corresponding sign vector, in a list of quadrants visited.

Algorithm 3: Calculating the coefficients using the relevant lambda and quadrant

Input:
The lambda, λ, and the sign vector, $sign(\beta)$, selected to use in the calculation for coefficients.

Output: Give the next set of coefficients, using the lambda and quadrant from the **Input** in the calculation.

Result:
Solve the following to get a vector of next coefficients, β, in the lasso process:

$$\beta = (X^T X)^-_Q (X^T Y)_Q - \lambda (X^T X)^-_Q sign(\beta)$$

The initial string of symbols is given by the sign vector of the OLS estimate, $sign(X^T X)^{-1} X^T Y$.

3.3.1 An example using the algorithm

For example, consider the sign vector, $(0, 1, 1, 0, 1, -1, -1)^T$:

The Algorithm 1 (1) gives:

Reactivation moves	Shrinkage moves
$(1,1,1,0,1,-1,-1)^T$	$(0,0,1,0,1,-1,-1)^T$
$(-1,1,1,0,1,-1,-1)^T$	$(0,1,0,0,1,-1,-1)^T$
$(0,1,1,1,1,-1,-1)^T$	$(0,1,1,0,0,-1,-1)^T$
$(0,1,1,-1,1,-1,-1)^T$	$(0,1,1,0,1,0,-1)^T$
	$(0,1,1,0,1,-1,0)^T$

The string of symbols, (0,1,1,0,1,-1,-1), has two zeros, and five non-zero signs.
The number of reactivation moves are number of zeros times two, as each zero has two potential moves, which is $2 \times 2 = 4$, and the number of shrinkage moves is equal to the number of non-zero sign, which is 5.
The reactivation move from (0,1,1,0,1,-1,-1) to (1,1,1,0,1,-1,-1) or (-1,1,1,0,1,-1,-1), is found by replacing the first '0' by '1' or '-1', while other symbols remain the unchanged.
The reactivation move from (0,1,1,0,1,-1,-1) to (0,1,1,1,1,-1,-1) or (0,1,1,-1,1,-1,-1), is found by replacing the forth '0' by '1' or '-1', while other symbols remain the unchanged.
The other five shrinkage moves are found by replacing the each non-zero symbol, by zero, one at a time, while the other symbols remain unchanged.

The Algorithm 2 (2) gives:

Shrinkage possible for the non-zero coefficients, $\beta_2, \beta_3, \beta_5, \beta_6, \beta_7$.
Sign vector to use for shrinkage moves: $(0, 1, 1, 0, 1, -1, -1)^T$
Solve component-wise for $\lambda_2, \lambda_3, \lambda_5, \lambda_6, \lambda_7$.

Reactivation possible for the zero coefficients, β_1, β_4.
Sign vectors to use for reactivation of first coefficient, β_1: $(1, 1, 1, 0, 1, -1, -1)^T, (-1, 1, 1, 0, 1, -1, -1)^T$
Sign vectors to use for reactivation of forth coefficient, β_4: $(0, 1, 1, 1, 1, -1, -1)^T, (0, 1, 1, -1, 1, -1, -1)^T$
Solve component-wise for λ_1, λ_4.

Pick the smallest positive λ_i and the corresponding quadrant, $sign(\beta)$.
Record the quadrant visited, which will need to be ignored in the further calculations to find next lambda and coefficients.

The Algorithm 3 (3) gives:

The next set of coefficients, β, using the λ_i and $sign(\beta)$ from Algorithm 2.

For the next step in lasso process, pick the next quadrant, by using sign vector of the coefficients given by Algorithm 3, if it is not repeated, and repeat all the algorithms again, starting from Algorithm 1, otherwise, use the initial quadrant, and use the next smallest lambda in Algorithm 2 for calculation of the next set of coefficients in Algorithm 3.

We will go through a more detailed example of the lasso process, including how to adjust the matrices for calculations depending on the quadrant, in the next chapter, using a synthetic data-set.

Chapter 4

Lasso path example

4.1 Detailed lasso path example

In this section, we will reproduce the results given by the R package `lars`, used for doing the lasso analysis, and gain an understanding of the Lasso methodology algorithm.

Consider the data set:

x1:	1	0	-1	0	0	1	1	1	1	-4
x2:	1	1	1	0	0	-1	0	0	0	-2
x3:	-1	-1	-1	0	0	1	1	1	1	-1
y:	1	0	-1	0	1	-1	-1	0	-1	2

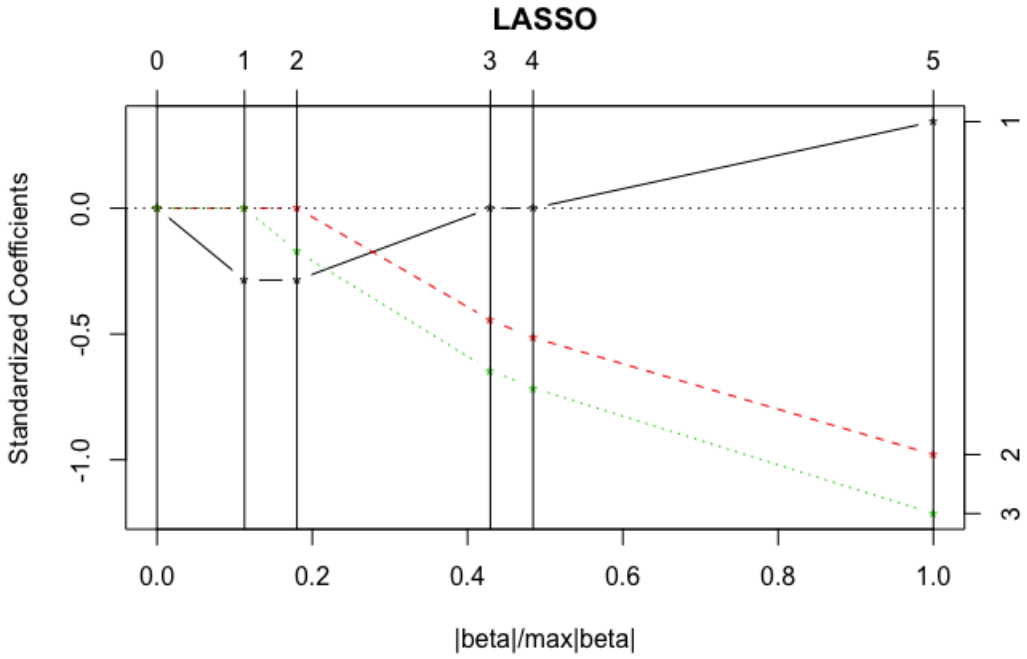
Writting the data-set as matrices, X and Y, for use in lasso calculations:

$$Y = \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 1 \\ -1 \\ -1 \\ 0 \\ -1 \\ 2 \end{pmatrix}, X = \begin{pmatrix} 1 & 1 & -1 \\ 0 & 1 & -1 \\ -1 & 1 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & -1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ -4 & -2 & -1 \end{pmatrix}$$

We use the following function in R to do the lasso analysis on the above data, considering the non-normalised case, with the intercept set to zero: `lars(y=Y, x=as.matrix(X),`

```
type="lasso", intercept=FALSE, normalize=FALSE)
```

The plot we get is:



4.1.1 Interpreting the lasso path

Let us try to interpret the lasso path plot.
The x-axis of the plot represents the percentage of shrinkage, denoted as $s(\lambda)$, and the y-axis represents the corresponding coefficients.

The percentage % shrinkage, $s(\lambda)= \frac{|\beta|_1}{\max|\beta|_1}$, where $\max|\beta|_1$ is the OLS estimate, at $\lambda = 0$, written as $\max(|beta|) = |\beta(\lambda = 0)|_1 = |\beta_{OLS}|_1$, and $|beta| = |\beta(\lambda)|_1$.
For the x-axis of the lasso plot:

$$s(\lambda) = \frac{|beta|}{\max|beta|} = \frac{|\beta(\lambda)|_1}{\max|\beta(\lambda)|_1} = \frac{|\beta(\lambda)|_1}{|\beta(\lambda = 0)|_1} = \frac{|\beta(\lambda)|_1}{|\beta_{OLS}|_1}$$

In the plot, the blue line represents β_1, the red line represents β_2, and the green line represents β_3, found by matching values from the coefficients table below. The lines in the plot represent the lasso steps, where a coefficient is shrunk or reactivated into another quadrant. At s(= 0) = 1, we have the OLS estimates, where there is no shrinkage. In the next line,

at $s(\lambda \approx 0.31) \approx 0.48$, we see that the β_1 has shrunk, as the blue line touches zero, and β_2 and β_3 are in the same negative quadrant, as they have the same sign. Notice that the sign vector of the coefficients has changed from $(+1, -1, -1)^T$ to $(0, -1, -1)^T$. In the next step, $s(\lambda \approx 0.73) \approx 0.43$, the sign vector remains the same and the coefficients are in the same quadrant but they are shrunk to a smaller magnitude to give a lower L1 norm of the coefficients, $|\beta|_1$, which gets smaller at every step of the lasso process and shrinks to zero after a finite number of steps. In the next step, at $s(\lambda \approx 1.34) \approx 0.18$, the coefficient, β_1, is reactivated to a negative quadrant, β_2 has shrunk to zero, and β_3 has shrunk to a smaller magnitude in the same quadrant. In the next step, at $s(\lambda \approx 2.71) \approx 0.11$, the β_1's value stays the same, β_2 is still zero, and β_3 also shrinks to zero. In the next step, at $s(\lambda \approx 9) \approx 0$, all the coefficients have shrunk to zero. Note, the overall sum of magnitudes, the L1 norm, of coefficients decreases at every step, as λ increases, and $s(\lambda)$ decreases, till it becomes zero at a finite λ. At $s(\lambda) = 0$, there is full shrinkage, where all the coefficients disappear. Depending on the data, the number of lasso steps to shrink all the coefficients to zero vary but it does happen after a finite number of steps. We can then perform the cross-validation, as described in the previous chapter, to find the optimal set of coefficient, which gives the lowest mean square error, the model which predicts the response values closest to the actual values.

We can make a table of the sign vectors of coefficients at each step in lasso, and compare it with the plot and the table of coefficients below, to help understand the lasso trajectory.

lasso step at a certain s(λ)	sign vector of coefficients, sign(β)
1.0000000	$(1,-1,-1)^T$
0.4843649	$(0,-1,-1)^T$
0.4292950	$(0,-1,-1)^T$
0.1802288	$(-1,0,-1)^T$
0.1126430	$(-1,0,0)^T$
0.0000000	$(0,0,0)^T$

We record the quadrants visited in the lasso process below:
This table shows the collection unique sign vectors of coefficients used in the calculations of lambda and coefficients, which we will go through in more detail later.

Starting from the OLS estimate, where standardised $\lambda = 1$, and the initial quadrant is $(1, -1, -1)^T$.
Between $(1, -1, -1)^T$ and $(0, -1, -1)^T$, the coefficients are in the quadrant, given by $sign(\beta) =$

The quadrants visited/ the sign vector of coefficients, $sign(\beta)$, between lasso steps
$(1,-1,-1)^T$
$(0,-1,-1)^T$
$(-1,-1,-1)^T$
$(-1,0,-1)^T$
$(-1,0,0)^T$
$(0,0,0)^T$

$(+1,-1,-1)^T$, which is in-between the two signs. In the plot, this is the area between line 5 and line 4. As seen in the plot, β_1 has a positive sign and β_2 and β_3 have negative signs. Also, the blue line representing β_1 is on the positive area of the plot between line 5 and line 4, and the other two lines representing β_2 and β_3 are on the negative area of the plot.
Next, the coefficients leave the quadrant $(1,-1,-1)^T$ to $(0,-1,-1)^T$, as the coefficient, β_1, shrinks.
Between $(0,-1,-1)^T$ and $(0,-1,-1)^T$, the coefficients are in the quadrant, given by $sign(\beta) = (0,-1,-1)^T$, which is in-between the two signs. In the plot, this is the area between line 4 and line 3, which makes sense because the coefficient β_1 is at zero and the other two lines in the negative area of the plot, with negative slope, show the negative values of the coefficients, β_2 and β_3, going closer towards zero, e.i. shrinking magnitude.
Next, the coefficients enter the quadrant $(-1,-1,-1)^T$ from $(0,-1,-1)^T$, as the coefficient, β_1, reappears.
Between $(0,-1,-1)^T$ and $(-1,0,-1)^T$, the coefficients are in the quadrant, given by $sign(\beta) = (-1,-1,-1)^T$, which is in-between the two signs. In the plot, this is the area between line 3 and line 2. The plot shows, the blue line, representing β_1, reactivating and moving into a negative quadrant, and the other two lines moving towards zero, and the red line touching zero at line 2, representing the total shrinkage of β_2, and the green line shows that the coefficient β_3 decreases in magnitude.
Next, the coefficients leave the quadrant $(-1,-1,-1)^T$ to $(-1,0,-1)^T$, as the coefficient, β_2, shrinks.
Between $(-1,0,-1)^T$ and $(-1,0,0)^T$, the coefficients are in the quadrant, given by $sign(\beta) = (-1,0,-1)^T$, which is in-between the two signs. In the plot, this is the area between line 2 and line 1. The plot shows the value of β_2 to be zero between this area, represented by the red line at zero, the green line, representing β_3 moves towards zero, and the blue line, representing β_1, stays at the same value.
Next, the coefficients leave the quadrant $(-1,0,-1)^T$ to $(-1,0,0)^T$, as the coefficient, β_3, shrinks.
Between $(-1,0,0)^T$ and $(0,0,0)^T$, the coefficients are in the quadrant, given by $sign(\beta) =$

$(-1, 0, 0)^T$, which is in-between the two signs. In the plot, this is the area between line 1 and line 0. In the plot, we see that the coefficients β_2 and β_3 are zero and the blue line, representing coefficient β_1, is in the negative quadrant and is moving towards zero till it touches zero at line 0, where all the coefficient have shrunk to zero.
Next, the coefficients leave the quadrant $-(1, 0, 0)^T$ to $(0, 0, 0)^T$, as the coefficient, β_1, shrinks.

As a summary, the sign vector of coefficients, $sign(\beta)$, varies from $(1, -1, -1)^T$ (between line 5 and 4), to $(0, -1, -1)^T$ (between line 4 and 3), to $(-1, -1, -1)^T$ (between line 3 and 2), to $(-1, 0, -1)^T$ (between line 2 and 1), to $(-1, 0, 0)^T$ (between line 1 and 0), to $(0, 0, 0)^T$ (at line zero).

This table below shows the coefficients, $\beta_1, \beta_2, \beta_3$, that we get at each step in the lasso process and the corresponding lambda, λ, and the percentage shrinkage, s(λ), which has values between 0 and 1, as used in the x-axis of the plot:

λ	β_1	β_2	β_3	s(λ) = $\frac{\lvert\beta\rvert_1}{max\lvert\beta\rvert_1}$
9.0000000	0.0000000	0.0000000	0.0000000	0.0000000
2.7142857	-0.2857143	0.0000000	0.0000000	0.1126430
1.3428571	-0.2857143	0.0000000	-0.1714286	0.1802288
0.7333333	0.0000000	-0.4444444	-0.6444444	0.4292950
0.3142857	0.0000000	-0.5142857	-0.7142857	0.4843649
0.0000000	0.3437500	-0.9791667	-1.2135417	1.0000000

Table 4.1: Lasso steps

Let us show an example for how the percentage shrinkage, $s(\lambda)$, is calculated, as this is the value used in the lasso plot, where $\max |\beta|_1$ is the sum of the magnitude of coefficients at the OLS estimate ($\lambda = 0$).

$$\max |\beta|_1 = |0.3437500| + | - 0.9791667| + | - 1.2135417|$$

$$= 0.3437500 + 0.9791667 + 1.2135417 = 2.536458$$

At $\lambda = 0.3142857$,

$$|\beta|_1 = |0.0000000| + |-0.5142857| + |-0.7142857|$$

$$= 0.0000000 + 0.5142857 + 0.7142857 = 1.228571$$

Hence,

$$s(\lambda = 0.3142857) = \frac{|\beta|_1}{max|\beta|_1} = \frac{1.228571}{2.536458} = 0.4843648$$

4.1.2 Step by step analysis of lasso path calculations

The table above shows the values, found by using the R package, `lars`. In this section, we will show the step by step analysis to retrieve the lambda and coefficients at each lasso step, as given by the `lars` function.
The OLS estimate or the coefficients where $\lambda = 0$ is given by $(X^TX)^{-1}X^TY$.

$$X^TX = \begin{pmatrix} 22 & 7 & 8 \\ 7 & 8 & -2 \\ 8 & -2 & 8 \end{pmatrix}, X^TY = \begin{pmatrix} -9 \\ -3 \\ -5 \end{pmatrix}$$

$$\hat{\beta}_{OLS} = \hat{\beta}(\lambda = 0) = (X^TX)^{-1}X^TY = \begin{pmatrix} 0.3437500 \\ -0.9791667 \\ -1.2135417 \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix}$$

So now we have retrieved the last column of the table, where $\lambda = 0$.
Let us call this coefficient vector as $\beta(\lambda = 0)$

The initial quadrant = $sign(\beta(\lambda = 0)) = (1, -1, -1)^T$

We find the next possible quadrants
In this case, only the shrinkage is possible.
We add this to the table:

shrink (1, -1, -1)^T	reactivate
(0,-1,-1)^T	-
(1,0,-1)^T	-
(1,-1,0)^T	-

To find which coefficient to shrink, we solve component-wise for λ

$$\lambda_i = \frac{((X^TX)_Q^-(X^TY)_Q)_i}{((X^TX)_Q^- sign(\beta))_i}$$

using the quadrant, Q=$(1,-1,-1)^T = sign(\beta)$ (no zeros, so we do not need to adjust the matrix X^TX a the vector X^TY), and decide to pick the coefficients which give the smallest λ, i.e. shrink β_i to zero if λ_i has the smallest value.

To perform the calculations for λ_i, find the vectors $(X^TX)_Q^-(X^TY)_Q$ and $(X^TX)_Q^- sign(\beta)$.

$$(X^TX)_Q^-(X^TY)_Q = \begin{pmatrix} 0.3437500 \\ -0.9791667 \\ -1.2135417 \end{pmatrix}, (X^TX)_Q^- sign(\beta) = \begin{pmatrix} 1.093750 \\ -1.479167 \\ -1.588542 \end{pmatrix}$$

Where we used $(X^TX)_Q^-(X^TY)_Q = (X^TX)^{-1}(X^TY) = \hat{\beta}_{OLS}$, and $sign(\beta) = sign(\hat{\beta}_{OLS}) = \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}$.

Now calculate λ_i, for $i = 1, 2, 3$, component-wise, by dividing $((X^TX)_Q^-(X^TY)_Q)_i$ by $((X^TX)_Q^- sign(\beta))_i$.

$$\lambda_1 = \frac{0.3437500}{1.093750} = 0.3142857, \lambda_2 = \frac{-0.9791667}{-1.479167} = 0.6619718, \lambda_3 = \frac{-1.2135417}{-1.588542} = 0.7639344$$

After finding the smallest λ_i using the above equations, find the corresponding coefficients using the value λ_i. Pick the quadrant which gives the smallest lambda, 0.3142857, thus shrink the coefficient β_1. The corresponding coefficients are given by

$$\beta(\lambda = 0.3142857) = (X^TX)_Q^-(X^TY)_Q - \lambda(X^TX)_Q^- sign(\beta)$$

$$\beta(\lambda = 0.3142857) = \begin{pmatrix} 0.3437500 \\ -0.9791667 \\ -1.2135417 \end{pmatrix} - (0.3142857) \begin{pmatrix} 1.093750 \\ -1.479167 \\ -1.588542 \end{pmatrix} = \begin{pmatrix} 0.0000000 \\ -0.5142857 \\ -0.7142857 \end{pmatrix}, \text{ where}$$

$$sign(\beta) = \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}$$

We have now retrieved the next set of coefficients, corresponding to the $\lambda = 0.3142857$. (The second last row in the table).

Pick the next quadrant, as given by $sign(\beta(\lambda = 0.3142857)) = (0, -1, -1)^T$.

We find the next possible quadrants, by considering the possible shrinkage and reactivation moves, and record this in the table below:

shrink (0, -1, -1)^T	reactivate (0, -1, -1)^T
(0,0,-1)^T	(1, -1, -1)^T – repeated quadrant
(0,-1,0)^T	(-1, -1, -1)^T

We will have to ignore the repeated quadrants, as then we will be repeating the steps.

For the shrinkage moves of $(0, -1, -1)^T$:
Adjust the matrix, $(X^TX)_Q$, by replacing the first row and column by zero, and vector, $(X^TY)_Q$, by replacing the first column by zero.

$$(X^TX)_Q = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 8 & -2 \\ 0 & -2 & 8 \end{pmatrix}, (X^TY)_Q = \begin{pmatrix} 0 \\ -3 \\ -5 \end{pmatrix}$$

As the $(X^TX)_Q$ is singular, need find its the generalised inverse, $(X^TX)_Q^-$.
Find the vectors $(X^TX)_Q^-(X^TY)_Q$ and $(X^TX)_Q^- sign(\beta)$, to help in the calculations of λ_i and finding coefficients corresponding to the smallest λ_i.

$$(X^TX)^-_Q(X^TY)_Q = \begin{pmatrix} 0.0000000 \\ -0.5666667 \\ -0.7666667 \end{pmatrix}, (X^TX)^-_Q sign(\beta) = \begin{pmatrix} 0.0000000 \\ -0.1666667 \\ -0.1666667 \end{pmatrix}$$

where $sign(\beta) = \begin{pmatrix} 0 \\ -1 \\ -1 \end{pmatrix}$

Compute the λ_is, component-wise, by dividing $((X^TX)^-_Q(X^TY)_Q)_i$ by $((X^TX)^-_Q sign(\beta))_i$. As the first entry of the above vectors, $((X^TX)^-_Q(X^TY)_Q)_1$ and $((X^TX)^-_Q sign(\beta))_1$, is zero, we ignore the calculation for λ_1, as we cannot divide by zero. We calculate the other two possible λ_is.

$$\lambda_2 = \frac{-0.5666667}{-0.1666667} = 3.4, \lambda_3 = \frac{-0.7666667}{-0.1666667} = 4.6$$

These values of λ cause the shrinkage moves, λ_2 shrinks β_2 and λ_3 shrinks β_3.
Let us consider the reactivation steps now, from the quadrant $(0, -1, -1)^T$, and see if we get a smaller value for λ.

For the reactivation move $(0, -1, -1)^T$ to $(-1, -1, -1)^T$:
To find the λ at which the quadrant $(0, -1, -1)^T$ is being reactivated to the quadrant $(-1, -1, -1)^T$, we only solve for the coefficient being reactivated, β_1. That is, we only need to solve

$$\lambda_1 = \frac{((X^TX)^-_Q(X^TY)_Q)_1}{((X^TX)^-_Q sign(\beta))_1}$$

As the reactivated quadrant, $(-1, -1, -1)^T$, does not have any zeros, we do not need to adjust the matrix X^TX and the vector X^TY, by zeros, and use its original forms. We use the updated $sign(\beta) = (-1, -1, -1)^T$, to find the λ at which β_1 reactivates into a negative quadrant from being at zero.

$$(X^TX)^-_Q(X^TY)_Q = (X^TX)^{-1}(X^TY) = \hat{\beta}_{OLS} = \begin{pmatrix} 0.3437500 \\ -0.9791667 \\ -1.2135417 \end{pmatrix}$$

,

$$sign(\beta) = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}, (X^T X)_Q^- sign(\beta) = (X^T X)^{-1} sign(\beta) = \begin{pmatrix} 0.4687500 \\ -0.7291667 \\ -0.7760417 \end{pmatrix}$$

.

$$\lambda_1 = \frac{0.3437500}{0.4687500} = 0.7333333$$

.

We see that this value of λ is smaller than the other two values, $\lambda_2 = 3.4$ and $\lambda_3 = 4.6$, found at shrinkage steps. Therefore, we select $\lambda_1 = 0.7333333$, found from the reactivation step of β_1. Thus, now we are in the quadrant $(-1, -1, -1)^T$, and the updated $sign(\beta) = (-1, -1, -1)^T$.

To find the next set of coefficients, at $\lambda \approx 0.73$, solve the equation:

$$\beta(\lambda = 0.7333333) = (X^T X)_Q^- (X^T Y)_Q - \lambda (X^T X)_Q^- sign(\beta)$$

$$\beta(\lambda = 0.7333333) = \begin{pmatrix} 0.3437500 \\ -0.9791667 \\ -1.2135417 \end{pmatrix} - (0.7333333) \begin{pmatrix} 0.4687500 \\ -0.7291667 \\ -0.7760417 \end{pmatrix} = \begin{pmatrix} 0.0000000 \\ -0.4444444 \\ -0.6444444 \end{pmatrix}, \text{ where}$$

$$sign(\beta) = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}$$

We have now retrieved the next set of coefficients. (The third last row in the table).

We are now in quadrant, as given by $sign(\beta(\lambda = 0.7333333)) = (0, -1, -1)^T$. As this is the same resultant quadrant as the previous set of coefficients, we will get the same result for the shrinkage moves. Let us consider the quadrant we moved into at the reactivation step in the previous step, $sign(\beta) = (-1, -1, -1)^T$, and consider shrinkage from there, the only possible move from this quadrant.

$$\lambda_1 = \frac{0.3437500}{0.4687500} = 0.7333333, \lambda_2 = \frac{-0.9791667}{-0.7291667} = 1.342857, \lambda_3 = \frac{-1.2135417}{-0.7760417} = 1.563758$$

We cannot pick λ_1 as it was picked in the previous step and is repeated. It will also result in repeated quadrant $(0, -1, -1)^T$. Pick the next smallest value, $\lambda_2 = 1.342857$, thus β_2 is shrunk.

To find the next set of coefficients, at $\lambda = 1.342857$, solve the equation:

$$\beta(\lambda = 1.342857) = (X^T X)_Q^- (X^T Y)_Q - \lambda (X^T X)_Q^- sign(\beta)$$

$$\beta(\lambda = 1.342857) = \begin{pmatrix} 0.3437500 \\ -0.9791667 \\ -1.2135417 \end{pmatrix} - (1.342857) \begin{pmatrix} 0.4687500 \\ -0.7291667 \\ -0.7760417 \end{pmatrix} = \begin{pmatrix} -0.2857143 \\ 0.0000000 \\ -0.1714286 \end{pmatrix}, \text{ where}$$

$$sign(\beta) = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}$$

We are now in the quadrant, as given by $sign(\beta) = (-1, 0, -1)^T$, after the coefficient, β_2, has shrunk.
We have now retrieved the next set of coefficients. (The fourth last row in the table).

We find the next possible quadrants, after $(-1, 0, -1)^T$, by considering the possible shrinkage and reactivation moves, and record this in the table below:

shrink (-1, 0, -1)^T	reactivate (-1, 0, -1)^T
(0,0,-1)^T	(-1, 1, -1)^T
(-1,0,0)^T	(-1, -1, -1)^T – repeated quadrant

We will have to ignore the repeated quadrants, as then we will be repeating the steps.

For shrinkage moves of $(-1, 0, -1)^T$:
Adjust the matrix, $(X^T X)_Q$, by replacing the second row and column by zero, and vector, $(X^T Y)_Q$, by replacing the second column by zero.

$$(X^T X)_Q = \begin{pmatrix} 22 & 0 & 8 \\ 0 & 0 & 0 \\ 8 & 0 & 8 \end{pmatrix}, (X^T Y)_Q = \begin{pmatrix} -9 \\ 0 \\ -5 \end{pmatrix}$$

As the $(X^T X)_Q$ is singular, need find its the generalised inverse, $(X^T X)_Q^-$.
Find the vectors $(X^T X)_Q^- (X^T Y)_Q$ and $(X^T X)_Q^- sign(\beta)$, to help in the calculations of λ_i and finding coefficients corresponding to the smallest λ_i.

$$(X^TX)^-_Q(X^TY)_Q = \begin{pmatrix} -0.2857143 \\ 0.0000000 \\ -0.3392857 \end{pmatrix}, (X^TX)^-_Q sign(\beta) = \begin{pmatrix} 0.0000000 \\ 0.0000000 \\ -0.1250000 \end{pmatrix}$$

where $sign(\beta) = \begin{pmatrix} -1 \\ 0 \\ -1 \end{pmatrix}$.

Compute the λ_is, component-wise, by dividing $((X^TX)^-_Q(X^TY)_Q)_i$ by $((X^TX)^-_Q sign(\beta))_i$. As the first two entries of the above vectors, $((X^TX)^-_Q(X^TY)_Q)_1$ and $((X^TX)^-_Q sign(\beta))_1$, are zero, we ignore the calculation for λ_1 and λ_2, as we cannot divide by zero (it will lead to a very large value for λ). We calculate the other possible value, λ_3.

$$\lambda_3 = \frac{-0.3392857}{-0.1250000} = 2.714286$$

This value of λ cause the shrinkage move, λ_3 shrinks β_3. Let us consider the reactivation steps now, from the quadrant $(-1, 0, -1)^T$, and see if we get a smaller value for λ.

For the reactivation move $(-1, 0, -1)^T$ to $(-1, 1, -1)^T$:
To find the λ at which the quadrant $(-1, 0, -1)^T$ is being reactivated to the quadrant $(-1, 1, -1)^T$, we only solve for the coefficient being reactivated, β_2. That is, we only need to solve

$$\lambda_2 = \frac{((X^TX)^-_Q(X^TY)_Q)_2}{((X^TX)^-_Q sign(\beta))_2}$$

As the reactivated quadrant, $(-1, 1, -1)^T$, does not have any zeros, we do not need to adjust the matrix X^TX and the vector X^TY, by zeros, and use its original forms. We use the updated $sign(\beta) = (-1, 1, -1)^T$, to find the λ at which β_2 reactivates into a positive quadrant from being at zero.

$$(X^TX)^-_Q(X^TY)_Q = (X^TX)^{-1}(X^TY) = \hat{\beta}_{OLS} = \begin{pmatrix} 0.3437500 \\ -0.9791667 \\ -1.2135417 \end{pmatrix}$$

,

$$sign(\beta) = \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}, (X^T X)_Q^- sign(\beta) = (X^T X)^{-1} sign(\beta) = \begin{pmatrix} -0.281250 \\ 0.437500 \\ 0.265625 \end{pmatrix}$$

.

$$\lambda_2 = \frac{-0.9791667}{0.437500} = -2.238095$$

.

We ignore this value of λ because it is negative. Therefore, we select $\lambda_3 = 2.714286$, found from the shrinkage step, where β_3 is shrunk.
To find the next set of coefficients, at $\lambda = 2.714286$, solve the equation:

$$\beta(\lambda = 2.714286) = (X^T X)_Q^- (X^T Y)_Q - \lambda (X^T X)_Q^- sign(\beta)$$

$$\beta(\lambda = 2.714286) = \begin{pmatrix} -0.2857143 \\ 0.0000000 \\ -0.3392857 \end{pmatrix} - (2.714286) \begin{pmatrix} 0.0000000 \\ 0.0000000 \\ -0.1250000 \end{pmatrix} = \begin{pmatrix} -0.2857143 \\ 0.0000000 \\ 0.0000000 \end{pmatrix}, \text{ where}$$

$$sign(\beta) = \begin{pmatrix} -1 \\ 0 \\ -1 \end{pmatrix}$$

We have retrieved the next set of coefficients. (The fifth last row in the table).
We are now in the quadrant $(-1, 0, 0)^T$, as $sign(\beta(\lambda = 2.714286)) = (-1, 0, 0)^T$.

We find the next possible quadrants, after $(-1, 0, 0)^T$, by considering the possible shrinkage and reactivation moves, and record this in the table below:

shrink (-1, 0, 0)^T	reactivate (-1, 0, 0)^T
(0,0,0)^T	(-1, 1, 0)^T
-	(-1, -1, 0)^T
-	(-1, 0, 1)^T
-	(-1, 0, -1)^T − repeated quadrant

We will have to ignore the repeated quadrants, as then we will be repeating the steps.

For shrinkage moves:
Adjust the matrix, $(X^T X)_Q$, by replacing the second and third row and column by zero, and

vector, $(X^TY)_Q$, by replacing the second and third column by zero.

$$(X^TX)_Q = \begin{pmatrix} 22 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, (X^TY)_Q = \begin{pmatrix} -9 \\ 0 \\ 0 \end{pmatrix}$$

As the matrix $(X^TX)_Q$ is singular, need find its the generalised inverse, $(X^TX)_Q^-$.
Find the vectors $(X^TX)_Q^-(X^TY)_Q$ and $(X^TX)_Q^- sign(\beta)$, to help in the calculations of λ_i and finding coefficients corresponding to the smallest λ_i.

$$(X^TX)_Q^-(X^TY)_Q = \begin{pmatrix} -0.4090909 \\ 0.0000000 \\ 0.0000000 \end{pmatrix}, (X^TX)_Q^- sign(\beta) = \begin{pmatrix} -0.04545455 \\ 0.00000000 \\ 0.00000000 \end{pmatrix}$$

where $sign(\beta) = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$
Compute the λ_is, component-wise, by dividing $((X^TX)_Q^-(X^TY)_Q)_i$ by $((X^TX)_Q^- sign(\beta))_i$.
As the last two entries of the vectors, $(X^TX)_Q^-(X^TY)_Q$ and $(X^TX)_Q^- sign(\beta)$, are zero, we ignore the calculation for λ_2 and λ_3, as we cannot divide by zero (it will lead to a very large value for λ). We calculate the other possible value λ_1.

$$\lambda_1 = \frac{-0.4090909}{-0.04545455} = 9$$

This value of λ cause the shrinkage move, λ_1 shrinks β_1. This now results in all coefficients being shrunk to zero.
Let us consider the reactivation steps also, from the quadrant $(-1, 0, 0)^T$, and see if we get a smaller value for λ.

For the reactivation move $(-1, 0, 0)^T$ to $(-1, 1, 0)^T$ or $(-1, -1, 0)^T$:
To find the λ at which the quadrant $(-1, 0, 0)^T$ is being reactivated to the quadrant $(-1, 1, 0)^T$ or $(-1, -1, 0)^T$, we only solve for the coefficient being reactivated, β_2. That is, we only need to solve

$$\lambda_2 = \frac{((X^TX)_Q^-(X^TY)_Q)_2}{((X^TX)_Q^- sign(\beta))_2}$$

Adjust the matrix, $(X^TX)_Q$, by replacing the third row and column by zero, and vector, $(X^TY)_Q$, by replacing the third column by zero.

$$(X^TX)_Q = \begin{pmatrix} 22 & 7 & 0 \\ 7 & 8 & 0 \\ 0 & 0 & 0 \end{pmatrix}, (X^TY)_Q = \begin{pmatrix} -9 \\ -3 \\ 0 \end{pmatrix}$$

As the $(X^TX)_Q$ is singular, need find its the generalised inverse, $(X^TX)_Q^-$.
Find the vectors $(X^TX)_Q^-(X^TY)_Q$ and $(X^TX)_Q^- sign(\beta)$, to help in the calculations of λ_i and finding coefficients corresponding to the smallest λ_i.

$$(X^TX)_Q^-(X^TY)_Q = \begin{pmatrix} -0.40157480 \\ -0.02362205 \\ 0.00000000 \end{pmatrix}$$

$$(X^TX)_Q^- sign(\beta) = \begin{pmatrix} -0.1181102 \\ 0.2283465 \\ 0.0000000 \end{pmatrix}$$

where $sign(\beta) = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}$

and

$$(X^TX)_Q^- sign(\beta) = \begin{pmatrix} -0.007874016 \\ -0.118110236 \\ 0.000000000 \end{pmatrix}$$

where $sign(\beta) = \begin{pmatrix} -1 \\ -1 \\ 0 \end{pmatrix}$

The value of λ on reactivating $(-1,0,0)^T$ to $(-1,1,0)^T$ is

$$\frac{-0.02362205}{0.2283465} = -0.1034483$$

which we ignore because it is negative.
The value of λ on reactivating $(-1,0,0)^T$ to $(-1,-1,0)^T$ is

$$\frac{-0.02362205}{-0.118110236} = 0.2$$

which we ignore because it is less than the λ we got in the previous step of the lasso process.

For the reactivation move $(-1,0,0)^T$ to $(-1,0,1)^T$ or $(-1,0,-1)^T$:
To find the λ at which the quadrant $(-1,0,0)^T$ is being reactivated to the quadrant $(-1,0,1)^T$ or $(-1,0,-1)^T$, we only solve for the coefficient being reactivated, β_3. That is, we only need to solve

$$\lambda_3 = \frac{((X^TX)_Q^-(X^TY)_Q)_3}{((X^TX)_Q^- sign(\beta))_3}$$

Adjust the matrix, $(X^TX)_Q$, by replacing the second row and column by zero, and vector, $(X^TY)_Q$, by replacing the second column by zero.

$$(X^TX)_Q = \begin{pmatrix} 22 & 0 & 8 \\ 0 & 0 & 0 \\ 8 & 0 & 8 \end{pmatrix}, (X^TY)_Q = \begin{pmatrix} -9 \\ 0 \\ -5 \end{pmatrix}$$

As the $(X^TX)_Q$ is singular, we need find its the generalised inverse, $(X^TX)_Q^-$.
Find the vectors $(X^TX)_Q^-(X^TY)_Q$ and $(X^TX)_Q^- sign(\beta)$, to help in the calculations of λ_i and finding coefficients corresponding to the smallest λ_i.

$$(X^TX)_Q^-(X^TY)_Q = \begin{pmatrix} -0.2857143 \\ 0.0000000 \\ -0.3392857 \end{pmatrix}, (X^TX)_Q^- sign(\beta) = \begin{pmatrix} -0.1428571 \\ 0.0000000 \\ 0.2678571 \end{pmatrix}$$

where $sign(\beta) = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$

The value of λ on reactivating $(-1, 0, 0)^T$ to $(-1, 0, 1)^T$ is

$$\frac{-0.3392857}{0.2678571} = -1.266667$$

which we ignore because it is negative.

Therefore, we select $\lambda_1 = 9$, found from the shrinkage step, where β_1 is shrunk.
To find the next set of coefficients, at $\lambda = 9$, solve the equation:

$$\beta(\lambda = 9) = (X^T X)_Q^- (X^T Y)_Q - \lambda (X^T X)_Q^- sign(\beta)$$

$$\beta(\lambda = 9) = \begin{pmatrix} 0.4090909 \\ 0.0000000 \\ 0.0000000 \end{pmatrix} - (9) \begin{pmatrix} 0.04545455 \\ 0.0000000 \\ 0.0000000 \end{pmatrix} = \begin{pmatrix} 0.0000000 \\ 0.0000000 \\ 0.0000000 \end{pmatrix}$$

where $sign(\beta) = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$

Thus, we conclude all the coefficients shrink to zero at $\lambda = 9$, and we retrieve the sixth last row/the first row in the table.

4.1.3 Summary of steps

As a summary, we were first in the quadrant $(1, -1, -1)^T$ as given by the OLS estimate, where only shrinkage was possible. Then, the coefficient, β_1 was shrunk to give the sign vector of coefficients being $(0, -1, -1)^T$. Then, the coefficient, β_1 was reactivated, giving the resulting quadrant $(-1, -1, -1)^T$, which then caused β_1 to be shrunk again, to give the sign vector of coefficients $(0, -1, -1)^T$. Considering the reactivation again to the quadrant $(-1, -1, -1)^T$, then the coefficient, β_2 was shrunk, giving the sign vector of coefficients $(-1, 0, -1)^T$. Then, the coefficient, β_3, was shrunk, giving the sign vector of coefficients $(-1, 0, 0)^T$. And finally, all the coefficients were shrunk to $(0, 0, 0)$.

Given the initial quadrant, $sign(\beta_{OLS}) = (1, -1, -1)^T$:

Moves made	Action	Using the quadrant/sign(β)
(0,-1,-1)^T(shrinkage)	solve for β_1	(1,-1,-1)^T
(-1,-1,-1)^T(reactivate)	solve for β_1	(-1,-1,-1)^T
(-1,0,-1)^T(shrinkage)	solve for β_2	(-1,-1,-1)^T
(-1,0,0)^T(shrinkage)	solve for β_3	(-1,0,-1)^T
(0,0,0)^T(shrinkage)	solve for β_1	(0,0,-1)^T

Given:

$$\beta = (X^T X)_Q^- (X^T Y)_Q - \lambda (X^T X)_Q^- sign(\beta)$$

To solve for $\beta_i = 0$, solve for λ_i:

$$\lambda_i = \frac{((X^T X)_Q^- (X^T Y)_Q)_i}{((X^T X)_Q^- sign(\beta))_i}$$

Find the coefficients using the relevant λ_i and $sign(\beta)$:

$$\beta = (X^T X)_Q^- (X^T Y)_Q - \lambda (X^T X)_Q^- sign(\beta)$$

Chapter 5

Conclusion

We have looked at the step by step process in lasso and tried to develop some algorithms, generating the shrinkage and reactivation moves, which we need to consider, a method to pick an optimal quadrant, and find corresponding coefficients. We have shown the main equations used to find coefficients in the lasso process and how the model is selected, with a cross-validation example to select the optimal percentage shrinkage, and use interpolation to find the optimal coefficients.

In the algorithms for lasso methodology discussed, we have to manually record the quadrants visited and ignore them in the further calculations, in order to not repeat the quadrants and calculations. As an improvement to the algorithm discussed, more advanced ways could be developed to give more efficient algorithms for the generation of moves between quadrants in the lasso methodology which generates only the relevant possible moves such that no quadrants are repeated, that is, an algorithm which generates the next possible moves, without giving the repeated quadrants, as it can save a lot of computations, improving the efficiency.

The lasso regression is used widely in machine learning to make better predictions. Lasso is very useful for detecting and preventing over-fitting in a model, and in simplifying a model so that only the most relevant parameters are included. This makes he model more interpretable, in practice, and is very useful for doing statistical analysis and making more reliable predictions.

Due to the nature of lasso constraint, and it making the regression model non-deferentiable after adding the constraint, we have to develop the analytical methods to optimise the lasso regression equation, and selection the tuning parameter, which selects a particular model in the lasso solution path. The solution for lasso equation is an interesting realisation in the mathematical area of optimisation of quadratic problem with linear constraint, and this idea can be applied in the development more areas in mathematics and machine learning.

Appendix A

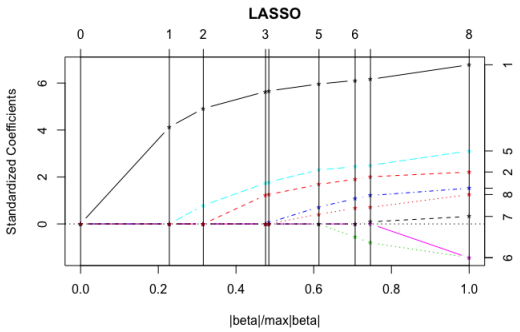
Appendix Title

A.1 Lasso analysis in R

The book, [15], chapter 3, pages 49-51, discusses the prostate cancer data example, which can be found in the R package `lasso2`. It shows that the coefficients, age, lcp, gleason and pgg45 are not significant using the F-test, and comparing the correlations of different predictors show that the coefficient lcavol, svi and lweight are most significant predictors for the response variable, lpsa, and lcavol and lcp are highly correlated, so lcp does not need to be included in a model once the coefficient lcavol is present. The pages 61-93 of the book compare the different regularisation methods and shows how the cross-validation is used to select the optimal coefficients of the model. More analysis on this example and the R codes is included in [6].

The least squares model, in which the predictors are standardised, is fitted as:
`lm(lpsa ~ scale(lcavol)+scale(lweight)+scale(age)+scale(lbph)`
`+scale(svi)+scale(lcp)+scale(gleason)+scale(pgg45),Prostate)`
We retrieve the exact coefficients, as in the paper [12], using the `summary` function.

The lars model is fitted as:
`lars(x=data[,-c(9)],y=data[,9],type="lasso", intercept = TRUE, normalize = TRUE)`
The coefficients are normalised in this model, which is slightly different from being scaled, as scaling divides each column of the data matrix, X, by its standard deviation after centering the data but normalising divides each column by its L1 norm.
The plot of the lasso trajectory we get is:



We can see from the plot that for this data, we only have shrinkage moves, as no coefficients are being reactivated after shrinking to zero, so it is a simple example with no reactivation steps. This plot shows the similar trajectory of lasso steps to the one on the lasso paper. The x-axis of the plot is referred to as "s" in the lasso paper, [12], representing the fraction of final L1 norm.

We will go through a synthetic data later for the detailed lasso path analysis but in the section, we will just focus on the method of getting the optimal coefficients using cross-validation in R.

We perform the cross-validation using the code:
`cv.lars(x=data[,-c(9)],y=data[,9],type="lasso", intercept = TRUE, normalize = TRUE)`
This returns a plot of the graph, which is plotted using the cross-validation MSE on the y-axis, against the fraction of final L1 norm on the x-axis, which is written as $\frac{|\beta(\lambda)|_1}{|\beta_{OLS}|_1}$, and referred as "$s$" in the lasso paper, [12]. The cross-validation was performed at each fraction of L1 norm continuously and the mean values are plotted, along with the standard error, the minimum and maximum value at each point.

We see the values for all cross-validated MSE, as default 100 values, using the code:
`cv.lars(x=data[,-c(9)],y=data[,9],type="lasso", intercept = TRUE, normalize = TRUE)$cv`

We can then use the `min` function to find the minimum mean cross-validated MSE and add it to its standard error to find the limit. The function `cv.error` is used to find the standard error of the cross-validated MSE curve.

We find the optimal value of the fraction of final L1 norm using cross-validation, using

the following code:

```
cvA<- cv.lars(x=data[,-c(9)],y=data[,9],type="lasso", intercept = TRUE, normalize
= TRUE)
limit <- min(cvA$cv) + cvA$cv.error[which.min(cvA$cv)]
optimal.s <- cvA$index[min(which(cvA$cv < limit))]
optimal.s
```

This gives the cross-validated value of the optimal s, the fraction of final L1 norm, which we will use it to find the optimal coefficients in the lasso model. We get slightly different value for s every time we run the cross validation, and in a certain iteration of running the code we get optimal.s=0.4444444. This is the value for optimal s, $\hat{s} = 0.44$, as given in the paper. We get the optimal coefficients, using the cross-validated fraction of final L1 norm, $\hat{s} = 0.44$, using the code:

```
predict(A,s=0.44,mode='fraction',type='coefficients')$coef
or coef(A, s=0.44, mode="fraction")
```

Using the code above, we get the following optimal coefficients:

lcavol	lweight	age	lbph	svi	lcp	gleason	pgg45
0.4740831	0.1953156	0	0	0.3758199	0	0	0

These coefficients are slightly different from the ones given in the paper, [12], which is likely because of the different scaling used. However, we do get the same coefficients selected by the model and same signs of the coefficients, like on the paper, "Table 1, page 274, on [12]".

We see that the model has been simplified, now having only the three most relevant features. Although, this model might give a slightly higher sum of error terms squares due to the added constraint on coefficients but we expect this model to perform better on the unseen data, and have a better prediction ability. The lasso methodology has detected the less significant features and shrunk them to zero, simplifying the model, and preventing the over-fitting.

A.2 Lasso analysis comparision for `normalise=TRUE` and `FALSE`

The $\hat{\beta}_0$ is often referred to as intercept or constant in the linear model. Without the loss of generality, we can take the intercept to be zero and get rid of $\hat{\beta}_0$ in the linear model, and

hence get rid of the ones column of in X matrix, by subtracting $\hat{\beta}_0$ from all the response variable values in the vector, Y. In other words, this is same as subtracting $\hat{\beta}_0$ from both sides of the equation in a linear model. Also, if we centre all the variables, i.e. subtract the mean observation from each feature and the mean result from the response variable, in a one feature model, then the intercept is zero.

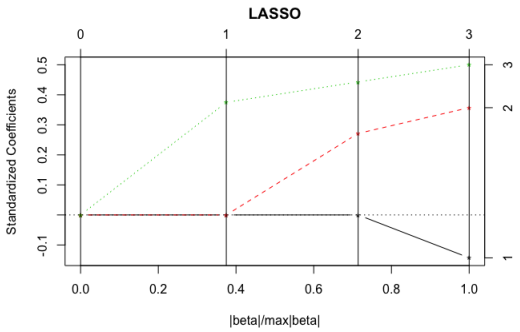
The intercept can be removed from the analysis after centering the response variable, where mean of the response variable, mean(Y), is $\hat{\beta}_{OLS}$. To centre the response variable, subtract the mean(Y) from the Y values.

For lasso analysis, using the `lars` function in R, we have the option to select whether the `intercept` = `TRUE` or `FALSE`, or `normalise` = `TRUE` or `FALSE`. If we select that intercept = `FALSE`, then the model is adjusted so that the intercept is zero. If we select that normalise = `TRUE`, then the columns in the X matrix are normalised or scaled such that they have a unit L2 norm. These different options or variations of the same linear model just scales the variables differently but the lasso trajectory plots can look quite different.

For the following data-set:

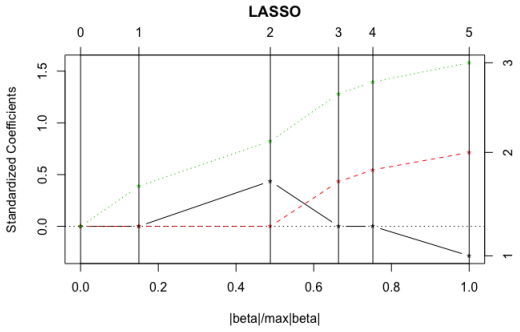
x1:	0	0	-1	0	1	-1	1
x2:	0	0	-1	1	1	-1	0
x3:	1	-1	-1	-1	1	-1	2
y:	0	-1	0	0	1	-1	1

The plot we get for `normalise=FALSE` (and `intercept=FALSE)`) is:



beta
x1 x2 x3
0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.3750000
0.0000000 0.2714286 0.4428571
-0.1428571 0.3571429 0.5000000
attr(,"scaled:scale")
1 1 1
lambda
5.00000000 1.25000000 0.02857143

The plot we get for `normalise=TRUE` (and `intercept=FALSE)`) is:



beta
x1 x2 x3
0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.1225148

0.2182863 0.0000000 0.2605712
0.0000000 0.2158682 0.4052894
0.0000000 0.2729161 0.4413696
-0.1428571 0.3571429 0.5000000
attr(,"scaled:scale")
2.000000 2.000000 3.162278
lambda
1.58113883 1.19371294 0.41199928 0.16297412 0.01279821

In this output, the R also normalises the beta coefficients from the lasso calculations in the same way as it did for all the data corresponding to that parameter, but the plot has the original values from the calculations.
The L2 norm of x1 data is 2, the L2 norm of the x2 data 2, and the L2 norm of the x3 data 3.162278.
The OLS estimate for this data is: $(-0.1428571, 0.3571429, 0.5000000)^T$ for the unnormalised data, and $(-0.2857142857, 0.7142857143, 1.5811388301)^T$ for the normalised data.
However, note that dividing the OLS vector component-wise by the scale gives:

$$\frac{(-0.2857142857, 0.7142857143, 1.5811388301)^T}{(2, 2, sqrt(10))^T} = (-0.1428571, 0.3571429, 0.5000000)^T$$

This is the value given by the R output, for the last row of beta, using the function `lars$beta`.
However, it can be seen that the lasso path plot uses the true values for the coefficients, found by calculations as discussed in the lasso algorithm, which are $\beta_1 = -0.2857142857, \beta_2 = 0.7142857143, \beta_3 = 1.5811388301$.
This is true for all values of coefficients, beta, given by R for the normalised case.
We get a different lasso path plot, depending on whether the data was normalised or not. The unnormalised data produces a plot which only shows the shrinkage moves of coefficients, with no reactivation observed. In the plot for normalised data, the coefficient, β_1, is reactivated once, while the other two coefficients were consistently moving towards zero, and the coefficient, β_3, being the last one to shrink.

A.3 More on cross-validation

Cross-validation is often used to test the generalisation performance. To test the prediction accuracy or generalisation ability of a model, we can use different types of cross validation techniques such as an out of sample validation, where we divide the data-set into two parts: a training set and testing set, using a certain proportion of data for training and a certain, usually smaller, proportion of data for testing. Other methods for cross-validation include a leave one out cross-validation, where we test on one value at a time, and the k-folds cross-validation, which is usually more accurate, as every observation in the data-set goes through being a part of testing sample once, and used for training the other times, and multiple testing and training is carried out.

The k-folds cross-validation works by splitting or partitioning the data-set into k parts, and using each part as a testing set, one by one, while the other parts are used as a training set, from which the model is trained.

Bibliography

[1] *Appendix A. Matrix Algebra, Generalized Inverse*. URL: https://www.stt.msu.edu/users/pszhong/Generalized-Inverse-and-Projectors-C.R.Rao_Book_Appendix.pdf.

[2] Taylor B. Arnold. *Intro to Lasso Regression*. 2015. URL: http://statsmaths.github.io/stat612/lectures/lec17/lecture17.pdf.

[3] IAIN JOHNSTONE BRADLEY EFRON TREVOR HASTIE and ROBERT TIBSHIRANI. "LEAST ANGLE REGRESSION". In: *The Annals of Statistics* 32.2 (2004), pp. 407–499. URL: http://statweb.stanford.edu/~tibs/ftp/lars.pdf.

[4] Andrew Chamberlain. *A Simple Explanation of Why Lagrange Multipliers Works*. URL: https://medium.com/@andrew.chamberlain/a-simple-explanation-of-why-lagrange-multipliers-works-253e2cdcbf74.

[5] S. Lunag´omez H. Maruri-Aguilar. *Lasso for hierarchical polynomial models*. 2020. URL: https://arxiv.org/pdf/2001.07778.pdf.

[6] Jo Hardin. *Shrinkage Methods – R code Ridge Regression LASSO*. URL: http://pages.pomona.edu/~jsh04747/courses/math158/shrink_RRLASSO.pdf.

[7] Trevor Hastie. *Regularization Paths*. 2006. URL: https://www.r-project.org/conferences/useR-2006/Slides/Hastie.pdf.

[8] ROBERT TIBSHIRANI Jacob Bien Jonathan Taylor. "A LASSO FOR HIERARCHICAL INTERACTIONS". In: *The Annals of Statistics* 41.3 (2013), 1111–1141. URL: https://arxiv.org/pdf/1205.5050.pdf.

[9] Frans Kanfer Lisa-Ann Kirkland and Sollie Millard. *LASSO TUNING PARAMETER SELECTION*. URL: https://www.researchgate.net/profile/Lisa_Kirkland/publication/287727878_LASSO_Tuning_Parameter_Selection/links/5678ffa908ae502c99d6d7pdf.

[10] *Statistical Modeling and Analysis of Neural Data, Least Squares Regression*. URL: http://pillowlab.princeton.edu/teaching/statneuro2018/slides/notes03b_LeastSquaresRegression.pdf.

[11] *The Lasso Page*. URL: http://statweb.stanford.edu/~tibs/lasso.html.

[12] Robert Tibshirani. "Regression Shrinkage and Selection Via the Lasso". In: *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B (Methodological)* 58.1 (1996), pp. 267–288. URL: http://statweb.stanford.edu/~tibs/lasso/lasso.pdf.

[13] Robert Tibshirani. *THE LASSO METHOD FOR VARIABLE SELECTION IN THE COX MODEL*. URL: http://statweb.stanford.edu/~tibs/lasso/fulltext.pdf.

[14] Ryan J. Tibshirani. *The Lasso Problem and Uniqueness*. URL: http://www.stat.cmu.edu/~ryantibs/papers/lassounique.pdf.

[15] Jerome Friedman Trevor Hastie Robert Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition*. Springer, 2009. URL: https://web.stanford.edu/~hastie/ElemStatLearn//printings/ESLII_print12.pdf.

[16] Martin Wainwright Trevor Hastie Robert Tibshirani. *Statistical Learning with Sparsity: The Lasso and Generalizations*. 2016. URL: https://web.stanford.edu/~hastie/StatLearnSparsity_files/SLS.pdf.

Code for generating the possible moves to other quadrants in lasso

In [1]:
```
from numpy import *
```

In [2]:
```
#reactivation moves
#for "a" being a list of symbols, with possible values being 0,-1,1
def moves_r(a):
    for i in range(len(a)):
        b=[0]*len(a)
        c=[0]*len(a)
        if a[i]==0:
            b[i]=a[i]+1
            print("reactivation:", array(b)+array(a))
            c[i]=a[i]-1
            print("reactivation:", array(c)+array(a))
```

In [3]:
```
#shrinkage moves
#for "a" being a list of symbols, with possible values being 0,-1,1
def moves_s(a):
    for i in range(len(a)):
        d=[0]*len(a)
        if a[i]==1:
            d[i]=a[i]-2
            print("shrinkage:", array(d)+array(a))
        if a[i]==-1:
            d[i]=a[i]+2
```

In [4]:
```
#example
a=[0,-1,1,0,1,0,-1,0,1]
moves_r(a)
moves_s(a)
```

```
reactivation: [ 1 -1  1  0  1  0 -1  0  1]
reactivation: [-1 -1  1  0  1  0 -1  0  1]
reactivation: [ 0 -1  1  1  1  0 -1  0  1]
reactivation: [ 0 -1  1 -1  1  0 -1  0  1]
reactivation: [ 0 -1  1  0  1  1 -1  0  1]
reactivation: [ 0 -1  1  0  1 -1 -1  0  1]
reactivation: [ 0 -1  1  0  1  0 -1  1  1]
reactivation: [ 0 -1  1  0  1  0 -1 -1  1]
shrinkage: [ 0  0  1  0  1  0 -1  0  1]
shrinkage: [ 0 -1  0  0  1  0 -1  0  1]
shrinkage: [ 0 -1  1  0  0  0 -1  0  1]
shrinkage: [ 0 -1  1  0  1  0  0  0  1]
shrinkage: [ 0 -1  1  0  1  0 -1  0  0]
```

In [5]:
```
#all moves
#for "a" being a list of symbols, with possible values being 0,-1,1
from numpy import *
def moves(a):
    for i in range(len(a)):
        b=[0]*len(a)
        c=[0]*len(a)
        d=[0]*len(a)
        if a[i]==0:
            b[i]=a[i]+1
            print("reactivation:", array(b)+array(a))
            c[i]=a[i]-1
            print("reactivation:", array(c)+array(a))
        if a[i]==1:
            d[i]=a[i]-2
            print("shrinkage:", array(d)+array(a))
        if a[i]==-1:
            d[i]=a[i]+2
            print("shrinkage:", array(d)+array(a))
```

In [6]:
```
#example
a=[0,-1,1,0,1,0,-1,0,1]
moves(a)
```

```
reactivation: [ 1 -1  1  0  1  0 -1  0  1]
reactivation: [-1 -1  1  0  1  0 -1  0  1]
shrinkage: [ 0  0  1  0  1  0 -1  0  1]
shrinkage: [ 0 -1  0  0  1  0 -1  0  1]
reactivation: [ 0 -1  1  1  1  0 -1  0  1]
reactivation: [ 0 -1  1 -1  1  0 -1  0  1]
shrinkage: [ 0 -1  1  0  0  0 -1  0  1]
reactivation: [ 0 -1  1  0  1  1 -1  0  1]
reactivation: [ 0 -1  1  0  1 -1 -1  0  1]
shrinkage: [ 0 -1  1  0  1  0  0  0  1]
reactivation: [ 0 -1  1  0  1  0 -1  1  1]
reactivation: [ 0 -1  1  0  1  0 -1 -1  1]
shrinkage: [ 0 -1  1  0  1  0 -1  0  0]
```

In [ ]:

In [ ]: