# REGULARIAZTION AND VARIABLE SELECTION VIA THE FIXED-SHAPE ELASTIC NET AND EXPONENTIAL NORM

BY

**Abdulrahman Khan**

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

**KING FAHD UNIVERSITY OF PETROLEUM & MINERALS**

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

**MASTER OF SCIENCE**

**In**

**INDUSTRIAL AND SYSTEMS ENGINEERING**

**May 2017**

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **ABDULRAHMAN KHAN** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN INDUSTRIAL AND SYSTEMS ENGINEERING**.

**Thesis Committee**

Mujahid

Dr. Syed N. Mujahid (Adviser)

Dr. Hesham K. Al-Fares (Member)

Dr. Shokri Z. Selim (Member)

Dr. Hesham K. Al-Fares
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

30/11/17

Date

*To my beloved mother Duha*
*To my beloved father Mohyiddin*
*To my beloved wife Jumana*
*To my beloved daughters Duha and Jana*

# ACKNOWLEDGMENTS

*All praise is to Allah, and peace and blessings be upon his messenger. I would like to express my gratitude to King Fahd University of Petroleum & Minerals for facilitating the good environment for study and research. I would like to express my gratitude for my thesis advisor Dr. Syed Mujahid for his enormous support and dedication. I would like to thank my thesis committee Dr. Hesham Al-Fares and Dr. Shokri Selim. I would like to thank my parents for their prayers and motivation. I would like to thank my wife for her sacrifice and support for me to complete the thesis.*

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# THESIS ABSTRACT

| | |
|---|---|
| **NAME:** | Abdulrahman Khan |
| **TITLE OF STUDY:** | Regularization And Variable Selection Via The Fixed-Shape Elastic Net And Exponential Norm |
| **MAJOR FIELD:** | Industrial And Systems Engineering |
| **DATE OF DEGREE:** | May 2017 |

*The Penalized Linear Regression (PLR) is one of the tools that provide regularization and variable selection for the coefficient estimates of regression model. The PLR usually consists of the loss function, such as the Ordinary Least Squares (OLS), plus the penalized regularization term. The OLS method is hardly useful in real data interpretation. The Mean Square Error (MSE), which consists of the sum of variance error and bias-squared error, can be high for OLS, even though there is no bias-squared error. The PLR, such as the ridge regression, the lasso, and the elastic net, adds a bias-squared error, which may reduce the variance error, thereby, reducing the total less MSE. Some methods of the PLR that have $L_1$-penalty regularization term, such as the lasso and the elastic net, can give a sparse solution. In statistical learning, sparsity is critical* *in terms of interpretation as well as selection of the effective features in the regression model. In this thesis, a novel method called as the fixed-shape elastic net will be introduced. It supersedes the limitations of the ordinary elastic net by exploiting all the combinations of the $L_1$-norm and $L_2$-norm. Moreover, in this thesis, another novel family of the regularization terms will be introduced called as the exponential norms. Specifically, an extensive study will be conducted on the $L_1$-exponential norm, and its applicability in the PLR. The coordinate descent algorithm will be designed to solve for the fixed-shape elastic net and the $L_1$-exponential norm. Numerical examples and simulation studies will be presented to highlight the performance of the novel methods.*

***Keywords:*** *Penalized linear regression; Regularization; Variable selection; Ordinary least squares; Statistical learning; Sparsity; Ridge regression; Lasso; Elastic net; Fixed-shape elastic net; Exponential norm.*

# ملخص الرسالة

**الاسم الكامل:** عبدالرحمن محي الدين محمود خان

**عنوان الرسالة:** تنظيم واختيار المتغيرات عبر الشبكة المرنة ذات الشكل الثابت والمعيار الأسّي الطبيعي

**التخصص:** هندسة نظم صناعية

**تاريخ الدرجة العلمية:** مايو 2015

الانحدار الخطّي المُغرَّم (ا.خ.م) هو أحد الأدوات التي توفر تنظيم واختيار المتغيرات بالنسبة للمعاملات المقدّرة لنموذج الانحدار، ا.خ.م يتكوّن من دالة الخسارة مثل المربعات الصغرى المألوفة (م.ص.م) بالإضافة لحدّ التنظيم المُغرَّم، م.ص.م بالكاد يكون نافعا في تفسير البيانات الحقيقية، معدل خطأ المربعات (م.خ.م) - الذي يتكوّن من مجموع خطأ التباين وخطأ الانحياز المربع – من الممكن أن يكون عاليا لـ (م.ص.م) بالرغم من عدم وجود خطأ الانحياز المربع، الانحدار الخطي المُغرَّم مثل انحدار القمة (رِدْج)، وحبل الصيد (لاسو)، والشبكة المرنة (إلاستك نت) تضيف خطأ الانحياز المربع والتي ربما تقلل خطأ التباين، وبذلك، تقلل محصلة (م.خ.م)، بعض طرق ا.خ.م التي فيها حد تنظيم غرامة-ل1 مثل حبل الصيد والشبكة المرنة تستطيع أن تعطي حلا متناثرا، في التعلم الإحصائي؛ التناثر مهم من نواحي التفسير وكذلك اختيار المزايا المؤثرة فقط في نموذج الانحدار، في هذه الرسالة؛ ستُقدّم طريقة جديدة تسمى الشبكة المرنة ذات الشكل الثابت، تتجاوز قيود الشبكة المرنة المألوفة باستغلال جميع أمزجة معيار-ل1 ومعيار-ل2، أيضا؛ ستُقدّم أسرة جديدة من حدود التنظيم تُسمى المعايير الأسّية الطبيعية (إكسبونّنْشَل نورمز)، بشكل خاص، سوف تُجرى دراسة موسعة لمعيار الأس الطبيعي-ل1 وتطبيقه في ا.خ.م، خوارزم التناقص الإحداثي سيصمم لحل للشبكة المرنة ذات الشكل الثابت ومعيار الأس الطبيعي-ل1، أمثلة ودراسات محاكاة ستُقدم لتسلط الضوء على أداء الطريقتين الجديدتين.

**الكلمات المفتاحية:** الانحدار الخطي المُغرَّم، تنظيم، اختيار المتغيرات، المربعات الصغرى المألوفة، التعلم الإحصائي، التناثر، انحدار القمة، حبل الصيد، الشبكة المرنة، الشبكة المرنة ذات الشكل الثابت، معيار الأس الطبيعي.

# CHAPTER 1

# INTRODUCTION

In the $14^{th}$ century, the philosopher William Ockham proposed the law of parsimony [1]. The law states that "plurality should not be posited without necessity." In other words, "entities are not to be multiplied beyond necessity." The necessity of such a law is becoming obvious every day, especially in this era of various applications with large amounts of data and high computing and storing capacities.

An active field of research "statistical learning with sparsity" [2] applies Ockham's law, also called as Ockham's razor. Statistical Learning consists of a set of tools that facilitates understanding a certain dataset [3]. It overlaps with machine learning in computer science. Some of these tools are regression, classification, and clustering. Moreover, statistical learning offers some sparse methods that simplify a model expression by selecting some parameters or features and eliminating others, by zeroing them. Thus, statistical learning with sparsity offers a more explainable model expression than the non-sparse methods.

Regression is a tool that estimates the relationship between a single dependent variable, and one or more independent variables. There are many methods to estimate the coefficients of the independent variables (or predictors) in order to reflect that into the dependent variable (or response). In multiple linear regression, the Ordinary Least Square (OLS) method estimates the coefficients with the minimum residual sum of squares. Let $\mathbf{y} \in \mathbb{R}^N$ be a vector with $N$ observations containing the response, and let $\mathbf{X} \in \mathbb{R}^{N \times p}$ be a matrix of $p$ predictors for the $N$ observations. Let $\mathbf{x}_j \in \mathbb{R}^N$ be the $j^{th}$ column of $\mathbf{X}$ $\forall$ $j = 1, \ldots, p$. Let $\boldsymbol{\beta} \in \mathbb{R}^p$ be a vector of unknown coefficients, and let $\boldsymbol{\varepsilon} \in \mathbb{R}^N$ be a vector of errors. The linear regression model can be written as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \beta_I \mathbf{1}_N + \boldsymbol{\varepsilon}, \tag{1.1}$$

where $\mathbf{1}_N \in \mathbb{R}^N$ is an all-one vector, and $\beta_I$ is the intercept or the bias term.

## 1.1 Data Standardization

Elements of $\mathbf{X}$ are typically standardized before solving the regression model. This is done in order to have solutions that are independent from different measured units (like gram, kilogram or pound). Moreover, elements of $\mathbf{y}$ are centered to make $\beta_I = 0$. By standardizing $\mathbf{X}$ and centering $\mathbf{y}$, one dimension of the model is reduced (i.e., from $p+1$ to $p$), and later it will be seen that this reduction is convenient to solve Penalized Linear Regression (PLR) models. Standardization and centering can be done as follows:

$$x_{ij,std} = \frac{x_{ij} - \bar{x}_j}{\sigma_{\mathbf{x}_j}}, \quad \forall\, i = 1, \ldots, N \;\&\; j = 1, \ldots, p, \tag{1.2}$$

$$y_{i,std} = y_i - \bar{y} \quad \forall\, i = 1, \ldots, N, \tag{1.3}$$

where $x_{ij,std}$ denotes the standardized $i^{th}$ element of $\mathbf{x}_j$, and $x_{ij}$ denotes the original $i^{th}$ element of $\mathbf{x}_j$. $\bar{x}_j$ denotes the mean and $\sigma_{\mathbf{x}_j}$ denotes the standard deviation of the elements of $\mathbf{x}_j$. $y_{i,std}$ denotes the centered $i^{th}$ element of $\mathbf{y}$, $\bar{y}$ denotes the mean of the elements of $\mathbf{y}$, and $y_i$ denotes the original $i^{th}$ element of $\mathbf{y}$. The standardization will result the following:

$$\frac{1}{N}\sum_{i=1}^{N} x_{ij} = 0, \qquad \frac{1}{N}\sum_{i=1}^{N} x_{ij}^2 = 1, \qquad \frac{1}{N}\sum_{i=1}^{N} y_i = 0 \qquad \forall\, j = 1, 2, \ldots p. \tag{1.4}$$

Let $\hat{\beta}_j$ be a coeffecient estimate for predictor $j$. Let $\hat{\beta}_{j,std}$ be a standardaized coeffecient estimate for predictor $j$. The transformed linear regression model and the original linear regression model will be related as:

$$\hat{\beta}_j = \frac{\hat{\beta}_{j,std}}{\sigma_{\mathbf{x}_j}} \quad \forall\, j \tag{1.5}$$

$$\hat{\beta}_I = \bar{y} - \sum_{j=1}^{p} \hat{\beta}_j \bar{x}_j. \tag{1.6}$$

In the following part of the thesis, all coeffecient estimates $\hat{\beta}_j$ are assumed to be standardized, (i.e., $\mathbf{X}$ is standardized & $\mathbf{y}$ is centered).

Let $\hat{\boldsymbol{\beta}}_{\boldsymbol{o}}$ be a vector of coeffecient estimates found by OLS method. In order to linearly fit the readings, OLS solves the model by minimizing the sum of squared errors, described as follows:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min}\, S(\boldsymbol{\beta}) = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min}\, \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}), \tag{1.7}$$

where $S(\boldsymbol{\beta})$ denotes the objective function of OLS method. Consider the hessian of the above function:

$$\boldsymbol{\nabla}^2 S(\boldsymbol{\beta}) = \mathbf{X}^T\mathbf{X}, \tag{1.8}$$

the hessian matrix is always positive semi-definite, hence the function is convex and the local optimum is the global optimum. Since the objective function is convex, the following should be true at optimality:

$$\boldsymbol{\nabla} S(\boldsymbol{\beta}) = \mathbf{0}, \tag{1.9}$$

$$-\mathbf{X}^T\mathbf{y} + \mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = \mathbf{0}, \tag{1.10}$$

$$\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\mathbf{y}. \tag{1.11}$$

If $(\mathbf{X}^T\mathbf{X})$ is invertible, then OLS coefficient estimates will be:

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{o}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{1.12}$$

## 1.2 OLS Drawbacks

There are two criteria that measure the goodness of any regression method: Prediction accuracy and Interpretability. Prediction accuracy is measured in terms of Mean Squared Error (MSE). A low value of MSE indicates higher prediction accuracy. MSE consists of two types of error: variance error and bias-squared error. Interpretability is inversely proportional to the number of nonzero coefficients. The less the number of nonzero coefficient estimates, the easier to interpret the model. For a dataset that has a high number of predictors, setting some of the coefficient estimates to zero will make the model more interpretable. This parsimonious characteristic, i.e. setting some coefficient estimates to zero, is called “sparsity”.

Unfortunately, OLS behaves poorly in terms of prediction accuracy and/or interpretability. It is true that bias-squared error in OLS is zero, but variance error can potentially be high with more predictors and observations. The unbiased characteristic of OLS does not allow us to reduce MSE by trading off between bias-squared error and variance error. The uniqueness of the solution of OLS for $N > p$ case does not provide flexibility in finding other alternatives.

Moreover, OLS does not give a sparse solution, which can make the model very hard to interpret.

In order to overcome the above issues, biased estimates are used. Some of the well-known biased estimate methods are the ridge regression [4], the Least Absolute Shrinkage and Selection Operator (LASSO) [5], and the elastic net [6] methods. By compromising between bias-squared error and the variance error, the total prediction error can be reduced. Moreover, by reducing some estimates to zero, the model can be more interpretable. There is no method that can generally perform better than other methods for a given dataset. Therefore, we have to compare between many regression methods for a given dataset, and based on the two critical criteria, we might decide which method performs better.

## 1.3 Thesis Organization

Chapter 2 will present a literature review on the regression methods, like biased PLR, the ridge regression, lasso, and elastic net methods. A special case of the orthogonal data will also be used to analyze the PLR, ridge regression, lasso and elastic net. The last section of Chapter 2 will demonstrate the optimal selection of the tuning parameters for the PLR.

Chapter 3 will illustrate the methods to solve the PLR by the coordinate descent algorithm in detail for the ridge regression, lasso, and elastic net with different types of updates. Chapter 4 will introduce a novel method called as the fixed-shape elastic net, which is a modification of the elastic net that utilizes the full capacity of the elastic net. Chapter 5 will introduce a novel family of regularization terms for the PLR, called as the exponential norms.

Chapter 6 will depict the performance of the proposed methods and compare with the existing methods. A numerical study will be carried out to compare the prediction accuracy of the proposed and existing methods. The discussion and conclusion will be presented in Chapter 7.

## CHAPTER 2

# LITERATURE REVIEW

There are numerous biased estimate methods in the literature for multiple linear regression. One class of the biased estimate methods are known as PLR methods. They have a general formulation that can be written as:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min}\, f(\boldsymbol{\beta}) = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min}\, S(\boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta}), \tag{2.1}$$

where $P(\boldsymbol{\beta})$ is denoted as the penalized regularization term, and $f(\boldsymbol{\beta})$ is denoted as the unconstrained objective function of the PLR. $\lambda$ denotes the penalty parameter. The constrained form can be written as follows:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min}\, S(\boldsymbol{\beta}) \qquad s.t. \quad g(\boldsymbol{\beta}) = P(\boldsymbol{\beta}) - \theta \leq 0, \tag{2.2}$$

where $g(\boldsymbol{\beta})$ is the constraint set, and $\theta$ is any constant term. Next, well-known PLR methods will be presented.

## 2.1 The Ridge Regression

Typically, the OLS estimate, $\hat{\boldsymbol{\beta}}_{\boldsymbol{o}}$ gives unsatisfactory MSE in multiple linear regression with high number of predictors. To overcome this issue, Hoerl and Kennard [4] proposed the ridge regression method which deals with the high correlation among the predictors, i.e., when $\mathbf{X}^T\mathbf{X} \neq N\mathbf{I}_p$, and very ill-conditioned. The idea can be mathematically written as:

$$\hat{\boldsymbol{\beta}}_{ridge} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^T\mathbf{y}, \tag{2.3}$$

or alternatively:

$$\hat{\boldsymbol{\beta}}_{ridge} = (\mathbf{I} + \lambda(\mathbf{X}^T\mathbf{X})^{-1})^{-1}\hat{\boldsymbol{\beta}}_{\boldsymbol{o}}, \tag{2.4}$$

where $\lambda \geq 0$, and $\hat{\boldsymbol{\beta}}_{ridge}$ denotes the vector of coefficient estimates found by the ridge regression. The range of $\hat{\boldsymbol{\beta}}_{ridge}$ will be from OLS estimate $\hat{\boldsymbol{\beta}}_{\boldsymbol{o}}$ until $\hat{\boldsymbol{\beta}}_{ridge} = \mathbf{0}$. When $\lambda \geq 0$, a bias-squared error is added to MSE. However, when bias-squared error is added, the variance error may reduce as may the MSE. By shrinking coefficient estimates, the ridge regression trades off between bias-squared and variance errors to get coefficient estimates that provide the minimum MSE.

Ridge regression can be stated as a constrained or penalized Non-Linear Programming (NLP) problem, described as:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min}\ \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) \qquad s.t.\ \ \frac{1}{2}\sum_{j=1}^{p}\beta_j^2 \leq \frac{1}{2}t^2, \tag{2.5}$$

where $t \geq 0$ is a tuning parameter, such that when $t = 0$, $\hat{\beta}_{j,ridge}$ will be zero $\forall\ j$, where $\hat{\beta}_{j,ridge}$ denotes the $j^{th}$ element of $\hat{\boldsymbol{\beta}}_{ridge}$. When $t \geq t_o$ where $t_o^2 = \sum_{j=1}^{p}\hat{\beta}_{oj}^2$, then the constraint will be redundant and $\hat{\beta}_{j,ridge}$ will be the same as $\hat{\beta}_{oj}\ \forall\ j$ (See Figure (2.1)), where $\hat{\beta}_{oj}$ denotes the $j^{th}$ element of $\hat{\boldsymbol{\beta}}_{\boldsymbol{o}}$. The constrained NLP problem can be equivalently formulated in a penalized form as follows:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min}\quad \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) + \frac{\lambda}{2}\sum_{j=1}^{p}\beta_j^2, \tag{2.6}$$

where $\lambda \geq 0$ is a penalty parameter, such that when $\lambda \to \infty$, $\hat{\beta}_{j,ridge}$ will be zero $\forall\ j$. When $\lambda = 0$, $\hat{\beta}_{j,ridge}$ will be the same as $\hat{\beta}_{oj}\ \forall\ j$. For every value of $\lambda$, there exists a constant $t$ such that both formulations will result in the same solution. However, no direct mapping exists between $\lambda$ and $t$ for the PLR [7].

Although the ridge regression performs well when there is a high correlation between predictors, it has a drawback that it lacks interpretability (by not providing a sparse solution). In other words, it may excel in only one of the two criteria that measure the goodness of any regression method, i.e. prediction accuracy. In fact, when $t \to 0$, the squared norm tends to equalize the coefficient estimates, which makes it more difficult to interpret.

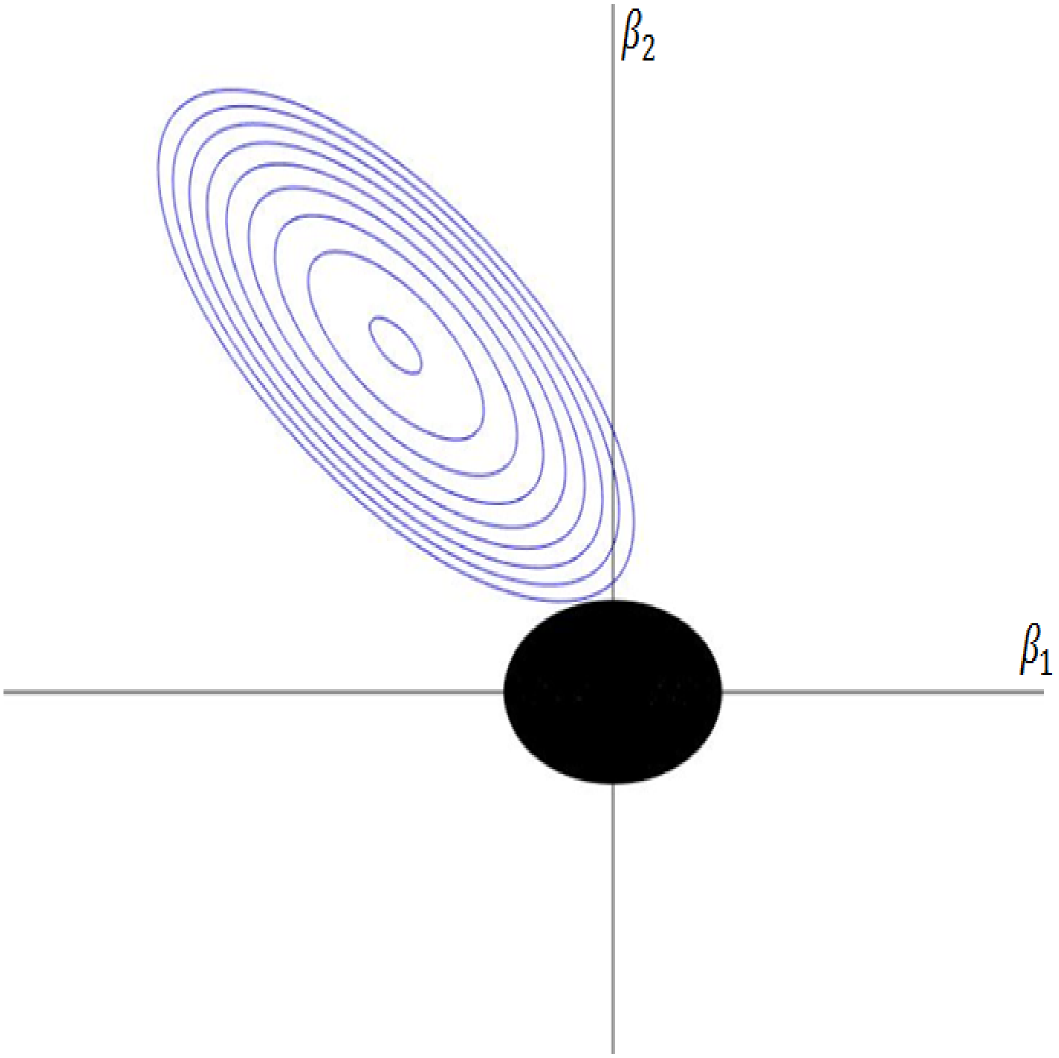

Figure 2.1: For two predictors, the elliptical contours are the objective function of the OLS (the center of the contours is the optimal solution of OLS) . The shaded area is the norm constraint of the ridge regression.

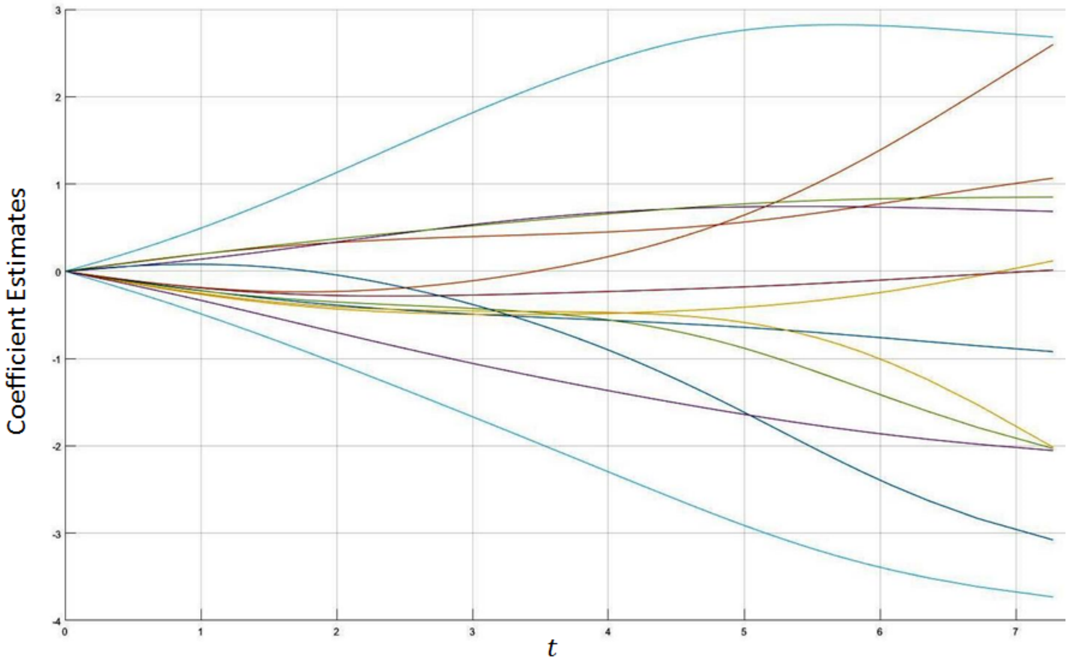

Figure 2.2: Trace plot of Boston dataset by the ridge regression. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

The PLR provides a range of solutions for $t \in [0, t_o]$ that can be visually summarized in a trace plot (See Figure (2.2)). There are no zero coefficient estimates for $t > 0$ except at one passing point when some coefficient estimate changes the sign. After solving for the whole range of $\lambda$ or $t$, the optimal $\lambda^*$ or $t^*$ will be selected. The selection of the optimal tuning parameter $\lambda^*$ or $t^*$ will be discussed in detail in Section 2.6.

## 2.2 The Lasso

The lasso method was proposed by Tibshiriani [5] to overcome the drawback of the interpretability of the ridge regression, and the drawback of the high prediction error of the subset selection. Breiman [8] pointed that the subset selection has a high variability because it is a discrete process. It uses a hard thresholding approach for sparsity. On the other hand, the lasso exploits a flexibility of shrinking the coefficient estimates by using a soft thresholding approach. The similarity between the lasso and the ridge regression is that both have the same objective function of OLS. However, instead of using $L_2$-norm as a constraint, the lasso uses $L_1$-norm (See Figure (2.3)). To find the lasso estimate $\hat{\boldsymbol{\beta}}_{lasso}$, the problem can be formulated as:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min}\, \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) \qquad s.t. \quad \sum_{j=1}^{p}|\beta_j| \leq\ t, \tag{2.7}$$

where $t \geq 0$ is a tuning parameter, such that when $t = 0$, all $\hat{\beta}_{j,lasso}$ will be zero $\forall\ j$, where $\hat{\beta}_{j,lasso}$ denotes the $j^{th}$ element of $\hat{\boldsymbol{\beta}}_{lasso}$. When $t \geq t_o$, where $t_o = \sum_{j=1}^{p}|\hat{\beta}_{oj}|$, then the constraint will be redundant, and $\hat{\beta}_{j,lasso}$ will be the same as $\hat{\beta}_{oj}$ $\forall\ j$. The constrained NLP problem can be equivalently formulated in a penalized form as follows:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min}\, \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) + \lambda\sum_{j=1}^{p}|\beta_j|, \tag{2.8}$$

where $\lambda > 0$, is the penalty parameter, such that when $\lambda \to \infty$, $\hat{\beta}_{j,lasso}$ will be zero $\forall\, j$, and when $\lambda = 0$, $\hat{\beta}_{j,lasso}$ will be the same as $\hat{\beta}_{oj}$ $\forall\, j$. In addition to that, for every value of $\lambda$, there exists a constant $t$ that both the formulations will give the same solution.

Figure (2.3) geometrically shows the reason that the lasso can set some estimates to 0. Due to the fact that the corners of the norm are not smooth, and usually exposed to the objective function, some of the coefficient estimates are set to 0. For example, the trace plot of the lasso in Figure (2.4) indicates, how the coefficient estimates are set to zero at some $t > 0$.

### 2.2.1 Comparing the Lasso and the Non-negative Garrote

Tibshiriani [5] compared the lasso with the non-negative garrote method [9]. To find the non-negative garrote estimate $\hat{\boldsymbol{\beta}}_{non}$, the problem can be formulated as:

$$\underset{\boldsymbol{u}\in\mathbb{R}^p}{min} \frac{1}{2}\sum_{i=1}^{N}(y + \sum_{j=1}^{p} u_j\hat{\beta}_{oj})^2 + \lambda\sum_{j=1}^{p} u_j, \qquad s.t. \quad u_j \geq 0, \qquad \forall j. \tag{2.9}$$

Let $\hat{\beta}_{j,non}$ denotes the $j^{th}$ element of non-negative garrote coefficient estimate $\hat{\boldsymbol{\beta}}_{non}$, then it can be shown that $\hat{\beta}_{j,non} = u_j\hat{\beta}_{oj}$.

The drawback of the non-negative garrote method is that it is directly affected by

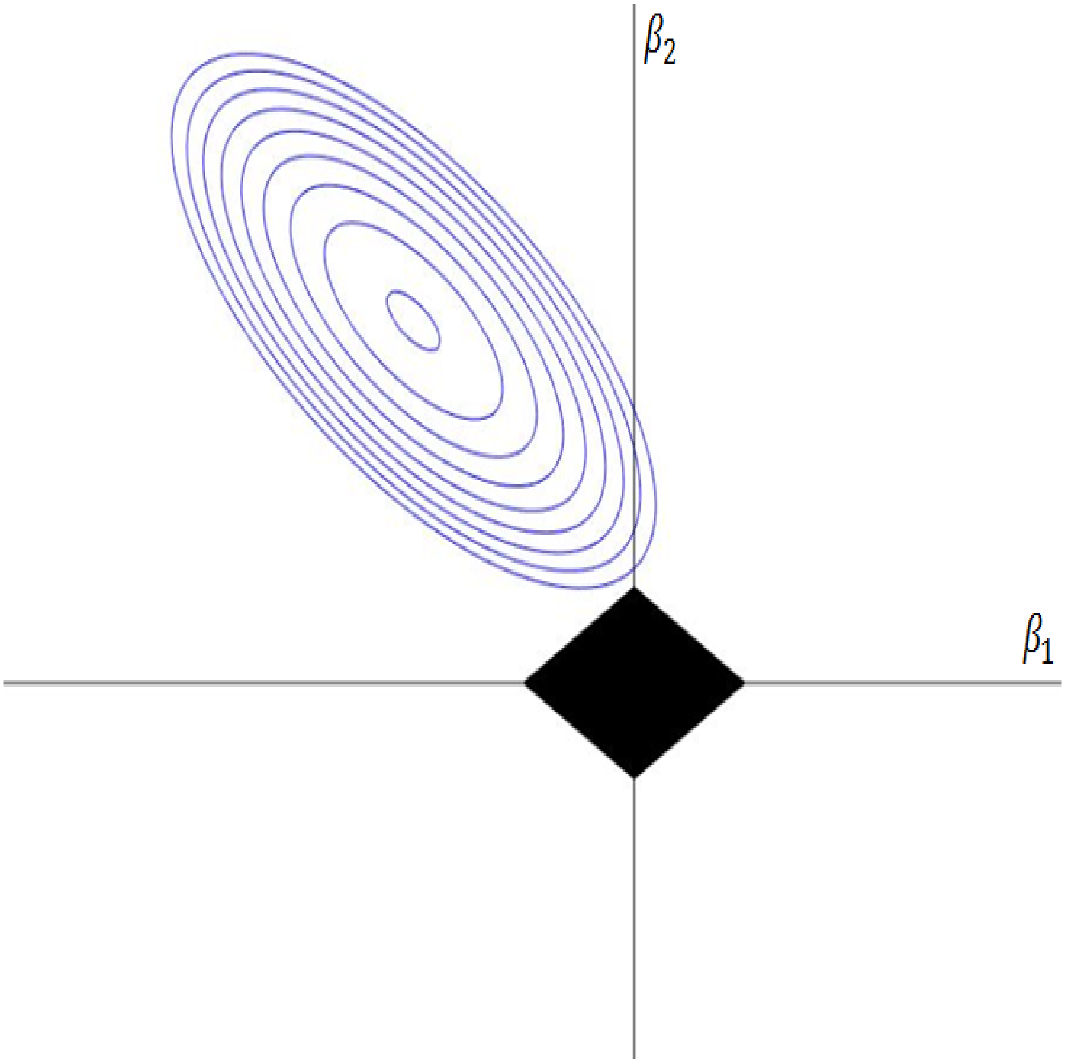

Figure 2.3: For two predictors, the elliptical contours are the objective function of the OLS (the center of the contours is the optimal solution of OLS) . The shaded area is the norm constraint of the lasso.

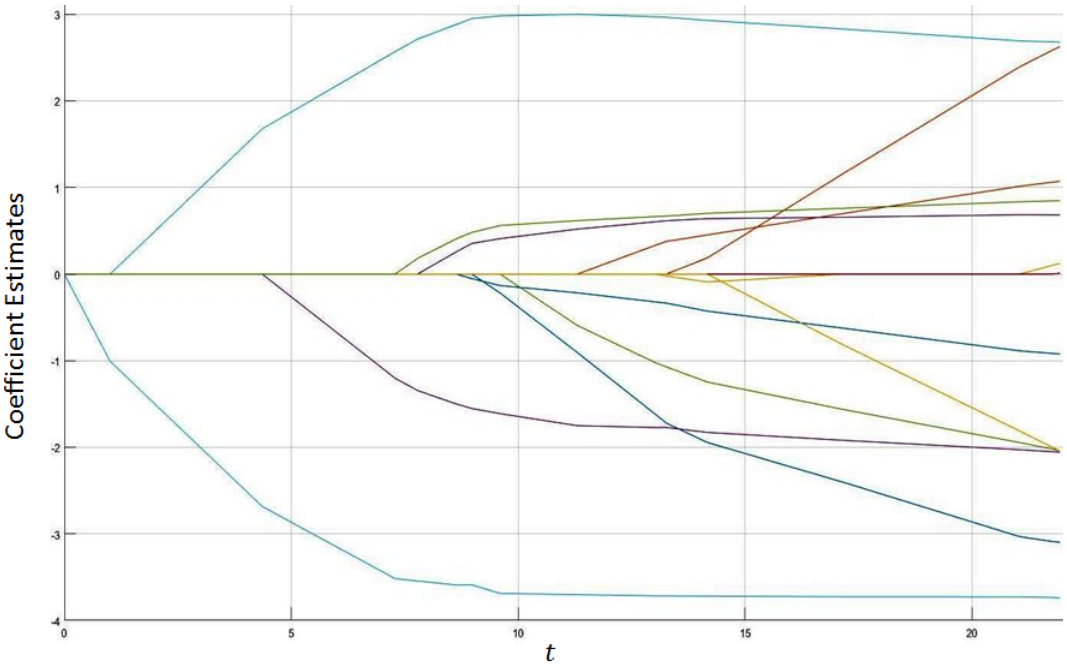

Figure 2.4: Trace plot of Boston dataset by the lasso. The x-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

the sign and magnitude of OLS estimates. When $\hat{\beta}_{oj}$ is large the shrinkage will be less. For example, the diamond in Figure (2.3) would be stretched towards the larger OLS coefficient estimate. The lasso method avoids the direct use of OLS estimates as in Equation (2.9).

### 2.2.2 Comparing the Lasso and the Ridge Regression

With high correlated predictors, the lasso tends to randomly select one variable from one group and ignore the other variables. In such cases, the ridge regression usually performs better than the lasso by a high margin. Another limitation of the lasso is for $p \gg N$ case, where the lasso cannot select more than $N$ nonzero coefficient estimates. The ridge regression, on the other hand, will have $p$ nonzero coefficient estimates for $p \gg N$ case.

### 2.2.3 Generalization with the Bridge Regression

Frank and Friedman [10] introduced the bridge regression, which can be considered as a generalization to both the lasso and ridge regression. If we add a $\lambda \sum_{j=1}^{p} |\beta_j|^q$ term to the OLS objective function, then the resultant formulation will be called as the bridge regression. When $q = 1$, the bridge regression boils down to the lasso, and when $q = 2$, the bridge regression boils down to the ridge regression. Figure (2.5) shows a variety of different $L_q$-norms. The lasso has the minimum norm in which the feasible set is convex. When the bridge regression has a strictly convex norm, i.e. $q > 1$, the coefficient estimates will not be set to 0 (proved by Fan and Li [11]). Thus, these norms are not good alternatives for the lasso in terms of sparsity.

## 2.3 The Elastic Net

Zou and Hastie [6] proposed the elastic net method. It has a regularization term that is a weighted combination between both $L_1$-norm and $L_2$-norm. In the same paper, Zou and Hastie also introduced what they called as the naive elastic net.

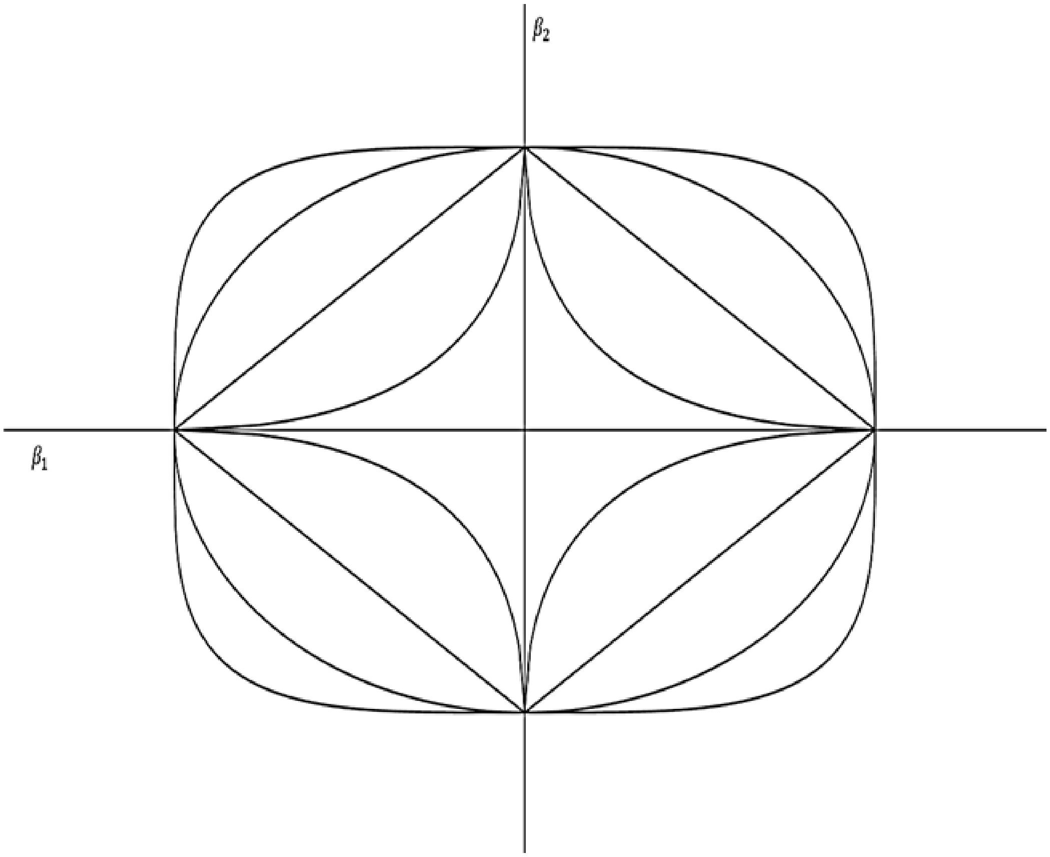

Figure 2.5: For two predictors, four different norms of the bridge regression with $q = 0.5$, 1, 2, and 4, respectively, starting from inside.

To find the naive elastic net estimate, $\hat{\boldsymbol{\beta}}_{nnet}$, the problem can be formulated as:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min}\,\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) \qquad s.t. \quad \sum_{j=1}^{p}\left(\alpha|\beta_j|+\frac{1-\alpha}{2}\beta_j^2\right) \leq \; \alpha t+\frac{1-\alpha}{2}t^2, \tag{2.10}$$

where $t \geq 0$ controls the size of the norm, such that the maximum value any $|\hat{\beta}_{j,nnet}|$ can reach is $t$, where $\hat{\beta}_{j,nnet}$ denotes the $j^{th}$ element of $\hat{\boldsymbol{\beta}}_{nnet}$. The parameter $\alpha \in [0,1]$ denotes the weight of the combination between $L_1$-norm and $L_2$-norm, such that if $\alpha = 1$, then the problem will be the same as the lasso; and if $\alpha = 0$, then the problem will be the same as the ridge regression. In a similar manner to the lasso and ridge regression, let $t_o$ be the minimum value, such that the above constraint is redundant. if $t \geq t_o$, then the constraint will be similar to solving OLS problem. If $t = 0$, then all coefficient estimates will be set to 0.

The equivalent penalized form can be written as:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min}\,\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+\lambda\sum_{j=1}^{p}\left(\alpha|\beta_j|+\frac{1-\alpha}{2}\beta_j^2\right), \tag{2.11}$$

where $\lambda \geq 0$ is the penalty parameter. If $\lambda = 0$, then the problem will be similar to OLS problem. If $\lambda \to \infty$, then all coefficient estimates will be set to 0.

The elastic net is just a rescaling of the naive elastic net to avoid the double amount of shrinkage, since the method uses two different norms. More detail will be provided in Section 2.5.

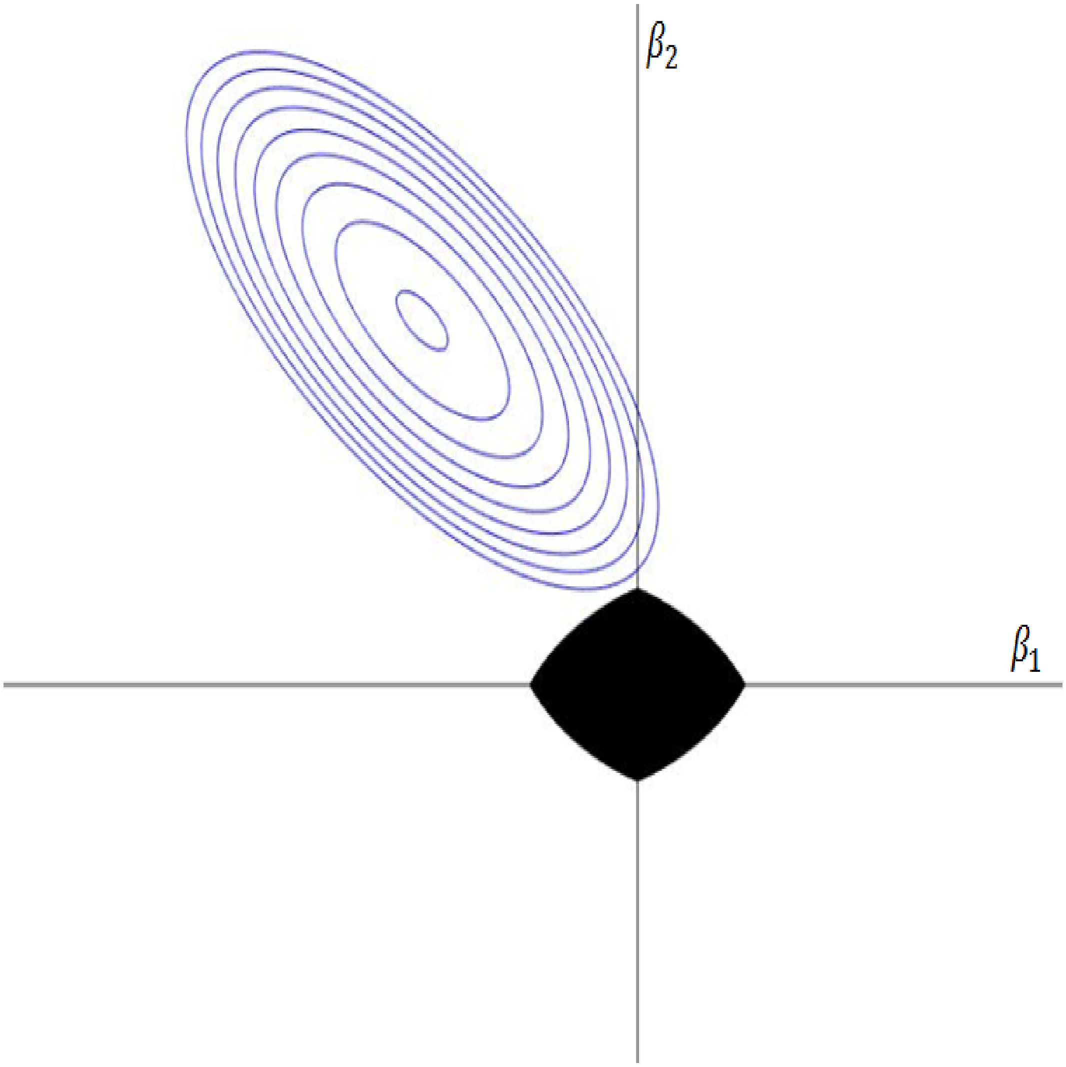

Figure 2.6: For two predictors, the elliptical contours are the objective function of the OLS (the center of the contours is the optimal solution of OLS) . The shaded area is the norm constraint of the elastic net.

The elastic net is always convex, since it is a non-negative weighted sum of two convex functions (See Figure (2.6)). The elastic net overcomes the issue of high correlated variables that cannot be handled by the lasso. In addition to that, it can do variable selection that cannot be performed by the ridge regression. The elastic net has another advantage over the lasso for $p \gg N$ case, i.e., it overcomes the limitation of the lasso, and can have more than $N$ nonzero coefficient estimates.

## 2.4 Other Generalizations of The PLR

There are other generalizations of the PLR which incorporates the $L_1$-penalty in different forms. For example, Yuan and Lin [12] introduced the group lasso where there are $\rho$ groups; $G_\tau$, for $\tau = 1, 2, \ldots, \rho$, such that $G_\tau = \sum_{j \in G_\tau} \sqrt{\beta_j^2}$. The penalized term is $\lambda \sum_{\tau=1}^{\rho} G_\tau$, such that when $\rho = p$, the problem boils down to the lasso method. The group lasso method lets coefficient estimates under one group go simultaneously to zero. Puig et al. [13] and Simon et al. [14] extended the group lasso by adding a weighted lasso penalty as follows: $\lambda\big(\sum_{\tau=1}^{\rho} (1-\alpha) G_\tau + \sum_{j=1}^{p} \alpha|\beta_j|\big)$. This method is called the sparse-group lasso which allows sparsity within the group. Jacob et al. [15] introduced the overlap group lasso by allowing coefficient estimates to be in more than one group.

Another generalization by Tibshiriani et al. [16] is the fused lasso where it has two constraints: the first is the ordinary lasso constraint $\sum_{j=1}^{p} |\beta_j| \leq t_1$ and the second is the difference between two successive coefficient estimates $\sum_{j=2}^{p} |\beta_j - \beta_{j-1}| \leq t_2$. The first constraint as it is known encourages sparsity for some coefficient estimates, and the second one encourages sparsity of some differences between two coefficient estimates.

Zou [17] introduced the adaptive lasso, which can have a sparser characteristic than the ordinary lasso, where the penalty is $\lambda \sum_{j=1}^{p} w_j |\beta_j|$ and $w_j$ is some weight that can be selected as $w_j = |\hat{\beta}_{oj}|^{-v}$, $v > 0$. The non-negative garrote can be considered as a specific case of the adaptive lasso. There are other methods for the PLR that have a nonconvex penalty, such as Smoothly Clipped Absolute Deviation (SCAD) by Fan and Li [11], Minimax Concave (MC+) by Zhang [18], and Seamless $L-0$-Penalty (SELO) by Dicker et al. [19].

Meinshausen [20] introduced the relaxed lasso method. It is a two-stage method: the first stage is solving the problem as the ordinary lasso. The second stage is re-solving only for the non-zero coefficient estimates as an OLS problem. This will help in removing the shrinkage effect on the nonzero coefficient estimates, which might improve the prediction accuracy, since nonzero coefficient estimates are usually shrunk. This relaxation can be applied to any shrinking sparse method like the elastic net and the novel proposed methods (see Chapter 4 & 5).

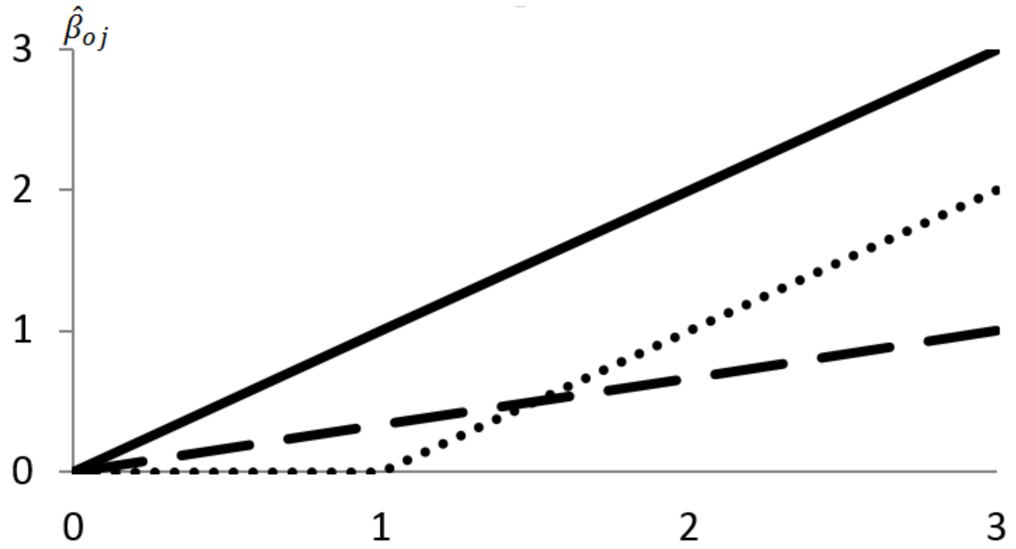

Figure 2.7: The $x$-axis is for the OLS estimate for orthogonal case. The continuous line is for OLS, the dashed line is for ridge, and the dot-line is for the lasso.

There are other generalizations related to the objective function like the LAD-lasso by Wang et al. [21] which uses $L_1$-norm loss function instead of $L_2$-norm. LAD stands for "Least Absolute Deviation". It is less sensitive to the outliers.

## 2.5 Orthogonal Case

A special case of the PLR is when the predictors are orthogonal (See Figure (2.7)), i.e. $\mathbf{X}^T\mathbf{X} = N\mathbf{I}_p$. The OLS coefficient estimates, in this case, can be written as:

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{o}} = \frac{1}{N}\mathbf{X}^T\mathbf{y} \tag{2.12}$$

The PLR for the orthogonal case can be written as:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{argmin}\Big\{S(\boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta})\Big\}, \tag{2.13}$$

$$= \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{argmin}\Big\{\frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta})\Big\} \tag{2.14}$$

$$= \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{argmin}\Big\{\frac{1}{2}\mathbf{y}^T\mathbf{y} - \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{y} + \frac{1}{2}\boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} + \lambda P(\boldsymbol{\beta})\Big\} \tag{2.15}$$

From the above, the orthogonal case, every $\beta_j\ \forall\ j = 1, 2, \ldots, p$ when $\beta_j \neq 0$ is separable, and can be solved independently. In general, the solution can be obtained as:

$$\frac{\partial}{\partial\beta_j}\Big(S(\boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta})\Big)\Big|_{\beta_j=\hat{\beta}_j} = 0 \tag{2.16}$$

$$-N\hat{\beta}_{oj} + N\hat{\beta}_j + \lambda\ \frac{\partial P(\boldsymbol{\beta})}{\partial\beta_j}\Big|_{\beta_j=\hat{\beta}_j} = 0 \tag{2.17}$$

$$\hat{\beta}_j = \hat{\beta}_{oj} - \frac{\lambda}{N}\ \frac{\partial P(\boldsymbol{\beta})}{\partial\beta_j}\Big|_{\beta_j=\hat{\beta}_j} \tag{2.18}$$

For the ridge regression, the solution will be:

$$\hat{\beta}_{j,ridge} = \hat{\beta}_{oj} - \frac{\lambda}{N}\hat{\beta}_{j,ridge} \tag{2.19}$$

$$\hat{\beta}_{j,ridge} = \frac{\hat{\beta}_{oj}}{1 + \frac{\lambda}{N}} \tag{2.20}$$

For the lasso, the solution will be:

$$\hat{\beta}_{j,lasso} = \hat{\beta}_{oj} - \frac{\lambda}{N}sign(\hat{\beta}_{j,lasso}) \tag{2.21}$$

Now, if $\hat{\beta}_{j,lasso} > 0$, then say $\hat{\beta}^{+}_{j,lasso}$ will be:

$$\hat{\beta}^{+}_{j,lasso} = \hat{\beta}_{oj} - \frac{\lambda}{N}. \tag{2.22}$$

Note that, in order to have $\hat{\beta}_{j,lasso} > 0$, then $\hat{\beta}_{oj} > \frac{\lambda}{N}$. And if $\hat{\beta}_{j,lasso} < 0$, then say $\hat{\beta}^{-}_{j,lasso}$ will be:

$$\hat{\beta}^{-}_{j,lasso} = \hat{\beta}_{oj} + \frac{\lambda}{N}. \tag{2.23}$$

Note that, in order to have $\hat{\beta}_{j,lasso} < 0$, then $\hat{\beta}_{oj} < \frac{-\lambda}{N}$. Otherwise, $\hat{\beta}_{j,lasso} = 0$. The general formula for all the cases will be:

$$\hat{\beta}_{j,lasso} = sign(\hat{\beta}_{oj})\left(|\hat{\beta}_{oj}| - \frac{\lambda}{N}\right)_{+}, \tag{2.24}$$

where $(\beta)_{+}$ denotes $max\{0, \beta\}$.

For a single predictor, the orthogonal case for the lasso becomes the exact soft thresholding proposed in Donoho and Johnson [22], and Donoho et al. [23] for signal recovery by de-noising the wavelet coefficients.

For the naive elastic net, the solution will be:

$$\hat{\beta}_{j,nnet} = \hat{\beta}_{oj} - \frac{\lambda\alpha}{N} sign(\hat{\beta}_{j,nnet}) - \frac{\lambda(1-\alpha)}{N}\hat{\beta}_{j,nnet} \tag{2.25}$$

Now, if $\hat{\beta}_{j,nnet} > 0$, then say $\hat{\beta}^{+}_{j,nnet}$ will be:

$$\hat{\beta}^{+}_{j,nnet} = \frac{\hat{\beta}_{oj} - \frac{\lambda\alpha}{N}}{1 + \frac{\lambda(1-\alpha)}{N}} \tag{2.26}$$

Note that, in order to have $\hat{\beta}_{j,nnet} > 0$, then $\hat{\beta}_{oj} > \frac{\lambda\alpha}{N}$. Now, if $\hat{\beta}_{j,nnet} < 0$, then say $\hat{\beta}^{-}_{j,nnet}$ will be:

$$\hat{\beta}^{-}_{j,nnet} = \frac{\hat{\beta}_{oj} + \frac{\lambda\alpha}{N}}{1 + \frac{\lambda(1-\alpha)}{N}} \tag{2.27}$$

Note that, in order to have $\hat{\beta}_{j,nnet} < 0$, then $\hat{\beta}_{oj} < \frac{\lambda\alpha}{N}$. Otherwise, $\hat{\beta}_{j,nnet} = 0$. The general formula for all the cases will be:

$$\hat{\beta}_{j,nnet} = \frac{sign(\hat{\beta}_{oj})\bigg(|\hat{\beta}_{oj}| - \frac{\lambda\alpha}{N}\bigg)_{+}}{1 + \frac{\lambda(1-\alpha)}{N}} \tag{2.28}$$

The elastic net estimate $\hat{\beta}_{j,net}$ is a rescaling of the naive elastic net to avoid "double shrinkage", which can be obtained as:

$$\hat{\beta}_{j,net} = \left(1 + (1-\alpha)\frac{\lambda}{N}\right)\hat{\beta}_{j,nnet} \qquad \forall\, j. \tag{2.29}$$

The orthogonal case gives us a good idea about the reason for which the lasso and elastic net have the sparse characteristic, while the ridge does not. The ridge regression is a rescaling of the OLS estimate, and thus the ridge regression will never be set estimate to 0 (unless the OLS estimate is equal to 0). The lasso and elastic net have a subtraction of the thresholds $\frac{\lambda}{N}$ and $\frac{\lambda\alpha}{N}$, respectively. For the lasso, if $|\hat{\beta}_{oj}| < \frac{\lambda}{N}$, then $\hat{\beta}_{j,lasso}$ will automatically be set to 0. For the elastic net, if $|\hat{\beta}_{oj}| < \frac{\lambda\alpha}{N}$, then, $\hat{\beta}_{j,net}$ will automatically be set to 0. Furthermore, the naive elastic net has the rescaling characteristic as the ridge regression.

## 2.6 Tuning Parameter Selection

The PLR can give a range of coefficient estimates, which vary from OLS estimates until all of them are set to 0. The choice of the values of tuning parameters $t$ or $\lambda$, and $\alpha$ for the elastic net, is based on certain values of the tuning parameters that can give the highest prediction accuracy, i.e. the minimum prediction error. Cross validation is conducted on the dataset to find the parameter value that corresponds to the minimum prediction error. It is an effective tool that was originally used for multiple linear regression.

The parameters can be searched through resampling methods, such as cross validation and bootstrap methods. K-fold cross validation is one of the popular procedures used for the parameter selection. The key idea of the K-fold cross validation can be described as: the dataset is divided randomly into almost equal $K$ folds. Fold $k$, where $k = 1, \dots, K$, will be once considered as a validation set and the remaining larger dataset (all folds except fold $k$) $K^{\sim k}$ will be considered as a training set. For a certain tuning parameter $t$ or $\lambda$, K-fold cross-validation is applied as follows:

1. Find the coefficient estimates for the training set $\hat{\beta}_{j,K^{\sim k}}$ $\forall\ j$.

2. Find the cross validation error mean for fold $k$ with $n_k$ observations,

$$CV_k = \frac{1}{n_k} \sum_{i \in k} \left( y_i - \sum_{j=1}^{p} x_{ij} \hat{\beta}_{j,K^{\sim k}} \right)^2 \tag{2.30}$$

3. Repeat steps 1 and 2 for all $K$ folds.

4. Find the cross validation error by taking the mean of all $CV_k$s as:

$$\widehat{CV} = \frac{1}{K} \sum_{k=1}^{K} CV_k \tag{2.31}$$

5. Find standard deviation of all cross validations:

$$\sigma_{CV} = \sqrt{\frac{1}{K} \sum_{k=1}^{K} (CV_k - \widehat{CV})^2} \tag{2.32}$$

6. Find the standard error:

$$\widehat{SE} = \frac{\sigma_{CV}}{\sqrt{K}} \tag{2.33}$$

7. Repeat all the above for each $t$ or $\lambda$ grids. In case of the elastic net, the cross validation process will be two dimensional, by fixing $\alpha$, say $\alpha = 0.1$, and cross validating all grids of $t$ or $\lambda$, then fixing $\alpha$ to another value, say $\alpha = 0.2$, and cross validating all grids of $t$ or $\lambda$ and continue so forth for all grid points of $\alpha$.

8. Choose the tuning parameter(s) that has the minimum value of CV, and solve for the whole dataset.

9. One standard error rule: one can choose parameters to have sparser and more robust [24] results within one standard error range as follows:

$$\widehat{CV} \leq CV \leq \widehat{CV} + \widehat{SE} \tag{2.34}$$

Sometimes, the performance of different methods, (e.g. lasso or ridge), is compared by their minimum cross validation errors ($CV$). On the other hand, the performance is measured by comparing the Mean Prediction Squared Error (MSPE). For the MSPE comparison, a testing dataset is required.

The minimum number of folds that can be obtained is $K = 2$, and the maximum is $K = N$, which is, in this case called, leave-one-out cross validation. If $K = 2$, then the bias error will be high for the total prediction error. If $K = N$, then the validation is unbiased, but the variance error will be high. The trade-off between bias-squared and variance errors is related to the choice of $K$. Typically, fivefold or tenfold cross validation is used ($K = 5$ or $K = 10$).

## CHAPTER 3

# SOLUTION ALGORITHM

## 3.1 Optimality Conditions

The OLS objective function is convex, and the regularization term $\sum_{j=1}^{p} |\beta_j|^q$ is convex when $q \geq 1$. Hence, the sublevel set constraint is convex [25] for $q \geq 1$. Therefore, lasso, ridge regression, and elastic net are convex optimization problems. Thus, the KKT conditions are necessary and sufficient. For the constrained case, when the problem is differentiable (like the ridge regression), the optimality condition can be written as:

$$\nabla S(\boldsymbol{\beta}) + \lambda \nabla P(\boldsymbol{\beta}) = \mathbf{0} \tag{3.1}$$

In case of non-differentiable constraint, the optimality condition can be written as [26]:

$$\mathbf{0} \in \nabla S(\boldsymbol{\beta}) + \lambda \partial P(\boldsymbol{\beta}), \tag{3.2}$$

where $\partial$ denotes for a subgradient, and $\mathbf{0}$ is a zero vector.

The primal feasibility condition can be written as:

$$g(\boldsymbol{\beta}) = P(\boldsymbol{\beta}) - \theta \leq 0, \tag{3.3}$$

where $\theta$ is any constant term (that has no variable $\beta_j$) and does not exist in the penalized form. For example, $\theta = \alpha t + \frac{1-\alpha}{2} t^2$ for the elastic net, and $\theta = t$ for the lasso. The dual feasibility condition can be written as:

$$\lambda \geq 0, \tag{3.4}$$

And the complimentary slackness can be written as,

$$\lambda \, g(\boldsymbol{\beta}) = 0. \tag{3.5}$$

## 3.2 Solution Approach

The regularization term $\sum_{j=1}^{p} |\beta_j|$ is non-smooth and non-differentiable. To equivalently express the constraint $\sum_{j=1}^{p} |\beta_j| \leq t$ as a linear form, it requires $2^p$ constraints, with a permutation between the signs of coefficient estimates. Solving a quadratic problem with $2^p$ linear constraints is not practical when $p$ is large.

Efficient algorithms were proposed to solve the lasso, such as, the "homotopy method" by Osborne et al. [27], and a closely related method called the "Least Angle Regression" (LARS) by Efron et al. [28]. Zou and Hastie [6] applied LARS on the elastic net, and called it as "LAR-EN". It can be observed that the LARS algorithm may not perform well for the large scale problem.

Fu [29] and Daubechies et al. [30] suggested the coordinate descent algorithm on the bridge regression. Kooij et al. [31] applied this algorithm to solve for the elastic net. The coordinate descent algorithm can be considered a very efficient algorithm and competitive with LARS. The focus will be on this algorithm for the rest of the thesis.

## 3.3 Coordinate Descent Algorithm

Consider the following optimization problem: $\min f(\boldsymbol{\beta}) \mid \beta \in \mathbb{R}^p$, where $f(\boldsymbol{\beta})$ is differentiable (like OLS and the ridge regression), the gradient descent algorithm can be expressed as:

$$\boldsymbol{\beta}^{r+1} = \boldsymbol{\beta}^r - \gamma \nabla f(\boldsymbol{\beta}^r), \tag{3.6}$$

for $0 < \gamma < \gamma_o$, for some $\gamma_o > 0$. Let $\mathbf{d} = (d_1, \ldots, d_p)^T$ be a non zero direction vector and let $F$ be the cone of improving directions at $\bar{\boldsymbol{\beta}}$, defined as:

$$F = \{\mathbf{d} : f(\bar{\boldsymbol{\beta}} + \gamma d) < f(\bar{\boldsymbol{\beta}}) \quad \forall \gamma \in (0, \gamma_o)\} \tag{3.7}$$

When the function $f()$ at any point in its domain can be approximated by first order representation, then the cone of improving direction can also be expressed as:

$$F = \{\mathbf{d} : \nabla f(\bar{\boldsymbol{\beta}})^T \mathbf{d} < \mathbf{0}\}. \tag{3.8}$$

The gradient descent algorithm, also known as the steepest descent algorithm, chooses $\mathbf{d} = -\nabla f(\bar{\boldsymbol{\beta}})$, which is one of the improving directions (since $\nabla f(\bar{\boldsymbol{\beta}})^T \mathbf{d} = \nabla f(\bar{\boldsymbol{\beta}})^T(-\nabla f(\bar{\boldsymbol{\beta}})) = -||\nabla f(\bar{\boldsymbol{\beta}})||^2 < 0)$.

The coordinate descent algorithm does not take $\mathbf{d}$ as the steepest descent direction, i.e. $\mathbf{d} \neq -\nabla f(\bar{\boldsymbol{\beta}})$. Rather it sets another $\mathbf{d} \in \mathbb{R}^p$, which still belongs to the cone of improving directions. The direction vector is cyclically selected for all the elements $j = 1, \ldots, p$, defined as: $\mathbf{d} = -\mathbf{U}_j \nabla f(\bar{\boldsymbol{\beta}})$, where $\mathbf{U}_j$ is a $(p \times p)$ matrix that contains all zero's except at the $(j \times j)$ element, which is equal to one. The direction vector is an improving direction, as shown below:

$$\nabla f(\bar{\boldsymbol{\beta}})^T \mathbf{d} = \nabla f(\bar{\boldsymbol{\beta}})^T(-\mathbf{U}_j \nabla f(\bar{\boldsymbol{\beta}})) = -[\nabla f(\bar{\boldsymbol{\beta}})]_j{}^2 < 0, \tag{3.9}$$

where $[\nabla f(\bar{\boldsymbol{\beta}})]_j$ is the $j^{th}$ element of $\nabla f(\bar{\boldsymbol{\beta}})$. The coordinate descent algorithm is a cyclic iterative algorithm that minimizes the function with respect to one variable, and considers the remaining variables as constants. Then, it does this to every variable in a cyclic order until convergence. Although, this procedure is very simple, its efficiency in practice is proven to be very high for such PLR problems

[32]. The update rule for the coordinate descent algorithm can be expressed as:

$$\boldsymbol{\beta}^{r+1} = \boldsymbol{\beta}^{r} - \gamma \mathbf{U}_j \nabla f(\boldsymbol{\beta}^{r}). \tag{3.10}$$

where $r$ is the iteration number, and $\gamma \in (0, \gamma_0)$. For cycles of iterations $j = 1, \ldots, p$ respectively, the algorithm will converge to the global minimum. Alternatively, the coordinate descent algorithm can be written as:

$$\beta_1^{r+1} = \underset{\beta_1}{argmin}\left\{ f(\beta_1, \beta_2^r, \ldots, \beta_p^r) \right\} \tag{3.11}$$

$$\beta_2^{r+1} = \underset{\beta_2}{argmin}\left\{ f(\beta_1^{r+1}, \beta_2, \ldots, \beta_p^r) \right\} \tag{3.12}$$

$$\vdots \tag{3.13}$$

$$\beta_p^{r+1} = \underset{\beta_p}{argmin}\left\{ f(\beta_1^{r+1}, \beta_2^{r+1}, \ldots, \beta_p) \right\}, \tag{3.14}$$

and the cycle repeats iteratively $\forall\ j = 1, \ldots, p$, until the convergence.

The coordinate descent algorithm can converge for some special cases where the function is non-smooth or non-differentiable. Tseng [33, 34] proved that the coordinate descent algorithm can be generalized as "Block Coordinate Relaxation" (BCR) algorithm, which can be used for solving non-smooth regularization term.

Consider the following formulation:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{argmin}\left\{ f(\boldsymbol{\beta}) \right\} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{argmin}\left\{ S(\boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta}) \right\} \tag{3.15}$$

if $S(\boldsymbol{\beta})$ is convex and differentiable, and if $P(\boldsymbol{\beta})$ is convex, continuous in its effective domain, and separable such that $P(\boldsymbol{\beta}) = \sum_{j=1}^{p} \phi_j(\beta_j)$; then the coordinate descent algorithm converges to the global minimum [33]. Thus, this solution approach is applicable to the lasso and the elastic net.

The coordinate descent algorithm can be implemented in three different forms of update mechanisms. These update mechanisms will converge to the same solution, and have similar number of iterations. However, these updates mechanisms differ in the computing efficiencies. The updates mechanisms are: Naive updates, covariance updates, and coordinate newton updates. They are described as follows:

### 3.3.1 Solving OLS

**Naive Updates for OLS**

Consider the following OLS problem:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{argmin}\Big\{ S(\boldsymbol{\beta}) \Big\} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{argmin}\Big\{ \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \Big\}, \tag{3.16}$$

At the optimal solution, the partial derivative of $S(\boldsymbol{\beta})$ w.r.t. all $\beta_j$'s should be zero. To ultimately achieve this, the coordinate descent algorithm focuses on the partial derivative w.r.t. one variable in a give iteration:

$$\frac{\partial}{\partial \beta_j}\left( \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right) = 0 \tag{3.17}$$

$$-\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l=1}^{p} \beta_l x_{il}\right) = 0 \tag{3.18}$$

$$\sum_{i=1}^{N} x_{ij}\left(y_i - \beta_j x_{ij} - \sum_{l \neq j} \beta_l x_{il}\right) = 0 \tag{3.19}$$

$$\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l \neq j} \beta_l x_{il}\right) - \beta_j \sum_{i=1}^{N} x_{ij}^2 = 0 \tag{3.20}$$

$$\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l \neq j} \beta_l x_{il}\right) = \beta_j \sum_{i=1}^{N} x_{ij}^2 \tag{3.21}$$

$$\beta_j = \frac{\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l \neq j} \beta_l x_{il}\right)}{\sum_{i=1}^{N} x_{ij}^2} \tag{3.22}$$

For standardized parameters, $\sum_{i=1}^{N} x_{ij}^2 = N \; \forall \; j$. Now, let $z_j^{r+1}$ be described as:

$$z_j^{r+1} = \frac{1}{N} \sum_{i=1}^{N} x_{ij} \left(y_i - \sum_{l \neq j} \beta_l^* x_{il}\right), \tag{3.23}$$

where $\beta_l^*$ is the current or recent update value, i.e., for $1 \leq l < j$, $* = r+1$ and for $j \leq l \leq p$, $* = r$. Then, iteratively, executing $z_j^{r+1}$ for $j = 1, 2, \ldots, p, 1, 2, \ldots, p, \ldots$ until the convergence will lead to solve for the OLS estimate $\hat{\beta}_{oj} \; \forall \; j$.

### Covariance Updates for OLS

Friedman et al. [32] called the above way of solving coordinate descent iterations in Section 3.3.1 as "naive updates". They suggested another way of updates and called it as "covariance updates", which can be computationally efficient when $N \gg p$. Let $y_i^{(j)} = \sum_{l \neq j} \beta_l x_{il}$. the idea is described as follows:

$$y_i - y_i^{(j)} = y_i - \Bigg( \sum_{l=1}^{p} \beta_l x_{il} - \beta_j x_{ij} \Bigg). \tag{3.24}$$

From Equation (3.22), the following formulation is obtained:

$$\beta_j = \frac{1}{N} \Bigg( \sum_{i=1}^{N} \Bigg( x_{ij} y_i - \sum_{j=1}^{p} \beta_l \big( x_{ij} x_{il} \big) \Bigg) + N\beta_j \Bigg), \tag{3.25}$$

$$\beta_j = \frac{1}{N} \sum_{i=1}^{N} \Bigg( x_{ij} y_i - \sum_{j=1}^{p} \beta_l \big( x_{ij} x_{il} \big) \Bigg) + \beta_j. \tag{3.26}$$

Now, the update mechanism can be written as:

$$z_j^{r+1} = \frac{1}{N} \sum_{i=1}^{N} \Bigg( x_{ij} y_i - \sum_{j=1}^{p} \beta_l^{*} \left( x_{ij} x_{il} \right) \Bigg) + z_j^{r} \tag{3.27}$$

The covariance update is more efficient than the naive update when $N \gg p$, because $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}^T\mathbf{y}$ can be calculated and stored before the execution of the algorithm. On the contrary, the naive update approach calculates $y_i - y_i^{(j)}$ at every update.

### Coordinate Newton Updates for OLS

Coordinate newton update determines the step size from the partial second order derivative of $f(\boldsymbol{\beta})$. Fu [29] used the coordinate newton update for solving the lasso. In general, the update mechanism can be expressed as follows:

$$\beta_j^{r+1} = \beta_j^r - \left( \frac{\partial^2 f(\boldsymbol{\beta})}{\partial \beta_j \partial \beta_j} \right)^{-1} \frac{\partial f(\boldsymbol{\beta})}{\partial \beta_j} \bigg|_{\beta_j = \beta_j^r} \tag{3.28}$$

Since $f(\boldsymbol{\beta})$ in OLS, ridge regression, lasso and elastic net is quadratic, one coordinate newton update for $\beta_j$, when $\beta_j^r \neq 0$, will satisfy:

$$\frac{\partial f(\boldsymbol{\beta})}{\partial \beta_j} \bigg|_{\beta_j = \beta_j^r} = 0 \tag{3.29}$$

Coordinate newton update have the same number of iterations (like naive or covariance update), but without re-calculating $x_{ij}(y_i - y_i^{(j)})$ in every cycle, and does not require any storing of $\sum_{i=1}^{N} x_{ij} y_i$ and $\sum_{k=1}^{N} \sum_{i=1}^{N} x_{ij} x_{ik}$. The update mechanism for OLS can be expressed as follows:

$$\frac{\partial f(\boldsymbol{\beta})}{\partial \beta_j} = - \sum_{i=1}^{N} x_{ij} \left( y_i - \sum_{l=1}^{p} \beta_l x_{il} \right) \tag{3.30}$$

$$\frac{\partial^2 f(\boldsymbol{\beta})}{\partial \beta_j \partial \beta_j} = \sum_{i=1}^{N} x_{ij}^2 = N \tag{3.31}$$

Now, the update mechanism can be written as:

$$z_j^{r+1} = z_j^r + \frac{1}{N}\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l=1}^{p} \beta_l^* x_{il}\right) \tag{3.32}$$

Iteratively, the algorithm will be executed for $j = 1, 2, \ldots, p, 1, 2, \ldots, p, 1, \ldots$ until the convergence. Clearly, the coordinate newton update in OLS, is just another way of describing the naive or covariance update mechanism. However, it is usually computationally less expensive than the naive approach.

$$z_j^{r+1} = z_j^r + \frac{1}{N}\sum_{i=1}^{N} x_{ij}\left(y_i - z_j^r x_{ij} - \sum_{l\neq j} \beta_l^* x_{il}\right) \qquad \forall\ j \tag{3.33}$$

$$z_j^{r+1} = z_j^r - \frac{N}{N} z_j^r + \frac{1}{N}\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l\neq j} \beta_l^* x_{il}\right) \qquad \forall\ j \tag{3.34}$$

$$z_j^{r+1} = \frac{1}{N}\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l\neq j} \beta_l^* x_{il}\right) \qquad \forall\ j \tag{3.35}$$

### 3.3.2 Solving Penalized Linear Regression

#### The Ridge Regression

Consider the following ridge regression's mathematical model:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min} \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \frac{\lambda}{2}\sum_{l=1}^{p} \beta_l^2, \tag{3.36}$$

Since the above problem is convex, it can be solved by taking the derivative with respect to $\beta_j$ and equating it to zero:

$$-\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l=1}^{p} \beta_l x_{il}\right) + \lambda\beta_j = 0, \tag{3.37}$$

$$-\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l\neq 1} \beta_l x_{il}\right) - \beta_j \sum_{i=1}^{N} x_{ij}^2 + \lambda\beta_j = 0, \tag{3.38}$$

$$-\frac{1}{N}\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l\neq 1} \beta_l x_{il}\right) - \beta_j + \frac{\lambda}{N}\beta_j = 0, \tag{3.39}$$

the update of the ridge regression coefficient can be written as:

$$\beta_{j,ridge}^{r+1} = \frac{z_j^{r+1}}{1 + \dfrac{\lambda}{N}}. \tag{3.40}$$

Then, executing $\beta_{j,ridge}^{r+1}$ for $j = 1, 2, \ldots, p, 1, 2, \ldots, p, \ldots$ until the convergence, will lead to solve for the ridge estimate $\hat{\beta}_{j,ridge} \quad \forall\, j$.

The term $z_j^{r+1}$ can be updated by any of the above three update mechanisms. In addition to that, $\beta_{j,ridge}^{r+1}$ can also be updated directly by the coordinate newton update mechanism, which is described as follows:

$$\frac{\partial f(\boldsymbol{\beta})}{\partial \beta_j} = -\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l=1}^{p} \beta_l x_{il}\right) + \lambda\beta_j \tag{3.41}$$

$$\frac{\partial^2 f(\boldsymbol{\beta})}{\partial \beta_j \partial \beta_j} = \sum_{i=1}^{N} x_{ij}^2 + \lambda = N + \lambda \tag{3.42}$$

$$\beta_{j,ridge}^{r+1} = \beta_{j,ridge}^{r} + \frac{\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l=1}^{p} \beta_l^* x_{il}\right) - \lambda \beta_{j,ridge}^{r}}{N + \lambda} \tag{3.43}$$

This is just another way of re-writing coordinate newton update for the ridge regression. Let it be called as “direct coordinate newton updates.

### The Lasso

Consider the following lasso's mathematical model:

$$\underset{\boldsymbol{\beta} \in \mathbb{R}^p}{min} \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \sum_{l=1}^{p} |\beta_l|, \tag{3.44}$$

Although the above mathematical model is convex, but it is not differentiable when any $\beta_j$ is equal to zero. However, when $\beta_j \neq 0$ the gradient condition for optimality can be written as:

$$-\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l=1}^{p} \beta_l x_{il}\right) + \lambda\, sign\big(\beta_j\big) = 0, \tag{3.45}$$

If $\beta_{j,lasso}^{r+1} > 0$, then $\beta_{j,lasso}^{(+)r+1}$ is the positive update, defined as:

$$\beta_{j,lasso}^{(+)r+1} = \frac{1}{N} \sum_{i=1}^{N} x_{ij}\big(y_i - \sum_{l=1}^{p} \beta_l^* x_{il}\big) - \frac{\lambda}{N} \tag{3.46}$$

If $\beta_{j,lasso}^{r+1} < 0$, then $\beta_{j,lasso}^{(-)r+1}$ is the negative update, defined as:

$$\beta_{j,lasso}^{(-)r+1} = \frac{1}{N}\sum_{i=1}^{N} x_{ij}\big(y_i - \sum_{l=1}^{p} \beta_l^* x_{il}\big) + \frac{\lambda}{N} \tag{3.47}$$

Otherwise, $\beta_{j,lasso}^{r+1} = 0$. Since $\dfrac{\lambda}{N} > 0$, $\beta_{j,lasso}^{r+1}$ will always have a magnitude less than $z_j^{r+1}$ with the same sign of $z_j^{r+1}$ or will be zero. In general:

$$\beta_{j,lasso}^{r+1} = sign\big(z_j^{r+1}\big)\left(z_j^{r+1} - \frac{\lambda}{N}\right)_+ \tag{3.48}$$

where $(\beta)_+$ denotes $\max\{0, \beta\}$. Executing $\beta_{j,lasso}^{r+1}$ for $j = 1, 2, \ldots, p, 1, 2, \ldots, p, \ldots$ until the convergence, will lead to solve for the lasso estimate $\hat{\beta}_{j,lasso} \quad \forall\ j$.

The term $z_j^{r+1}$ can be updated by any of the above three update mechanisms. In addition to that, the lasso coefficient estimate can also be updated directly by the direct coordinate newton updates, as follows:

$$\frac{\partial f(\boldsymbol{\beta})}{\partial \beta_j} = -\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l=1}^{p} \beta_l x_{il}\right) + \lambda\, sign\big(\beta_j\big) \tag{3.49}$$

$$\frac{\partial^2 f(\boldsymbol{\beta})}{\partial \beta_j \partial \beta_j} = \sum_{i=1}^{N} x_{ij}^2 = N \tag{3.50}$$

Now, if $\beta_{j,lasso}^{r+1} > 0$, then $\beta_{j,lasso}^{(+)r+1}$ is the positive update, defined as:

$$\beta_{j,lasso}^{(+)r+1} = \beta_{j,lasso}^{r} - \frac{1}{N}\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l=1}^{p} \beta_l^* x_{il}\right) - \frac{\lambda}{N}. \tag{3.51}$$

And, if $\beta^{r+1}_{j,lasso} < 0$, then $\beta^{(-)r+1}_{j,lasso}$ is the negative update, defined as:

$$\hat{\beta}^{(-)r+1}_{j,lasso} = \beta^{r}_{j,lasso} - \frac{1}{N}\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l=1}^{p}\beta^{*}_{l}x_{il}\right) + \frac{\lambda}{N}. \tag{3.52}$$

The value of $z^{r+1}_{j}$ is not apparent in the above equations. However, its sign can be entirely estimated. It is very important to note that if $|z^{r+1}_{j}| > \frac{\lambda}{N}$, $\beta^{(+)r+1}_{j,lasso}$ and $\beta^{(-)r+1}_{j,lasso}$ will have the same sign as $z^{r+1}_{j}$. If $|z^{r+1}_{j}| < \frac{\lambda}{N}$, (i.e.$\beta^{r+1}_{j,lasso} = 0$), then $\beta^{(+)r+1}_{j,lasso}$ and $\beta^{(-)r+1}_{j,lasso}$ will have different signs, such that $\beta^{(+)r+1}_{j,lasso} < 0$, and $\beta^{(-)r+1}_{j,lasso} > 0$. Thus, one has to calculate both $\beta^{(+)r+1}_{j,lasso}$ and $\beta^{(-)r+1}_{j,lasso}$ to obtain the value of $\beta^{r+1}_{j,lasso}$, described as follows:

If $\beta^{(+)r+1}_{j,lasso} > 0$, $\beta^{(-)r+1}_{j,lasso} > 0$, then set $\beta^{r+1}_{j,lasso} = \beta^{(+)r+1}_{j,lasso}$.

Else if $\beta^{(+)r+1}_{j,lasso} < 0$, $\beta^{(-)r+1}_{j,lasso} < 0$, then set $\beta^{r+1}_{j,lasso} = \beta^{(-)r+1}_{j,lasso}$.

Otherwise, set $\beta^{r+1}_{j,lasso} = 0$.

Friedman et al. [35] and Wu and Lange [36] highlighted the significance of using coordinate descent method for solving the lasso. Coordinate descent algorithm is not often used in typical NLP approaches, but it is very efficient for the unconstrained convex problems (like lasso). Friedman et al. [32] argued that the efficiency of the algorithm for $L_1$-norm due to the fact that when the coefficient estimates are set to zero, then they will usually remain at zero. Thus, the updates are required only for the remaining nonzero coefficient estimates, until a zero level estimate increases to a non-zero value (very rare). It turns out that the threshold $(\frac{\lambda}{N})$ that emerged due to the $L_1$-norm gives the above stated efficiency to the coordinate descent algorithm.

### The Elastic Net

Consider the following mathematical formulation for elastic net:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min}\ \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+\lambda\Bigg(\alpha\sum_{l=1}^{p}|\beta_l|+\big(\frac{1-\alpha}{2}\big)\sum_{l=1}^{p}\beta_l^2\Bigg), \tag{3.53}$$

The model is convex, and following the same approach (as we did for lasso), we get:

$$-\sum_{i=1}^{N}x_{ij}\Bigg(y_i-\sum_{l=1}^{p}\beta_l x_{il}\Bigg)+\lambda\Bigg(\alpha\ sign\big(\beta_j\big)+\big(1-\alpha\big)\beta_l\Bigg)=0, \tag{3.54}$$

$$-\sum_{i=1}^{N}x_{ij}\Bigg(y_i-\sum_{l\neq 1}\beta_l x_{il}\Bigg)+\beta_j\sum_{i=1}^{N}x_{ij}^2+\lambda\,\alpha\ sign\big(\beta_j\big)+\lambda\big(1-\alpha\big)\beta_l=0, \tag{3.55}$$

$$-\frac{1}{N}\sum_{i=1}^{N}x_{ij}\Bigg(y_i-\sum_{l\neq 1}\beta_l x_{il}\Bigg)+\beta_j+\frac{\lambda\alpha}{N}\ sign\big(\beta_j\big)+\frac{\lambda\big(1-\alpha\big)}{N}\beta_l=0, \tag{3.56}$$

Now, if $\beta_{j,nnet}^{r+1} > 0$, then $\beta_{j,nnet}^{(+)r+1}$ is the positive update, defined as:

$$\beta_{j,nnet}^{(+)r+1} = \frac{z_j^{r+1} - \dfrac{\lambda\alpha}{N}}{1 + (1-\alpha)\dfrac{\lambda}{N}} \tag{3.57}$$

And, if $\beta_{j,nnet}^{r+1} < 0$, then $\beta_{j,nnet}^{(-)r+1}$ is the negative update, defined as:

$$\beta_{j,nnet}^{(-)r+1} = \frac{z_j^{r+1} + \dfrac{\lambda\alpha}{N}}{1 + (1-\alpha)\dfrac{\lambda}{N}} \tag{3.58}$$

Otherwise $\beta_{j,nnet}^{r+1} = 0$. Since $\dfrac{\alpha\lambda}{N} > 0$, $\beta_{j,nnet}^{r+1}$ will always have a magnitude less than $z_j^{r+1}$ with the same sign of $z_j^{r+1}$ or will be zero. In general:

$$\beta_{j,nnet}^{r+1} = \frac{sign\big(z_j^{r+1}\big)\left(z_j^{r+1} - \dfrac{\lambda}{N}\right)_+}{1 + (1-\alpha)\dfrac{\lambda}{N}} \tag{3.59}$$

Executing $\beta_{j,nnet}^{r+1}$ for $j = 1, 2, \ldots, p, 1, 2, \ldots, p, \ldots$ until the convergence, will lead to solve for the naive elastic net estimate $\hat{\beta}_{j,nnet} \quad \forall\ j$. The elastic net estimate $\hat{\beta}_{j,net}$ is a rescaling of the naive elastic net to avoid "double shrinkage", which can be obtained as:

$$\hat{\beta}_{j,net} = \left(1 + (1-\alpha)\frac{\lambda}{N}\right)\hat{\beta}_{j,nnet} \qquad \forall\ j. \tag{3.60}$$

The term $z_j^{r+1}$ can be updated by any of the three updates. The naive elastic net coefficient estimate also can be updated directly by direct coordinate newton updates as follows:

$$\frac{\partial f(\boldsymbol{\beta})}{\partial \beta_j} = -\sum_{i=1}^{N} x_{ij}\Bigg(y_i - \sum_{l=1}^{p} \beta_l x_{il}\Bigg) + \lambda\,\alpha\; sign\big(\beta_j\big) + \lambda(1-\alpha)\beta_j \tag{3.61}$$

$$\frac{\partial^2 f(\boldsymbol{\beta})}{\partial \beta_j \partial \beta_j} = \sum_{i=1}^{N} x_{ij}^2 + \lambda(1-\alpha) = N + \lambda(1-\alpha) \tag{3.62}$$

If $\beta_{j,nnet}^{r+1} > 0$, then $\beta_{j,nnet}^{(+)r+1}$ is the positive update, defined as:

$$\beta_{j,nnet}^{(+)r+1} = \beta_{j,nnet}^{r} + \frac{\sum_{i=1}^{N} x_{ij}\big(y_i - \sum_{l=1}^{p} \beta_l^* x_{il}\big) - \lambda\big(\alpha + (1-\alpha)\beta_{j,nnet}^{r}\big)}{N + \lambda(1-\alpha)}. \tag{3.63}$$

If $\beta_{j,nnet}^{r+1} < 0$, then $\beta_{j,nnet}^{(-)r+1}$ is the negative update, defined as:

$$\beta_{j,nnet}^{(-)r+1} = \beta_{j,nnet}^{r} + \frac{\sum_{i=1}^{N} x_{ij}\big(y_i - \sum_{l=1}^{p} \beta_l^* x_{il}\big) - \lambda\big(\alpha - (1-\alpha)\beta_{j,nnet}^{r}\big)}{N + \lambda(1-\alpha)}. \tag{3.64}$$

Similar to the lasso, one has to calculate both $\beta_{j,nnet}^{(+)r+1}$ and $\beta_{j,nnet}^{(-)r+1}$ as follows:

If $\beta_{j,nnet}^{(+)r+1} > 0$, & $\beta_{j,nnet}^{(-)r+1} > 0$, then set $\beta_{j,nnet}^{r+1} = \beta_{j,nnet}^{(+)r+1}$.

If $\beta_{j,nnet}^{(+)r+1} < 0$, & $\beta_{j,nnet}^{(-)r+1} < 0$, then set $\beta_{j,nnet}^{r+1} = \beta_{j,nnet}^{(-)r+1}$.

Otherwise, set $\beta_{j,nnet}^{r+1} = 0$.

It is worth to highlight the identicality between the orthogonal case and the coordinate descent algorithm. The orthogonal case is a special case of the coordinate descent algorithm, where it converges at the very first iteration $(r+1=1)$. This indicates that as the correlation between the predictors increases, the number of iterations of the coordinate descent algorithm also increase.

The path-wise coordinate algorithm is highly recommended when solving for the whole range of parameter $\lambda$, i.e. from OLS until all are set to zero. For example, for any estimate $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\beta}}(\lambda_1)$ will be a "warm start" for $\hat{\boldsymbol{\beta}}(\lambda_2)$, where $\lambda$ is monotonically increasing or decreasing. Let $Q_i$ be the number of iterations to get $\hat{\boldsymbol{\beta}}$ for $\lambda = \lambda_1$, $(\lambda_1 > \lambda_2 \ldots)$ starting from the same initial point. Let $M_{ij}$ be the number of iterations to get $\hat{\boldsymbol{\beta}}$ for $\lambda = \lambda_j$ starting from $\hat{\boldsymbol{\beta}}$ for $\lambda = \lambda_i$. Based on the path-wise approach, it can be seen that, $Q_i + M_{ij} < Q_i + Q_j$.

CHAPTER 4

# THE FIXED-SHAPE ELASTIC NET

## 4.1 Formulation of the Fixed-Shape Elastic Net

Consider solving for the naive elastic net (see Formulation (2.11)) for one predictor. It can be seen from Figure (4.1) that when $\hat{\beta}_{nnet}$ is large, a small change of $\lambda$ will shrink $\hat{\beta}_{nnet}$ by a large margin, and when $\hat{\beta}_{nnet}$ is small, a large change of $\lambda$ will shrink $\hat{\beta}_{nnet}$ by a small margin. On the other hand, the elastic net after rescaling rearranges the relationship between $\hat{\beta}_{net}$ and $\lambda$ to be linear for one predictor. However, they both lack the ability to fully capture the norm shape and norm size.

The lasso and ridge regression, have one tuning dimension; the norm size that is controlled by $t$ (or $\lambda$). On the other hand, the elastic net has two tuning

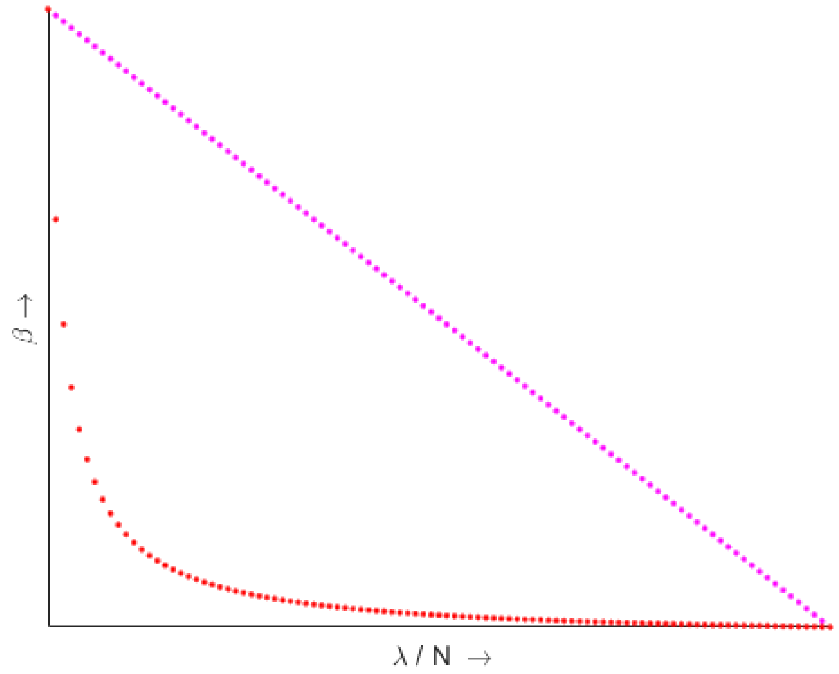

Figure 4.1: For one predictor, $y$-axis represents $\beta$ and $x$-axis represents $\frac{\lambda}{N}$. Rescaling $\beta$ after solving removes the effect of double shrinking.

dimensions; the first dimension is for the norm size, and the other is for the norm shape. These two dimensions provide a higher flexibility in the elastic net than compared to the lasso and to the ridge regression. However, the current method of solving the elastic net does not utilize the flexibility to its full capacity.

Typically, the role of $t$ (or $\lambda$) in the elastic net is to control the norm size. At first glance, one might think that the role of $\alpha$ in the elastic net is to control the norm shape. However, $\alpha$ does not fully control the norm shape. To have a clear picture, let us look at the weighted percentage of the elastic net penalty in Table (4.1) for different values of $t$:

$$\alpha t + \frac{1-\alpha}{2} t^2 \tag{4.1}$$

Table 4.1: When $\alpha = 0.5$, the composition of the lasso and ridge in Equation (4.1) for different values of $t$.

| $t$ | $\alpha t$ | $\frac{1-\alpha}{2}t^2$ | lasso % | Ridge % |
|---|---|---|---|---|
| 1000 | 500 | 250000 | 0.20% | 99.80% |
| 500 | 250 | 62500 | 0.40% | 99.60% |
| 100 | 50 | 2500 | 2.00% | 98.00% |
| 50 | 25 | 625 | 3.80% | 96.20% |
| 10 | 5 | 25 | 16.70% | 83.30% |
| 5 | 2.5 | 6.25 | 28.60% | 71.40% |
| 1 | 0.5 | 0.25 | 66.70% | 33.30% |
| 0.5 | 0.25 | 0.0625 | 80.00% | 20.00% |
| 0.1 | 0.05 | 0.0025 | 95.20% | 4.80% |
| 0.05 | 0.025 | 0.00063 | 97.60% | 2.40% |

From Table 4.1, it can be clearly seen, that at a constant value of $\alpha$, the norm shape is still changing. In addition to that, Figure 4.2 indicates the issue for a

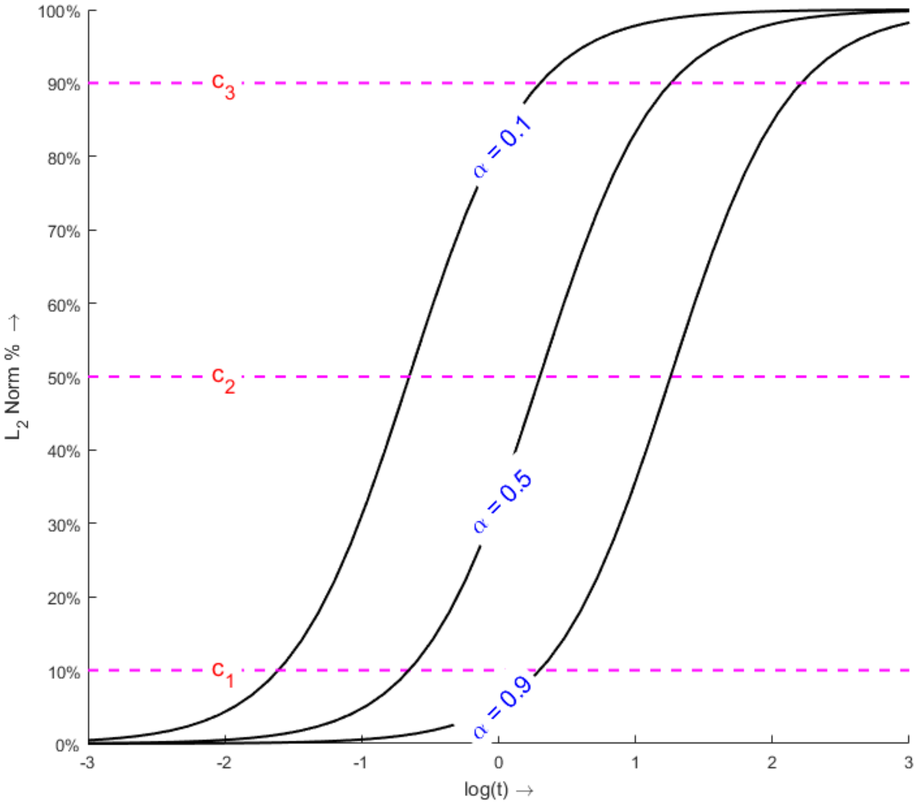

Figure 4.2: $y$-axis represents the composition of $L_2$-norm, and $x$-axis represents $log(t)$. Different fixed values of $\alpha$ will lead to different compositions at different values of $t$ for the conventional elastic net. The novel method will set a parameter $c$ that will have the same composition at different values of $t$.

fixed value of $\alpha$ at different values of $t$, where $0 \ll \alpha \ll 1$. It can be seen that, when $t$ is very high (very low), only $L_2$-norm ($L_1$-norm) is selected, irrespective of the value of $\alpha \in [0.1, 0.9]$. Furthermore, with a fixed value of $\alpha$, the trace plot contributes the solution to different norm shapes. Thus, with the current way of solving the elastic net, most of the combinations of norm shapes and norm sizes will not be fully explored.

In this chapter, a novel approach is proposed, called “fixed-shape elastic net”. The method overcomes the above issues of the elastic net, by having two tuning parameters such that one fully controls the norm shape and the other fully controls the norm size.

Table 4.2: The corresponding values $\alpha$ for different values of $t$ and $\alpha$ to give 50% for each norm.

| $\alpha$ | $t$ | $\alpha t$ | $\frac{1-\alpha}{2}t^2$ | lasso % | Ridge % |
|---|---|---|---|---|---|
| 0.998 | 1000 | 1000 | 1000 | 50% | 50% |
| 0.996 | 500 | 500 | 500 | 50% | 50% |
| 0.98 | 100 | 100 | 100 | 50% | 50% |
| 0.962 | 50 | 50 | 50 | 50% | 50% |
| 0.833 | 10 | 10 | 10 | 50% | 50% |
| 0.714 | 5 | 5 | 5 | 50% | 50% |
| 0.333 | 1 | 1 | 1 | 50% | 50% |
| 0.2 | 0.5 | 0.5 | 0.5 | 50% | 50% |
| 0.048 | 0.1 | 0.1 | 0.1 | 50% | 50% |
| 0.024 | 0.05 | 0.05 | 0.05 | 50% | 50% |

Table (4.2) shows the corresponding value of $\alpha$ and $t$ for a fixed norm shape at different norm sizes. In order to explore the full capacity of the elastic net, the traditional constraint of the elastic net needs to be reformulated. The traditional constraint is of the following form:

$$\sum_{j=1}^{p}\left(\alpha|\beta_j| + \frac{(1-\alpha)}{2}\beta_j^2\right) \leq \alpha t + \frac{1-\alpha}{2}t^2 \tag{4.2}$$

For $\alpha \in (0,1)$, divide Equation (4.2) on both sides by $\alpha$:

$$\sum_{j=1}^{p}\left(|\beta_j| + \frac{(1-\alpha)}{2\alpha}\beta_j^2\right) \leq t + \frac{1-\alpha}{2\alpha}t^2 \tag{4.3}$$

Let $\sigma > 0$ be defined as:

$$\sigma = \frac{\alpha}{1-\alpha} \tag{4.4}$$

The elastic net constraint can be written as:

$$\sum_{j=1}^{p} \left( |\beta_j| + \frac{\beta_j^2}{2\sigma} \right) \leq t + \frac{t^2}{2\sigma} \tag{4.5}$$

As $\alpha \to 1$, it can be seen that $\sigma \to \infty$. Thus the second term $\dfrac{\beta_j^2}{2\sigma} \to 0$, which implies that the norm shape tends towards lasso's shape. When $\alpha \to 0$, it can be seen that $\sigma \to 0$. Thus, the second term $\dfrac{\beta_j^2}{2\sigma} \to \infty$, which implies that the norm shape tends towards ridge's shape.

Let $\sigma = ct$, where $c > 0$ is the shape parameter. The traditional elastic net constraint can be updated as:

$$\sum_{j=1}^{p} \left( |\beta_j| + \frac{\beta_j^2}{2ct} \right) \leq t\left(1 + \frac{1}{2c}\right) \tag{4.6}$$

Therefore, the fixed-shape elastic net can be formulated as follows:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min} \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \qquad s.t. \sum_{j=1}^{p} \left( |\beta_j| + \frac{\beta_j^2}{2ct} \right) \leq t\left(1 + \frac{1}{2c}\right), \tag{4.7}$$

where $t \geq 0$ controls the norm size, such that the maximum value any $|\beta_j|$ can reach is $t$, and $c$ controls the norm shape. Equivalently, the penalized form of the fixed-shape elastic net can be formulated as follows:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min}\, \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p}\left(|\beta_j| + \frac{\beta_j^2}{2ct}\right), \tag{4.8}$$

where $\lambda \geq 0$ is the penalty parameter. If $\lambda = 0$, then it is similar to solving for OLS. If $\lambda \to \infty$, then all coefficient estimates will be set to 0.

### 4.1.1 Solving the Fixed-Shape Elastic Net Problem

A Coordinate descent algorithm based method is proposed to solve the fixed-shape elastic net. The algorithm is described in the following steps:

**Input**: The outer iteration counter $v$, the penalty parameter $\lambda^{(v)}$, and the shape parameter $c$.

**Step 1**: Set $r = 0$, the iteration counter. Initialize $\boldsymbol{\beta}^r_{fnet}$ as follows: if $v = 1$, then $\boldsymbol{\beta}^r_{fnet} = \hat{\boldsymbol{\beta}}_o$, otherwise $\boldsymbol{\beta}^r_{fnet}|_v = \hat{\boldsymbol{\beta}}_{fnet}|_{(v-1)}$, where $\boldsymbol{\beta}^r_{fnet}$ is the $r^{th}$ update for the fixed-shape elastic net estimate $\hat{\boldsymbol{\beta}}_{fnet}$

**Step 2**: Set $j = 1$.

**Step 3**: Update $t_j^{r+1}$ as per Equation (4.9), except when all $\beta_{r+1}^{j,fnet} = 0$.

$$t_j^{r+1} = \frac{\sum_{l=1}^{p}|\beta^*_{l,fnet}| + \sqrt{\left(\sum_{l=1}^{p}|\beta^*_{l,fnet}|\right)^2 + 2\left(1+\frac{1}{2c}\right)\left(\sum_{l=1}^{p}\frac{{\beta^*_{l,fnet}}^2}{c}\right)}}{2\left(1+\frac{1}{2c}\right)} \tag{4.9}$$

where $\beta^*_{l,fnet}$ is the current or recent updated value, i.e., for $1 \leq l < j$, $* = r+1$ and for $j \leq l \leq p$, $* = r$.

**Step 4**: Find $\beta_{j,fnet}^{r+1}$ as per Equation (4.10):

$$\beta_{j,fnet}^{r+1} = \frac{sign\left(z_j^{r+1}\right)\left(|z_j^{r+1}| - \frac{\lambda}{N}\right)_+}{1 + \frac{\lambda}{ct_j^{r+1}N}} \tag{4.10}$$

**Step 5**: Set $j = j+1$. If $j \leq p$, then repeat Step 3 and Step 4. If $j = p+1$, then set $r = r+1$. If termination criterion is met, then stop, otherwise go to Step 2.

The above algorithm is repeated by updating the value of $v$ as $v = v+1$, and $\lambda^{(v+1)} > \lambda^{(v)}$.

The KKT conditions of the new formulation can be written as follows:

**Optimality condition:**

$$\nabla S(\boldsymbol{\beta}) + \lambda \nabla g(\boldsymbol{\beta}) = \mathbf{0}, \qquad \forall\, \beta_j \neq 0 \tag{4.11}$$

$$\begin{bmatrix} -\sum_{i=1}^{N}\left(x_{i1}\left(y_i - \sum_{j=1}^{p}\beta_j x_{ij}\right)\right) \\ -\sum_{i=1}^{N}\left(x_{i2}\left(y_i - \sum_{j=1}^{p}\beta_j x_{ij}\right)\right) \\ \vdots \\ -\sum_{i=1}^{N}\left(x_{ip}\left(y_i - \sum_{j=1}^{p}\beta_j x_{ij}\right)\right) \end{bmatrix} + \lambda \begin{bmatrix} sign(\beta_1) + \dfrac{\beta_1}{ct} \\ sign(\beta_2) + \dfrac{\beta_2}{ct} \\ \vdots \\ sign(\beta_p) + \dfrac{\beta_p}{ct} \end{bmatrix} = \mathbf{0}, \qquad \forall\, \beta_j \neq 0 \tag{4.12}$$

**Dual feasibility:**

$$\lambda \geq 0 \tag{4.13}$$

**Primal feasibility:**

$$g(\boldsymbol{\beta}) \leq 0 \tag{4.14}$$

$$\sum_{j=1}^{p}\left(|\beta_j| + \frac{\beta_j^2}{2ct}\right) - t - \frac{t^2}{2ct} \leq 0 \tag{4.15}$$

**Complementary slackness:**

$$\lambda \; g(\boldsymbol{\beta}) = 0 \tag{4.16}$$

$$\lambda\left(\sum_{j=1}^{p}\left(|\beta_j| + \frac{\beta_j^2}{2ct}\right) - t - \frac{t^2}{2ct}\right) = 0 \tag{4.17}$$

Since the updating estimates of fixed-shape elastic net requires the information of $t$, following idea is used to get the corresponding value of $t$ at any iteration. When $\lambda > 0$, there exists a corresponding $t$, such that Equation (4.7) is active or binding. Rewriting $\alpha$ in term of $c$, the binding constraint can be written as:

$$\sum_{j=1}^{p}\left(|\beta_j| + \frac{\beta_j^2}{2ct}\right) \leq t\left(1 + \frac{1}{2c}\right) \tag{4.18}$$

$$t^2\left(1 + \frac{1}{2c}\right) - t\sum_{j=1}^{p}|\beta_j| - \sum_{j=1}^{p}\frac{\beta_j^2}{2c} = 0 \tag{4.19}$$

By solving the quadratic equation for $t$:

$$t = \frac{\sum_{j=1}^{p}|\beta_j| \pm \sqrt{\left(\sum_{j=1}^{p}|\beta_j|\right)^2 + 2\left(1 + \frac{1}{2c}\right)\left(\sum_{j=1}^{p}\frac{\beta_j^2}{c}\right)}}{2\left(1 + \frac{1}{2c}\right)} \tag{4.20}$$

Since $\sqrt{\left(\sum_{j=1}^{p}|\beta_j|\right)^2 + 2\left(1+\dfrac{1}{2c}\right)\left(\sum_{j=1}^{p}\dfrac{\beta_j^2}{c}\right)} \geq \sum_{j=1}^{p}|\beta_j|$ and $t \geq 0$,

$$t = \frac{\sum_{j=1}^{p}|\beta_j| + \sqrt{\left(\sum_{j=1}^{p}|\beta_j|\right)^2 + 2\left(1+\dfrac{1}{2c}\right)\left(\sum_{j=1}^{p}\dfrac{\beta_j^2}{c}\right)}}{2\left(1+\dfrac{1}{2c}\right)} \tag{4.21}$$

Furthermore, Equation (4.10) is obtained from the penalized fixed-shape elastic net formulation. The stationary point should satisfy the following criterion:

$$-\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l=1}^{p}\beta_l x_{il}\right) + \lambda\, sign\,(\beta_j) + \frac{\lambda\beta_j}{ct} = 0, \quad \beta_j \neq 0,\ \forall\, j. \tag{4.22}$$

If $\beta_{j,fnet}^{r+1} > 0$, then $\beta_{j,fnet}^{(+)r+1}$ is the positive update that can be defined as:

$$\beta_{j,fnet}^{(+)r+1} = \frac{z_j^{r+1} - \dfrac{\lambda}{N}}{1 + \dfrac{\lambda}{c\,t\,N}} \tag{4.23}$$

If $\beta_{j,fnet}^{r+1} < 0$, then $\beta_{j,fnet}^{(-)r+1}$ is the negative update that can be defined as:

$$\beta_{j,fnet}^{(-)r+1} = \frac{z_j^{r+1} + \dfrac{\lambda}{N}}{1 + \dfrac{\lambda}{c\,t\,N}} \tag{4.24}$$

Otherwise $\beta_{j,fnet}^{r+1} = 0$. In general:

$$\beta_{j,fnet}^{r+1} = \frac{sign\left(z_j^{r+1}\right)\left(|z_j^{r+1}| - \frac{\lambda}{N}\right)_+}{1 + \frac{\lambda}{c\,t\,N}} \tag{4.25}$$

Thus, using the proposed algorithm, the fixed-shape elastic net can be efficiently

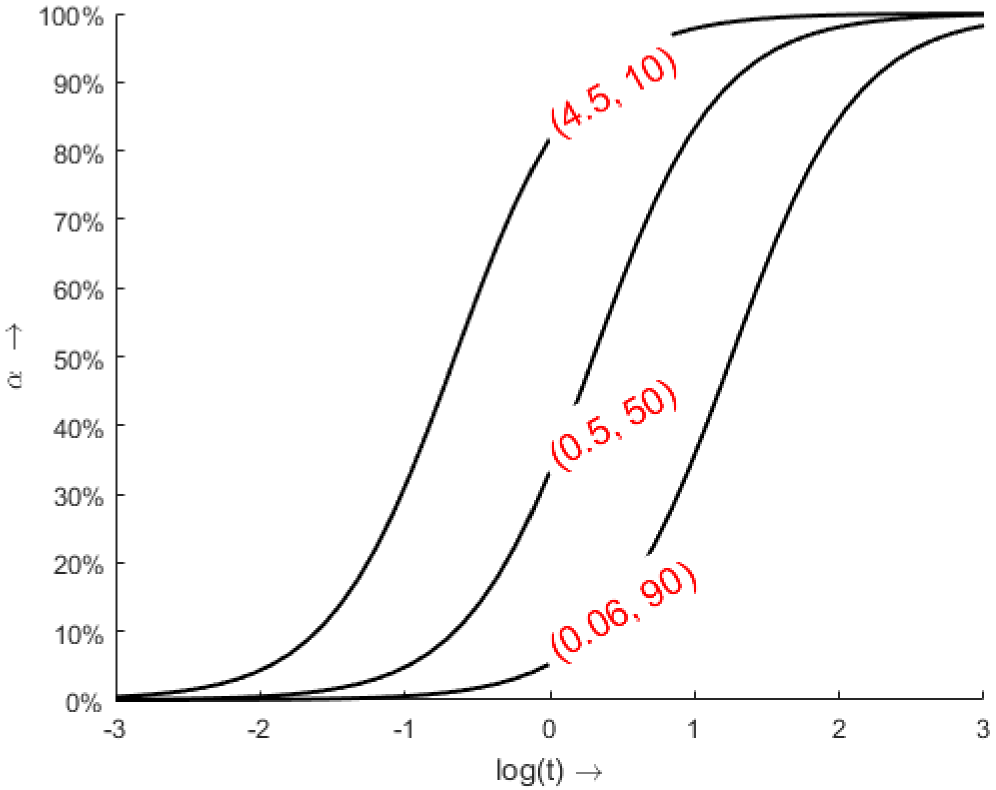

Figure 4.3: $y$-axis represents $\alpha$, and $x$-axis represents $log(t)$. The order pairs $(4.5, 10)$, $(0.5, 50)$, & $(0.06, 90)$ represent $c$ and $L_2$-norm (%), respectively.

solved for given values of $\lambda$ & $c$.

### 4.1.2 Parameter $c$ and Norm Shape

The fixed-shape elastic net requires solving the problem as the conventional elastic net and then correcting the norm shape in order to have consistent norm shape along the whole path. Thus, it utilizes every possible combination of $L_1$-norm & $L_2$-norm. The fixed-shape elastic net solution approach is similar to the projected gradient algorithm by having a corrective action after every iteration.

Notice that every parameter is responsible for controlling one dimension in the fixed-shape elastic net, i.e., $t$ controls only the norm size and $c$ controls only the norm shape. When $c \to \infty$, the problem tends towards the lasso, and when $c \to 0$, the problem tends towards the ridge regression (See Figure (4.3) and Figure (4.4)). The list of figures (Figure (4.5) - Figure (4.14)) indicates the trace plots for different values of $c$.

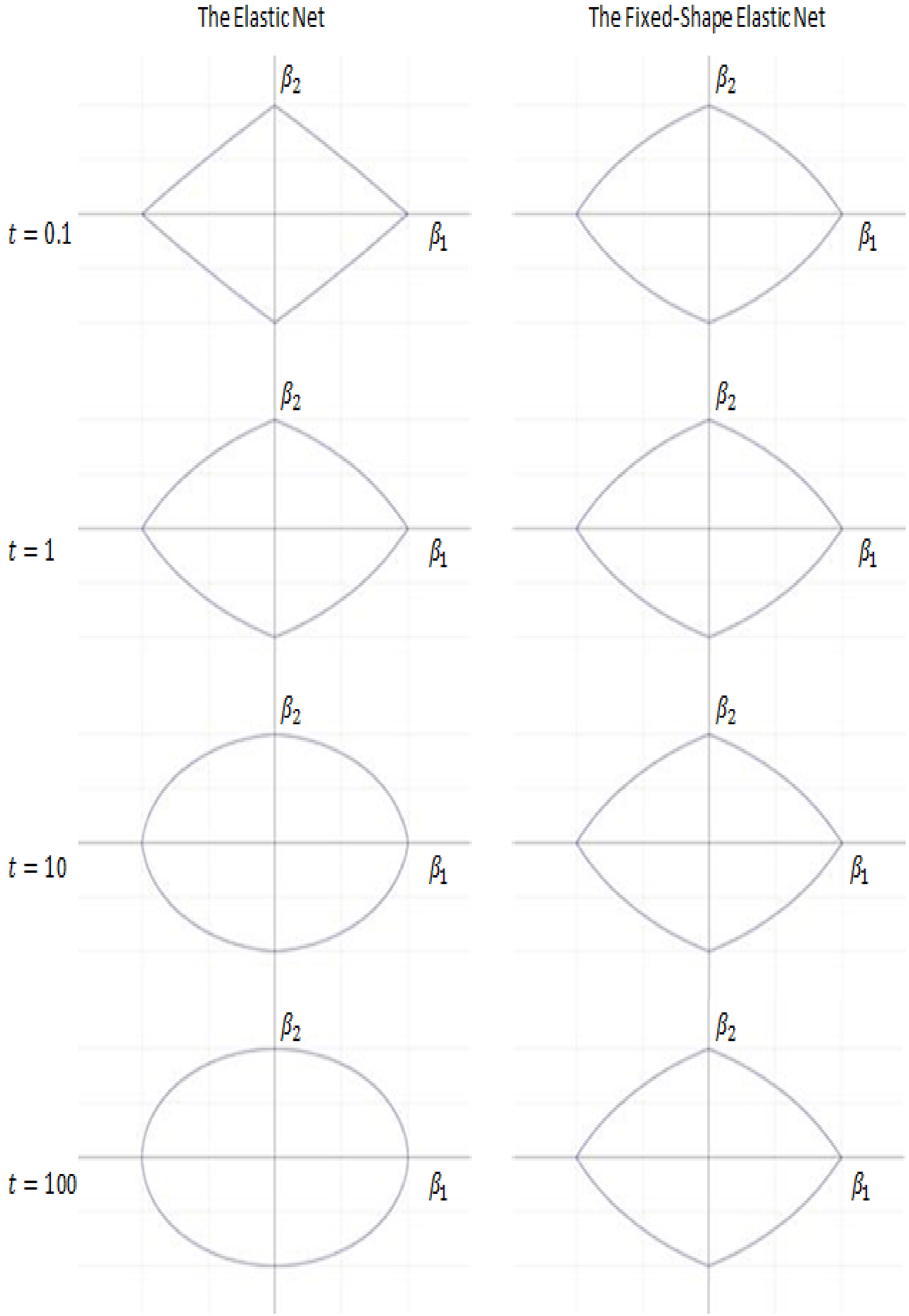

Figure 4.4: The first column is for the conventional elastic net with $\alpha = 0.5$, and the second column is for the fixed-shape elastic net with $c = 1$. The tuning parameter $t = 0.1, 1, 10, 100$ respectively as per rows from above.

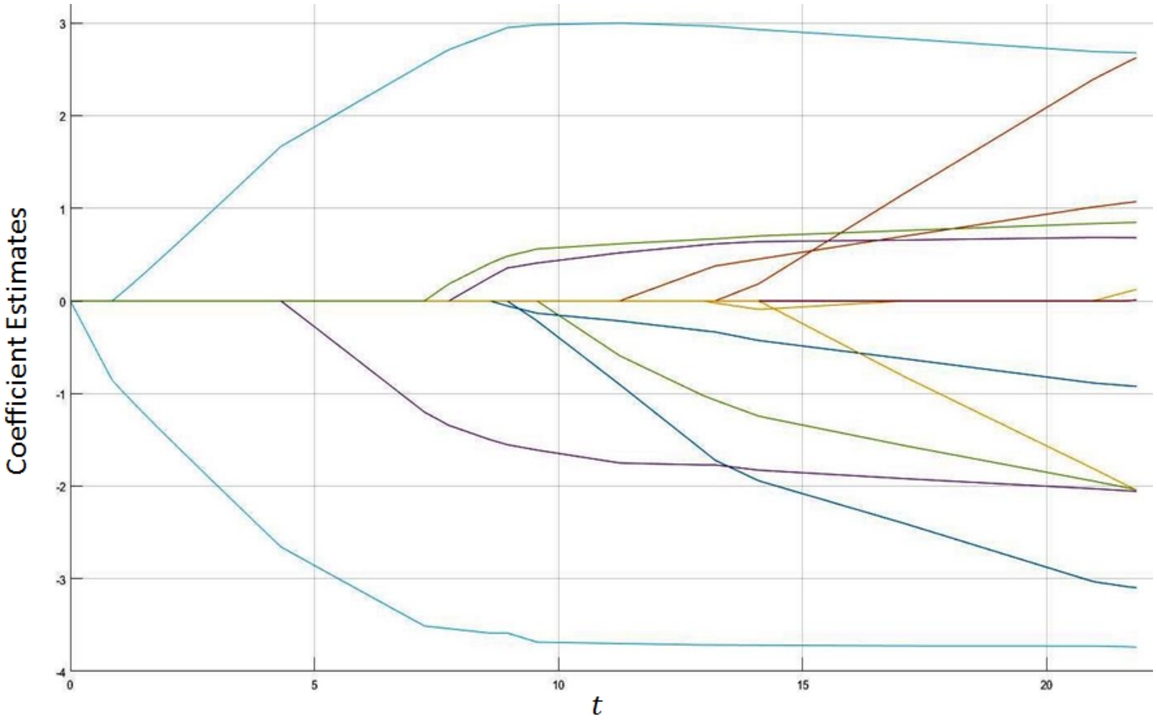

Figure 4.5: Trace plot of Boston dataset by the fixed-shape elastic net with $c = 1000$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

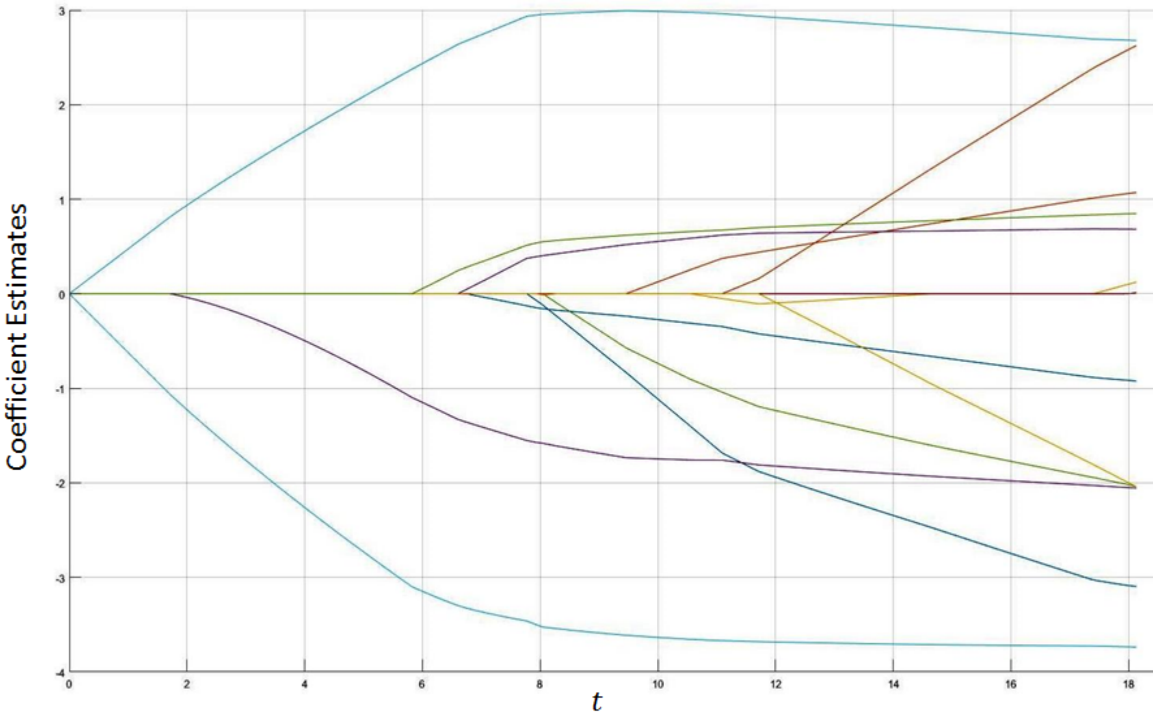

Figure 4.6: Trace plot of Boston dataset by the fixed-shape elastic net with $c = 2$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

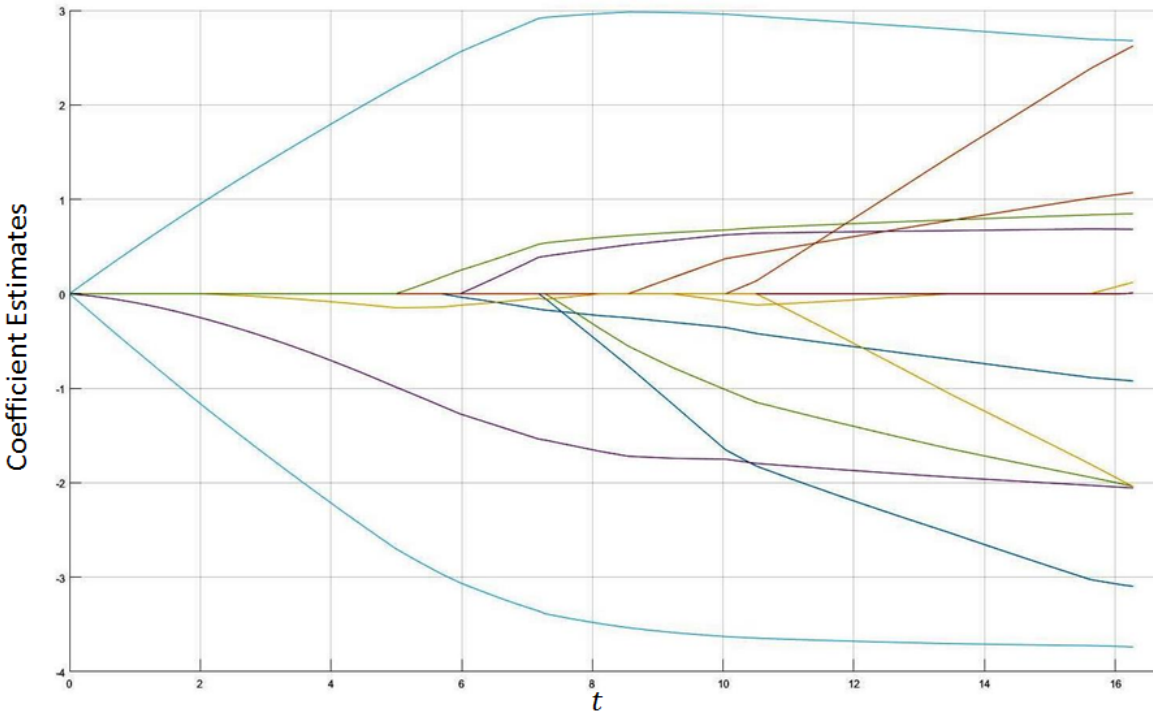

Figure 4.7: Trace plot of Boston dataset by the fixed-shape elastic net with $c = 1.15$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

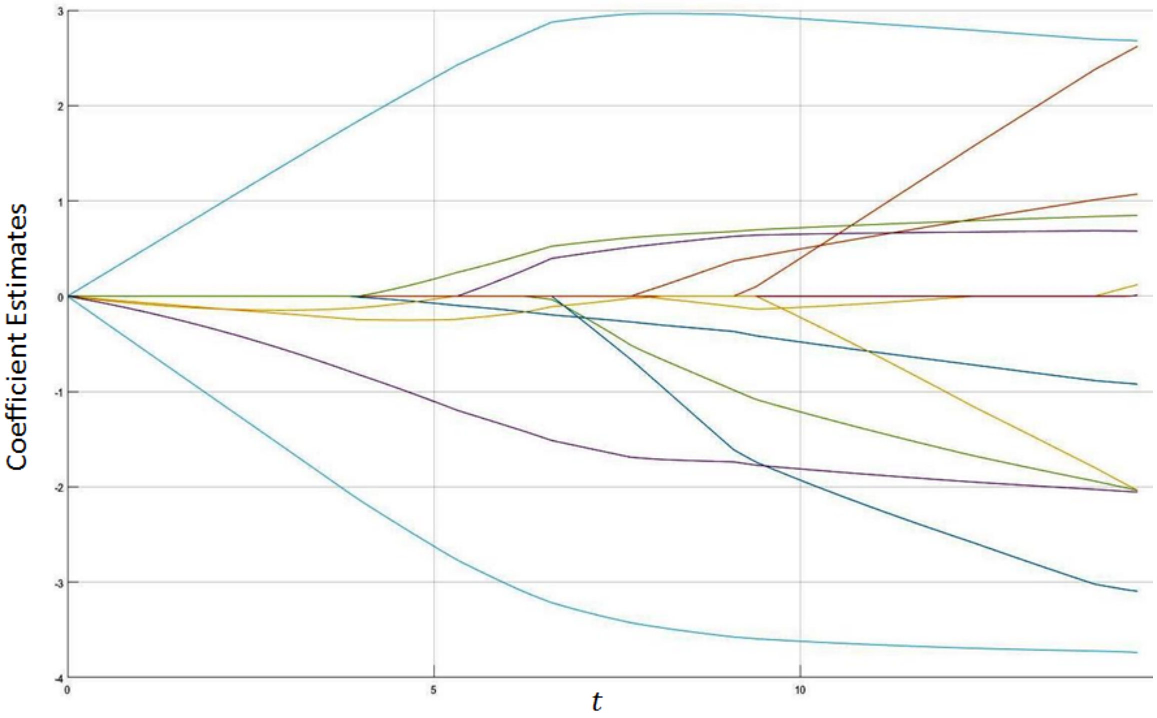

Figure 4.8: Trace plot of Boston dataset by the fixed-shape elastic net with $c = 0.75$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

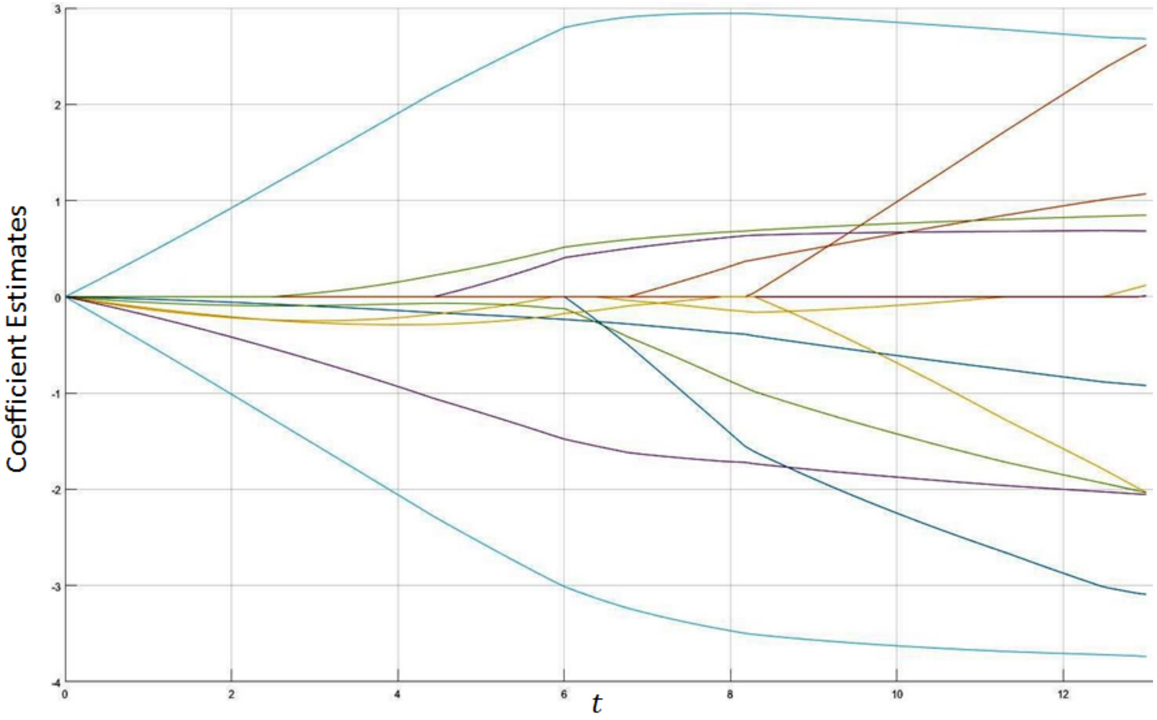

Figure 4.9: Trace plot of Boston dataset by the fixed-shape elastic net with $c = 0.5$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

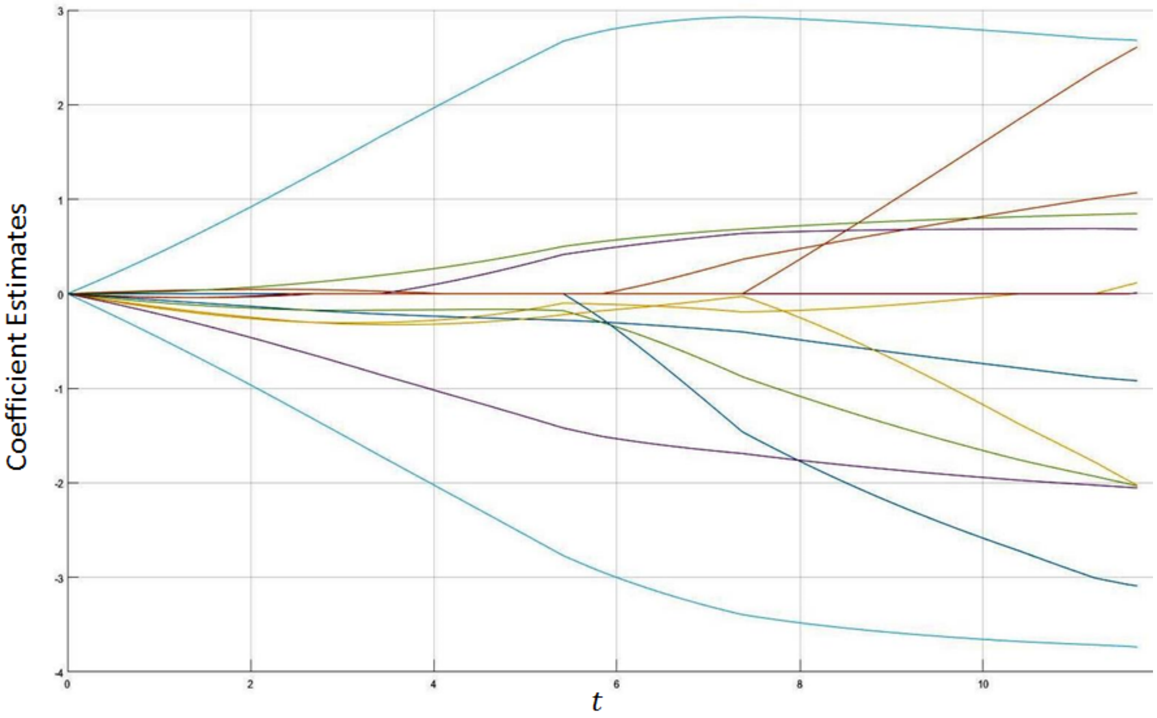

Figure 4.10: Trace plot of Boston dataset by the fixed-shape elastic net with $c = 0.35$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

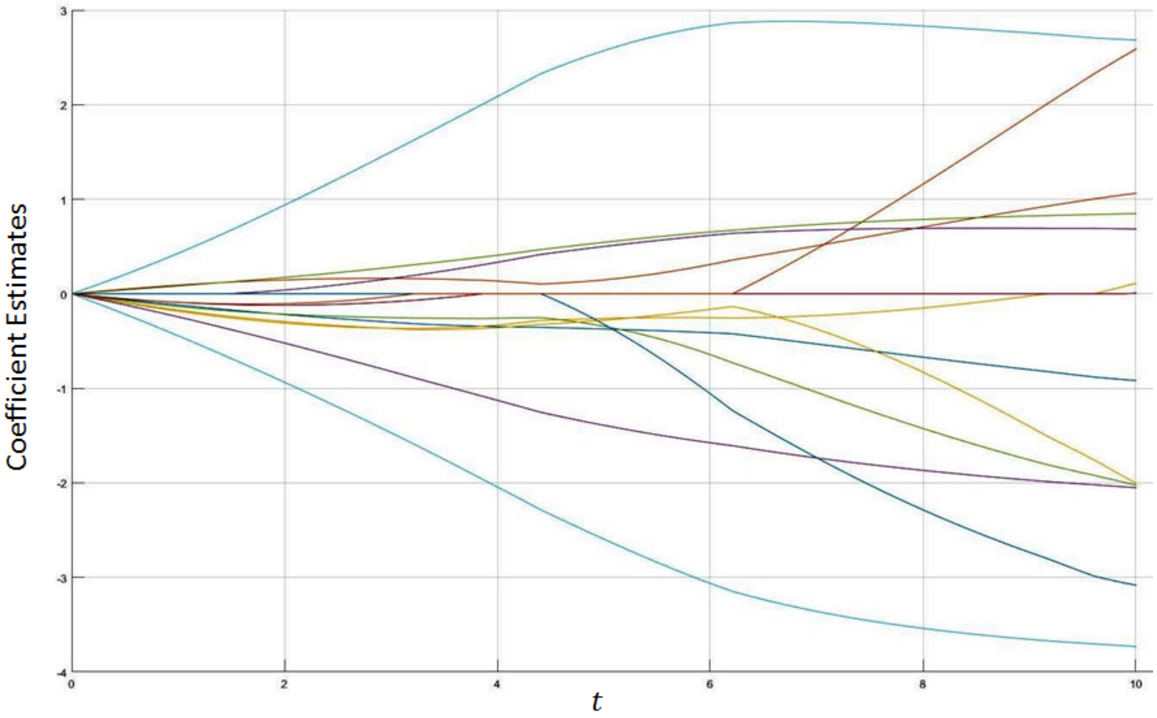

Figure 4.11: Trace plot of Boston dataset by the fixed-shape elastic net with $c = 0.2$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

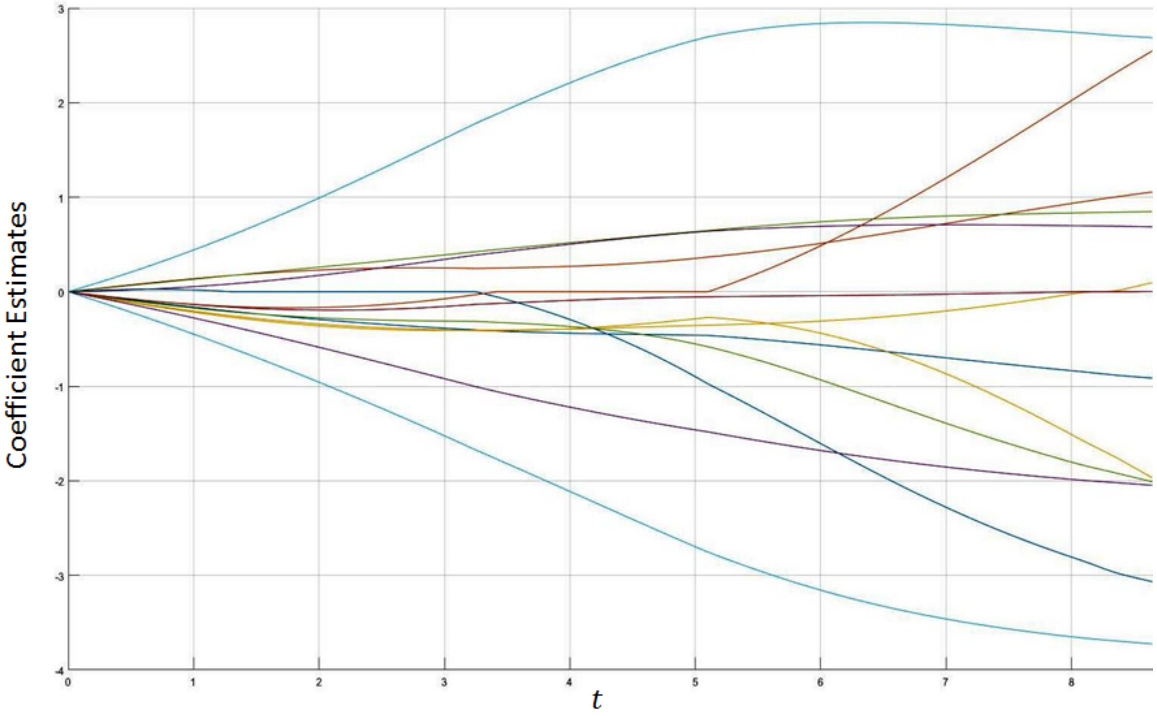

Figure 4.12: Trace plot of Boston dataset by the fixed-shape elastic net with $c = 0.1$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

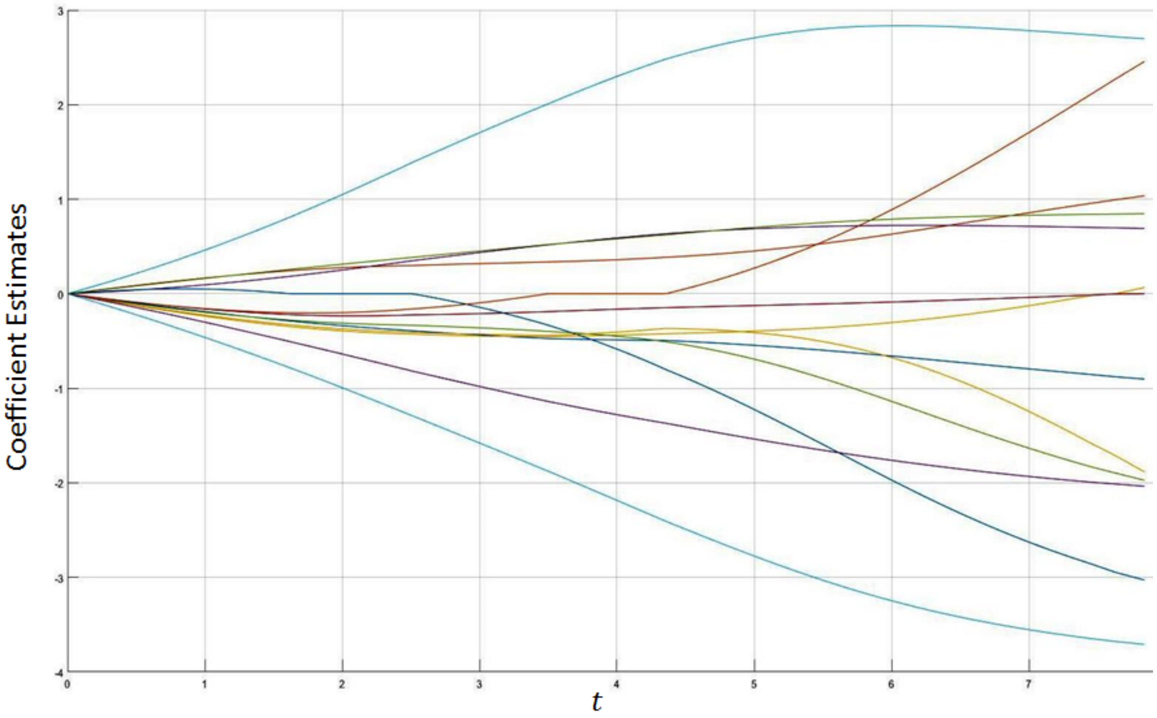

Figure 4.13: Trace plot of Boston dataset by the fixed-shape elastic net with $c = 0.05$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

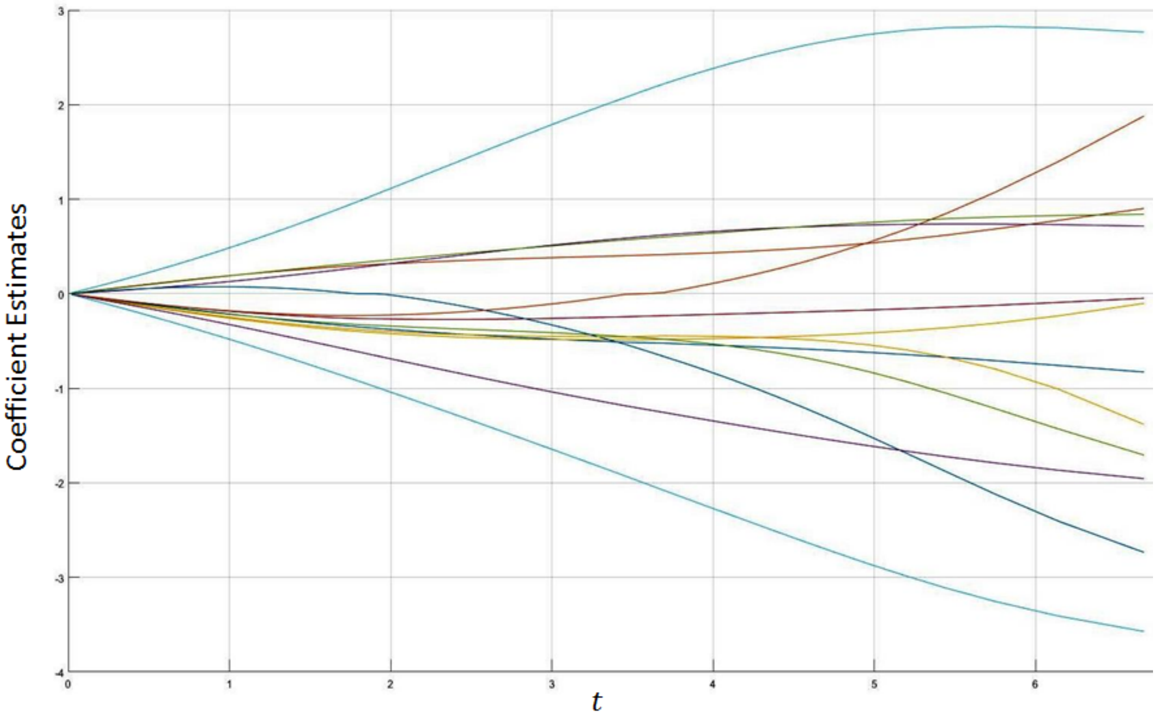

Figure 4.14: Trace plot of Boston dataset by the fixed-shape elastic net with $c = 0.01$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

## 4.2 The Relationship Between $\lambda$ and $t$

The relationship between $t$ and $\lambda$ is inversely proportional. When $\lambda$ monotonically increases, $t$ monotonically decreases. It is very critical to have the regularization term in this form $\sum_{j=1}^{p}\left(|\beta_j| + \frac{\beta_j^2}{2\sigma}\right)$. If the form was $\sum_{j=1}^{p}\left(\frac{|\beta_j|}{\sigma} + \frac{\beta_j^2}{2\sigma^2}\right)$ then the relationship between $t$ and $\lambda$ will not be inversely propotional, and the monotonic relationship between them will not exist. The corresponding coordinate update for $\beta_{j,fnet}^{r+1}$ can be expressed as:

$$\beta_{j,fnet}^{r+1} = \frac{sign\big(z_j^{r+1}\big)\left(|z_j^{r+1}| - \frac{\lambda}{\sigma N}\right)_+}{1 + \frac{\lambda}{\sigma^2 N}} \tag{4.26}$$

At the initial stage, if $\lambda$ increases, then $t$ decreases and so does $\sigma = ct$. The threshold term $\frac{\lambda}{\sigma N}$ in practice will always make $\hat{\beta}_{j,fnet} = 0$ at some finite $\lambda$. In other words, the trace plots will not be continuous. Looking closely at the case of one predictor for the form $\sum_{j=1}^{p}\left(\frac{|\beta_j|}{\sigma} + \frac{\beta_j^2}{2\sigma^2}\right)$, where $t = |\hat{\beta}_{j,fnet}|$ will provide an in-depth idea. Let $\hat{\beta}_o > \frac{\lambda}{\sigma N}$, then:

$$t = \frac{\hat{\beta}_o - \frac{\lambda}{\sigma N}}{1 + \frac{\lambda}{\sigma^2 N}} \tag{4.27}$$

Since $\sigma = ct$, we have:

$$t = \frac{\hat{\beta}_o - \frac{\lambda}{ctN}}{1 + \frac{\lambda}{c^2 t^2 N}} \tag{4.28}$$

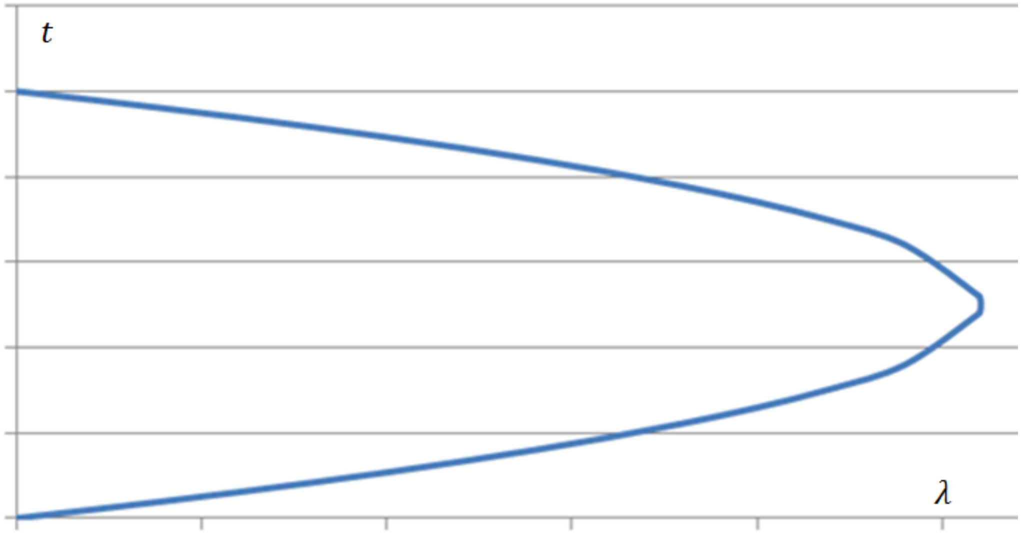

Figure 4.15: For one predictor, the form; $\dfrac{|\beta|}{\sigma}+\dfrac{\beta^2}{2\sigma^2}$ will let two values of $t$ ($y$-axis) correspond to one value of $\lambda$ ($x$-axis).

$$t=\frac{\hat{\beta}_o\pm\sqrt{\hat{\beta}_o^2-4\lambda(\dfrac{1}{c^2N}+\dfrac{1}{cN})}}{2} \tag{4.29}$$

Consider Equation (4.29), which defines the relationship between $t$ and $\lambda$. Figure (4.15) illustrates the relationship, and indicates the reason for the discontunuity in the trace plot. Since there is no direct relationship between $\lambda$ and $t$, it is very critical to make sure that for every dual variable $\lambda$, there is one unique optimal solution for $\hat{\boldsymbol{\beta}}$. Looking closely at the case of one predictor for the form $\sum_{j=1}^{p}\left(|\beta_j|+\dfrac{\beta_j^2}{2\sigma}\right)$ where $t=|\hat{\beta}_{j,fnet}|$ will provide an in-depth idea. Let $\hat{\beta}_o>\dfrac{\lambda}{N}$, then:

$$t=\frac{\hat{\beta}_o-\dfrac{\lambda}{N}}{1+\dfrac{\lambda}{\sigma N}} \tag{4.30}$$

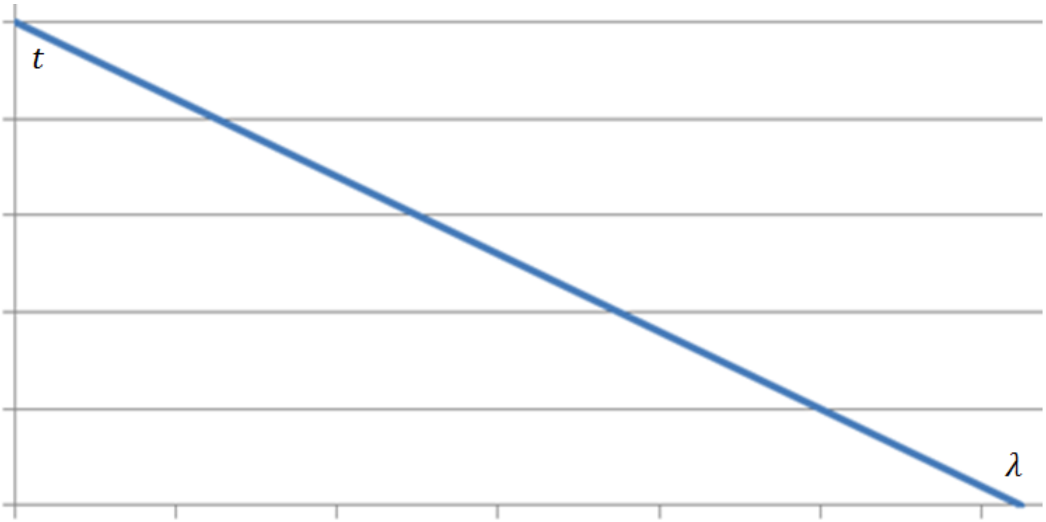

Figure 4.16: For one predictor, the form; $|\beta| + \dfrac{\beta^2}{2\sigma}$ will let two values of $t$ ($y$-axis) correspond to one value of $\lambda$ ($x$-axis).

Since $\sigma = ct$, we have:

$$t = \frac{\hat{\beta}_o - \dfrac{\lambda}{N}}{1 + \dfrac{\lambda}{ctN}} \tag{4.31}$$

$$t = \hat{\beta}_o - \lambda\left(\frac{1}{N} + \frac{1}{cN}\right) \tag{4.32}$$

Figure (4.16) illustrates the new relationship, and indicates the continuity in the trace plot for the fixed-shape elastic net. In addition to that, the threshold $\dfrac{\lambda}{N}$ is not affected by $\sigma$.

## CHAPTER 5

# THE EXPONENTIAL NORM REGRESSION

## 5.1 The Exponential Norms

As seen in the literature review, one family of the PLR is the bridge regression with $L_q$-norms, where the penalty is $\lambda \sum_{j=1}^{p} |\beta_j|^q$. When $q > 1$ the problem is strictly convex but not sparse. When $q < 1$ the problem is sparse but not convex. The only norm that is convex and sparse is when $q = 1$, i.e. the lasso. (See Table 5.1 ).

Table 5.1: The bridge regression status of convexity and sparsity with different $q$

| **$L_q$-norm** | $q > 1$ | $q < 1$ | $q = 1$ |
|---|---|---|---|
| **Convexity** | Strictly Convex | Non-convex | Convex |
| **Sparsity** | Non-sparse | Sparse | Sparse |

Another type of PLR is the combination between two different norms. The elastic net and the fixed-shape elastic net utilize a combination of two norms, $q = 1$ and $q = 2$ (i.e. the combination between the lasso and the ridge regression). The purpose of the first norm is to achieve sparsity, and the purpose of the second norm is to obtain the characteristics of grouping effect and to deal with high correlation among the predictors.

What will happen if a higher degree norm (say $q \geq 3$) is added to the combination $q = 1$ and $q = 2$ norms? Will the new norm add flexibility than compared to the elastic net (as the elastic net is flexible compared to the lasso)? In order to answer these questions, a novel family is introduced to the penalized linear regression, and the family is named as $L_q$-exponential norms. The $L_q$-exponential norm is defined as:

$$\sum_{j=1}^{p} \left( e^{\frac{|\beta_j|^q}{q!\sigma^q}} - 1 \right) \leq e^{\frac{t^q}{q!\sigma^q}} - 1 \tag{5.1}$$

where $t$ is the tuning parameter that controls the size of the norm, and $\sigma = ct$, where $c$ controls the size of the norm.

To find the predictor estimates $\hat{\boldsymbol{\beta}}_{exp}$, the problem can be formulated as:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \qquad s.t. \quad \sum_{j=1}^{p} \left( e^{\frac{|\beta_j|^q}{q!\sigma^q}} - 1 \right) \leq e^{\frac{t^q}{q!\sigma^q}} - 1, \tag{5.2}$$

where $t \geq 0$ is a tuning parameter that controls the size of the norm, such that when $t = 0$, then $\hat{\beta}_{j,exp}$ will be zero $\forall\ j$, where $\hat{\beta}_{j,exp}$ denotes the $j^{th}$ element of $\hat{\boldsymbol{\beta}}_{exp}$. Let $t_o$ be the minimum value such that when $t \geq t_0$, then the constraint will be redundant and $\hat{\beta}_{j,exp}$ will be the same as $\hat{\beta}_{oj}\ \forall\ j$. Parameter $\sigma = ct > 0$ has a similar role to the $\sigma$ in the fixed-shape elastic net, where $c$ controls the shape of the norm. The exponential norm is a combination of infinite norms that have the power of $mq$ where $m = 1, 2, 3, \ldots, \infty$. The Taylor series expansion of the $L_q$-exponential norm can reveal further understanding of its nature:

$$\sum_{j=1}^{p}\left(e^{\dfrac{|\beta_j|^q}{q!\sigma^q}} - 1\right) = \sum_{j=1}^{p}\sum_{m=1}^{\infty}\frac{|\beta_j|^{mq}}{(m!)(q!)^m\sigma^{mq}} = \sum_{j=1}^{p}\left(\frac{|\beta_j|^q}{(q!)\sigma^q} + \frac{|\beta_j|^{2q}}{2(q!)^2\sigma^{2q}} + \frac{|\beta_j|^{3q}}{3!(q!)^3\sigma^{3q}} + \ldots\right). \tag{5.3}$$

Since $\sigma = ct$, then as $c \to \infty$, it can be seen that except the first term of the Taylor expansion $\dfrac{|\beta_j|^q}{(q!)\sigma^q}$, the other remaining terms go to zero. Thus, they can be neglected. Therefore, the formulation will be similar to the $L_q$-norm PLR, where the penalty term is $\lambda\sum_{j=1}^{p}|\beta_j|^q$. On the other hand, as $c \to 0$, the formulation will be similar to the OLS subject to $L_\infty$-norm. Geometrically, $L_\infty$-norm would be square-shaped for two predictors and cube-shaped for three predictors. Moreover, when $0 \ll c \ll \infty$, the penalty term in the formulation will be a combination of infinite norms. For example, setting $q = 2$ will have an infinite combination of norms, ranging from $L_2$-norm (when $c \to \infty$) until $L_\infty$-norm (when $c \to 0$).

Now consider the formulation, which can be equivalently written in the penalized form as:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min}\ \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+\lambda\sigma^q\sum_{j=1}^{p}e^{\frac{|\beta_j|^q}{q!\sigma^q}}, \tag{5.4}$$

where $\lambda \geq 0$ is the penalty parameter. Multiplying $\sigma^q$ by the regularization term in the penalized form is essential to guarantee a monotonic relationship between $\lambda$ & $t$. Among this novel family of the $L_q$-exponential norms, the focus will be on the $L_1$-exponential norm. The corresponding constrained version of the linear regression problem can be written as:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min}\ \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) \qquad s.t. \quad \sum_{j=1}^{p}\left(e^{\frac{|\beta_j|}{\sigma}}-1\right)\leq e^{\frac{t}{\sigma}}-1, \tag{5.5}$$

Equivalently, the penalized formulation can be expressed as:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{min}\ \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+\lambda\sigma\sum_{j=1}^{p}e^{\frac{|\beta_j|}{\sigma}} \tag{5.6}$$

The Taylor series of the $L_1$- exponential norm PLR is as follows:

$$\sum_{j=1}^{p}\left(e^{\frac{|\beta_j|}{\sigma}}-1\right)=\sum_{j=1}^{p}\sum_{m=1}^{\infty}\frac{|\beta_j|^m}{m!\sigma^m}=\sum_{j=1}^{p}\left(\frac{|\beta_j|}{\sigma}+\frac{\beta_j^2}{2\sigma^2}+\frac{|\beta_j|^3}{6\sigma^3}+\dots\right). \tag{5.7}$$

There are three main reasons to focus on the $L_1$-exponential norm. Firstly, the norm is always convex for all $\sigma > 0$ (as can be seen from the Taylor expansion). The second reason is the fact that it exploits the sparsity due to the existence of $L_1$-norm in the first term of the Taylor expansion, i.e., the term will be $\frac{|\beta_j|}{\sigma}$. The third reason is that the $L_1$-exponential norm is the only one among its family of exponential norms that includes all the moments, i.e., all natural-number norms $m = 1, 2, \ldots, \infty$. Other class of $L_q$-exponential norms (where $q \neq 1$) does not have the three characteristics simultaneously. For example: the exponential norm for $q < 1$ is not always convex, since its shape can be similar to $L_q$-norm. Moreover, for $q > 1$, the exponential norm will never have a sparse solution, since it consists of summation of non-sparse norms. Lastly, for any value of $q$ (say $q = 2$), the exponential norm will only include even-number norms $m = 2, 4, \ldots, \infty$. Table 5.2 presents a brief summary of the above discussion.

Table 5.2: The exponential norm status of convexity, sparsity, & moments with different values of $q$.

| **$L_q$-exponential norm** | $q > 1$ | $q < 1$ | $q = 1$ |
|---|---|---|---|
| **Convexity** | Always convex | Not always convex | Always convex |
| **Sparsity** | Non-sparse | Sparse | Sparse |
| **Moments** | not all | not all | all moments |

Now, consider the first two terms as an approximation of the $L_1$-exponential norm:

$$\sum_{j=1}^{p} \left( e^{\frac{|\beta_j|}{\sigma}} - 1 \right) \approx \sum_{j=1}^{p} \left( \frac{|\beta_j|}{\sigma} + \frac{\beta_j^2}{2\sigma^2} \right) \tag{5.8}$$

Clearly, the fixed-shape elastic net is an approximation of the $L_1$-exponential norm. Similar to the fixed-shape elastic net, it is important to reformulate the $L_1$-exponential norm as well. From the $L_1$-exponential norm approximation, it can be seen that the relationship between $\lambda$ and $t$ is similar to the fixed-shape elastic net. Hence multiplying by $\sigma$ is essential to ensure a monotonic relationship between $\lambda$ and $t$. This will avoid having two corresponding $t$'s for one value of $\lambda$. See Figure (4.15) and Figure (4.16) as an illustration for the relationship.

In $L_1$-exponential norm, the tuning parameter $\lambda$ or $t$ controls the size of the norm, while $c$ controls the shape of the norm. To have a broader idea about the role that $c$ plays in shaping the exponential norm, see Table (5.3). When $c$ gets smaller, the presence of higher norms will be dominant w.r.t the changes in $c$ (See Figure (5.1)).

Table 5.3: The compositions of norms at different values of $c$ for the $L_1$-exponential norm

| $c$ | $L_1$-norm | $L_2$-norm | $L_3$-norm | $L_4$-norm | $L_5$-norm | Rest |
|---:|---:|---:|---:|---:|---:|---:|
| 1000000 | **100.00%** | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| 1000 | **99.95%** | 0.05% | 0.00% | 0.00% | 0.00% | 0.00% |
| 100 | **99.50%** | 0.50% | 0.00% | 0.00% | 0.00% | 0.00% |
| 5 | **90.33%** | 9.03% | 0.60% | 0.03% | 0.00% | 0.00% |
| 4 | **88.02%** | 11.00% | 0.92% | 0.06% | 0.00% | 0.00% |
| 3 | **84.26%** | 14.04% | 1.56% | 0.13% | 0.01% | 0.00% |
| 2 | **77.07%** | 19.27% | 3.21% | 0.40% | 0.04% | 0.00% |
| 1 | **58.20%** | 29.10% | 9.70% | 2.42% | 0.48% | 0.09% |
| 0.8 | **50.19%** | 31.37% | 13.07% | 4.08% | 1.02% | 0.26% |
| 0.6 | **38.81%** | 32.34% | 17.97% | 7.49% | 2.50% | 0.90% |
| 0.5 | **31.30%** | **31.30%** | 20.87% | 10.43% | 4.17% | 1.92% |
| 0.4 | 22.36% | **27.95%** | 23.29% | 14.55% | 7.28% | 4.58% |
| 0.3 | 12.33% | 20.55% | **22.84%** | 19.03% | 12.69% | 12.56% |
| 0.25 | 7.46% | 14.93% | **19.90%** | **19.90%** | 15.92% | 21.89% |
| 0.2 | 3.39% | 8.48% | 14.13% | **17.67%** | **17.67%** | 38.66% |
| 0.1 | 0.05% | 0.23% | 0.76% | 1.89% | 3.78% | 93.30% |
| 0.01 | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 100.00% |

In addition to that, for a fixed value of $c$, the shape of the norm will be fixed, regardless of the value of the tuning parameter $t$. On the other hand, the parameter

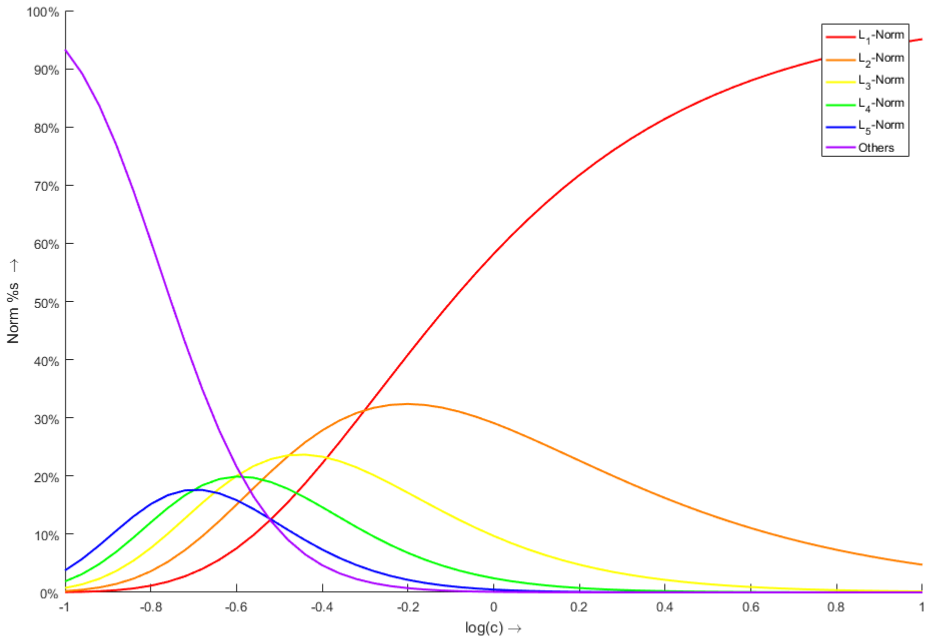

Figure 5.1: $y$-axis represents the composition of each norm. $x$-axis represents the shape controller $c$.

$t$ will control the size of the norm (See Figure (5.2)). The trace plots for different values of $c$ are shown in Figure (5.3) to Figure (5.10).

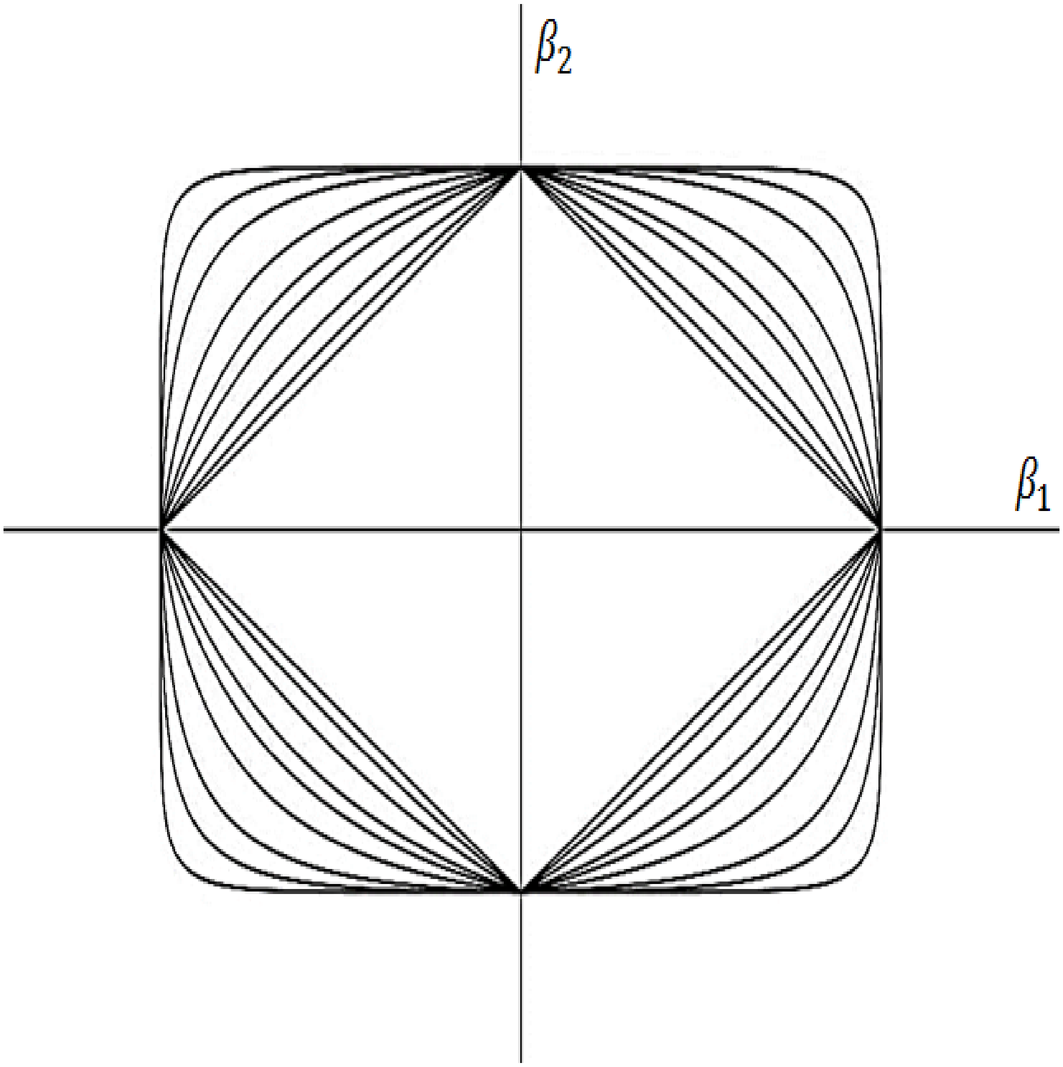

Figure 5.2: For two predictors, for the same value of $t$, the $L_1$-exponential norm with different values of $c$ from inside are $10^6$, 5, 2, 1, 0.75, 0.5, 0.3, 0.2, 0.1, respectively.

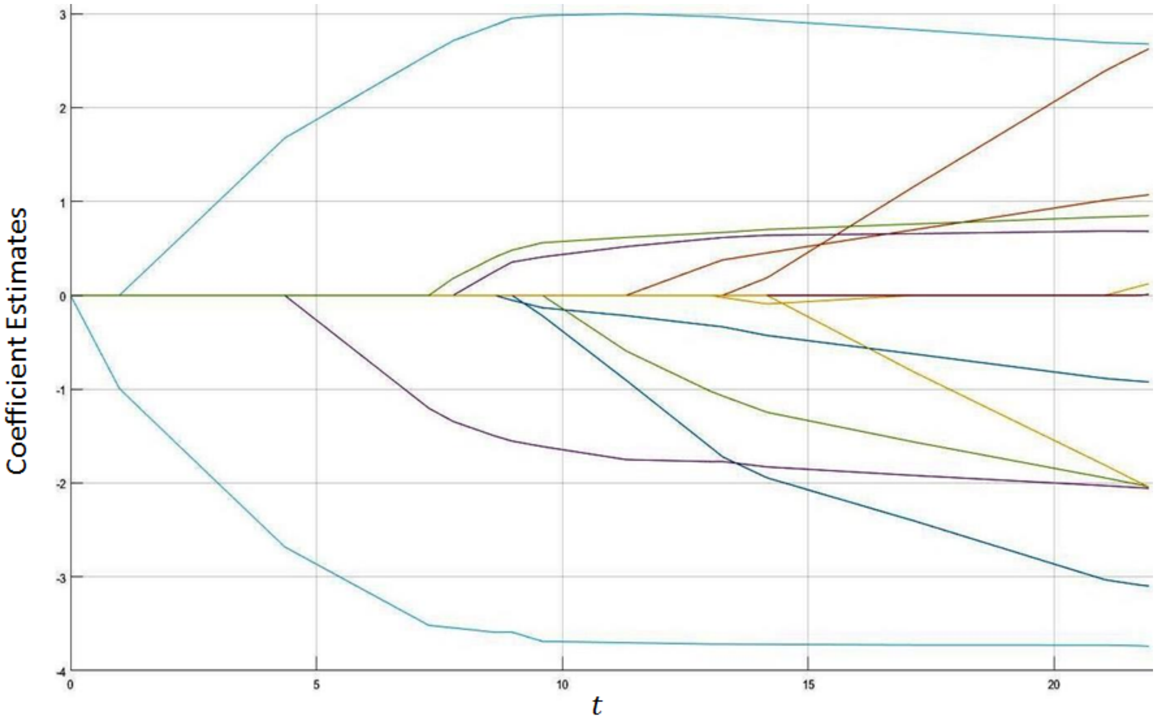

Figure 5.3: Trace plot of Boston dataset by the $L_1$-exponential norm with $c = 1000$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

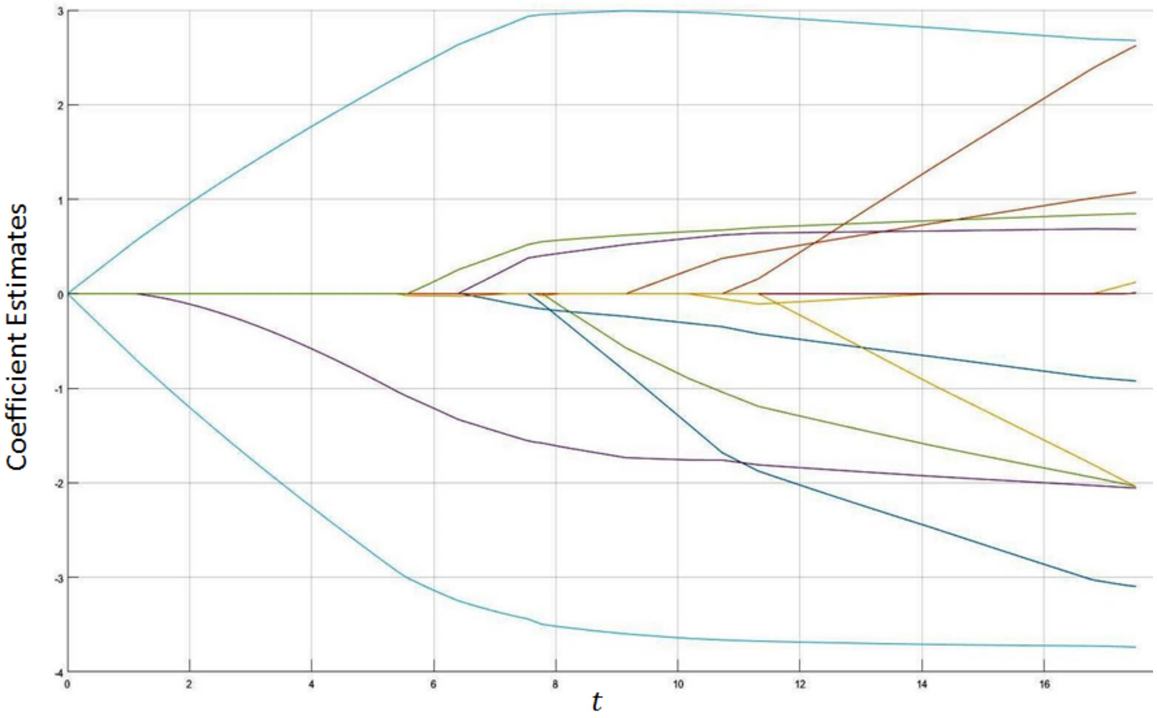

Figure 5.4: Trace plot of Boston dataset by the $L_1$-exponential norm with $c = 2$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

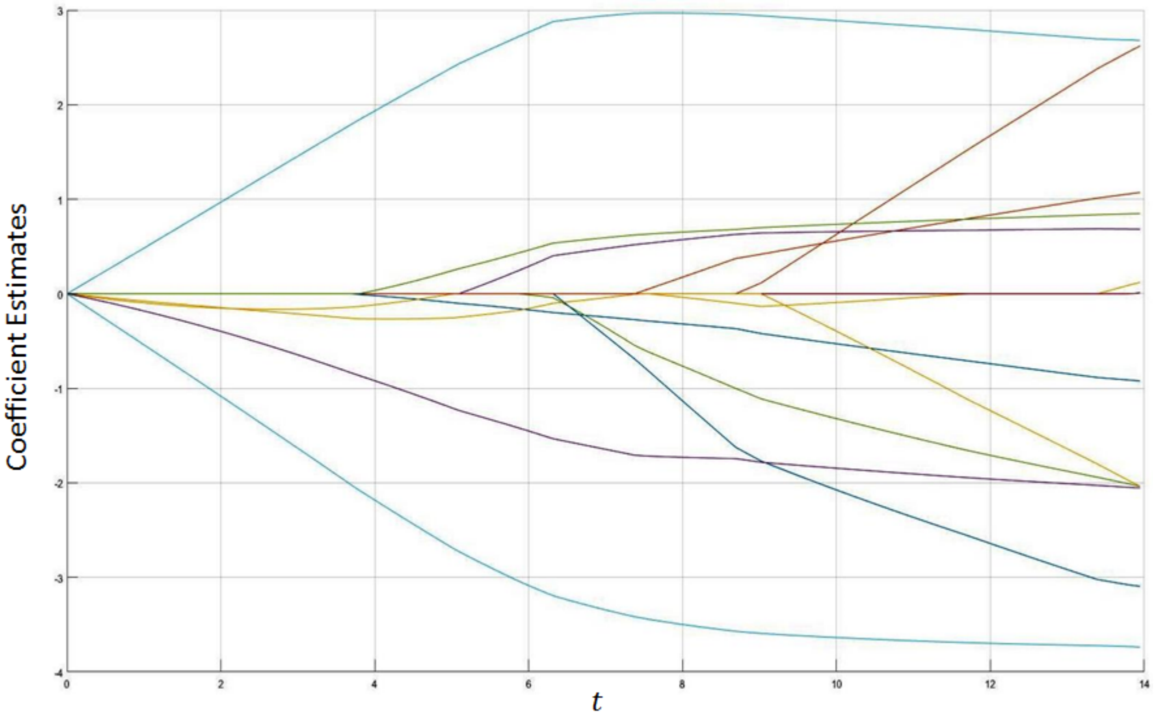

Figure 5.5: Trace plot of Boston dataset by the $L_1$-exponential norm with $c = 1$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

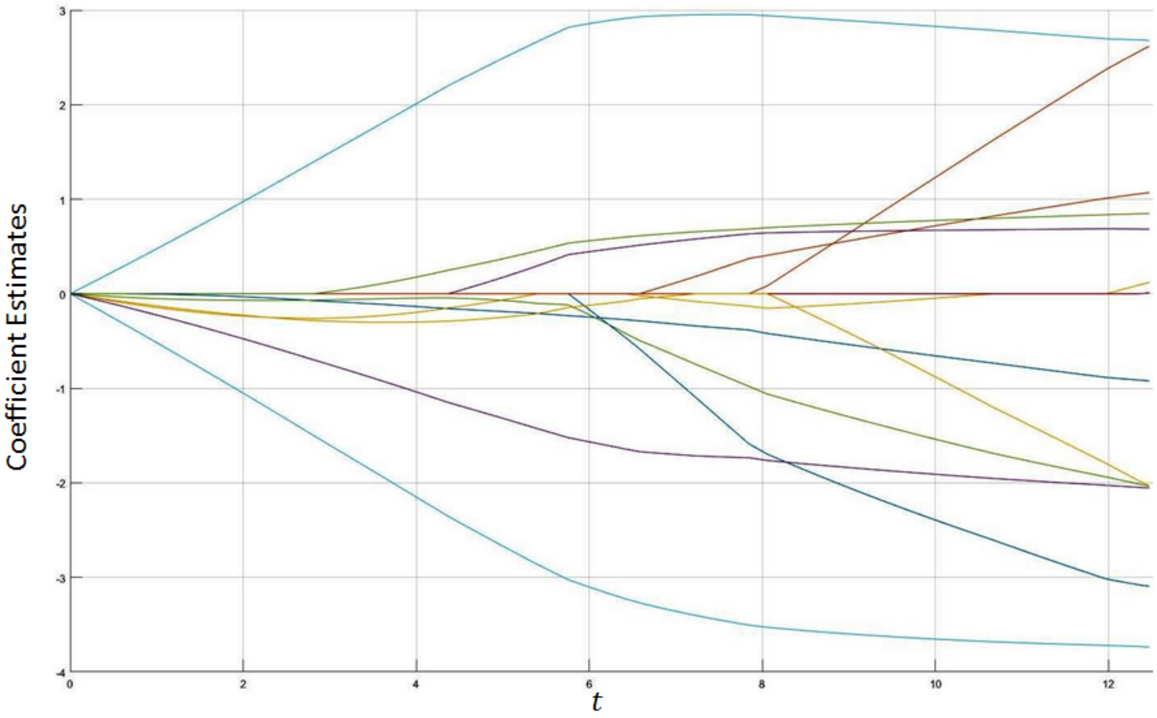

Figure 5.6: Trace plot of Boston dataset by the $L_1$-exponential norm with $c = 0.8$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

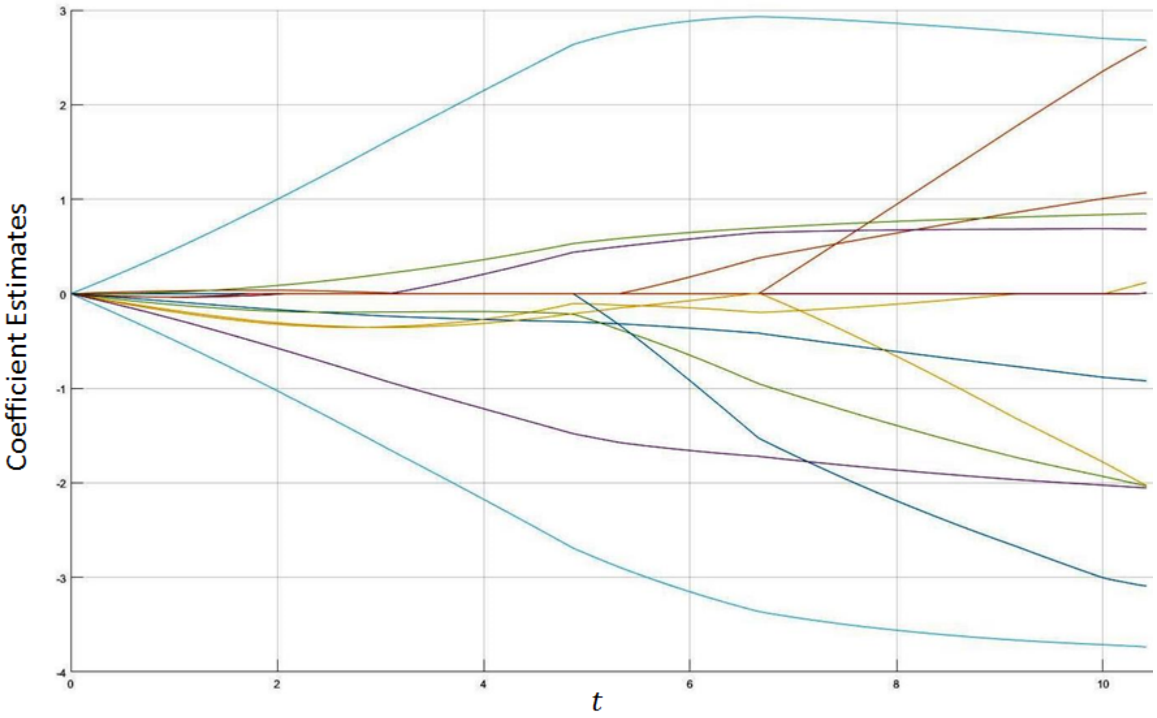

Figure 5.7: Trace plot of Boston dataset by the $L_1$-exponential norm with $c = 0.6$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

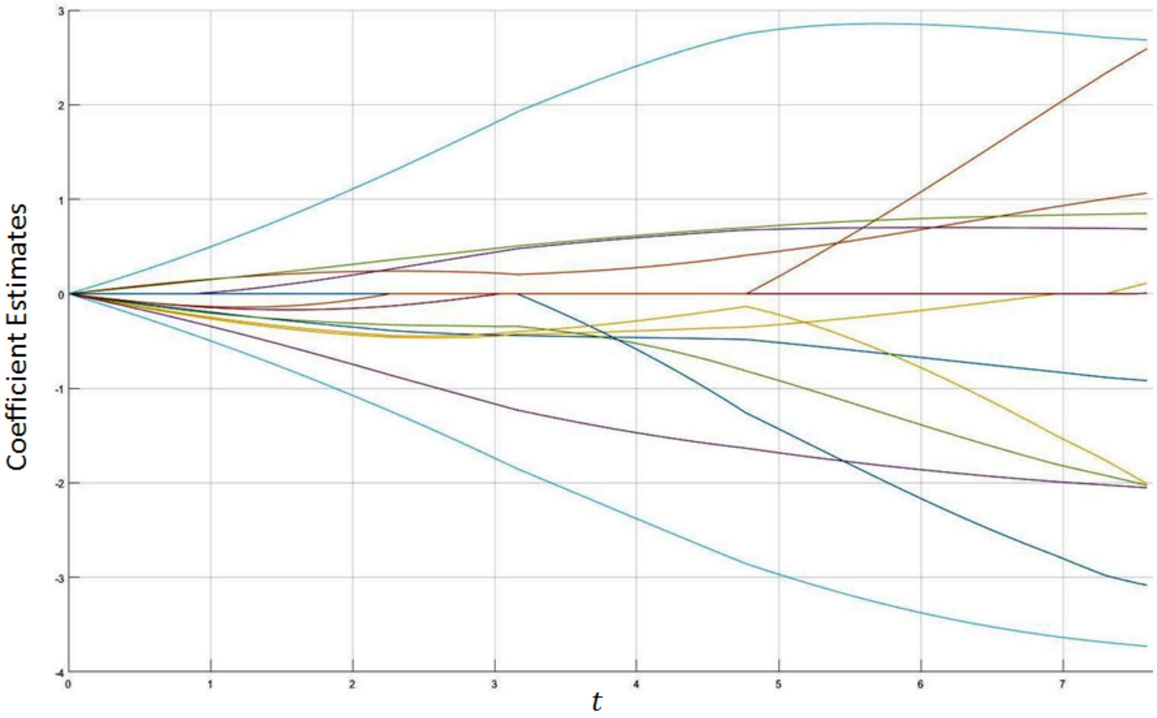

Figure 5.8: Trace plot of Boston dataset by the $L_1$-exponential norm with $c = 0.4$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

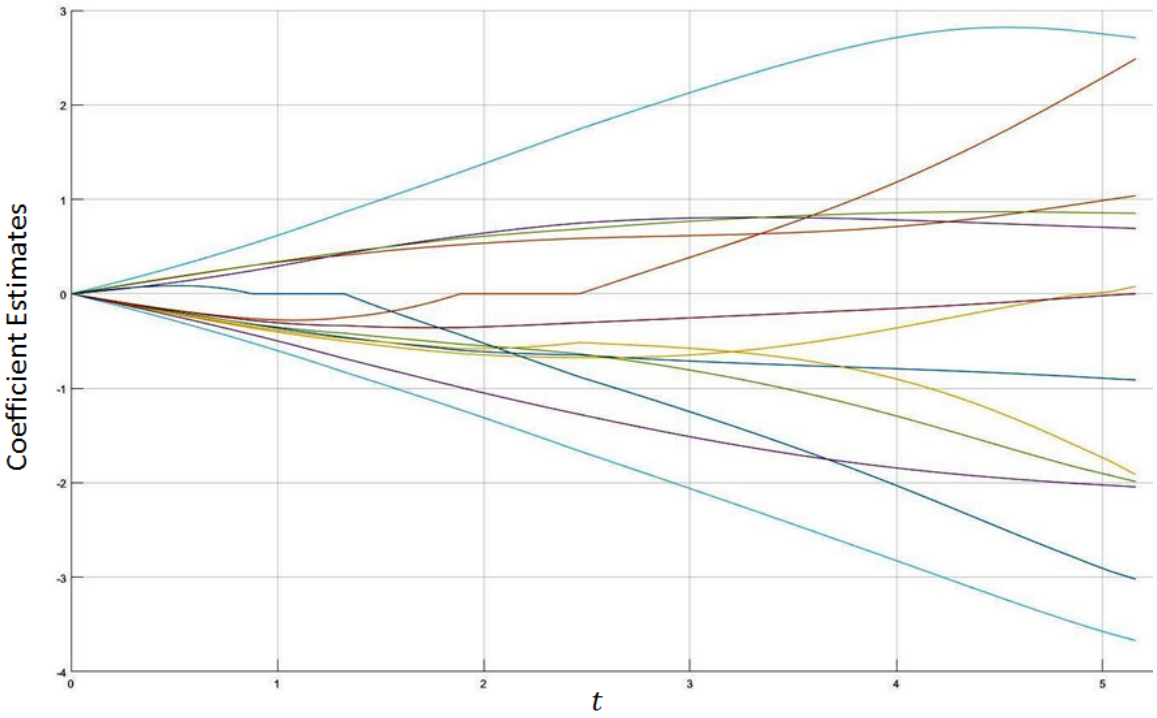

Figure 5.9: Trace plot of Boston dataset by the $L_1$-exponential norm with $c = 0.25$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

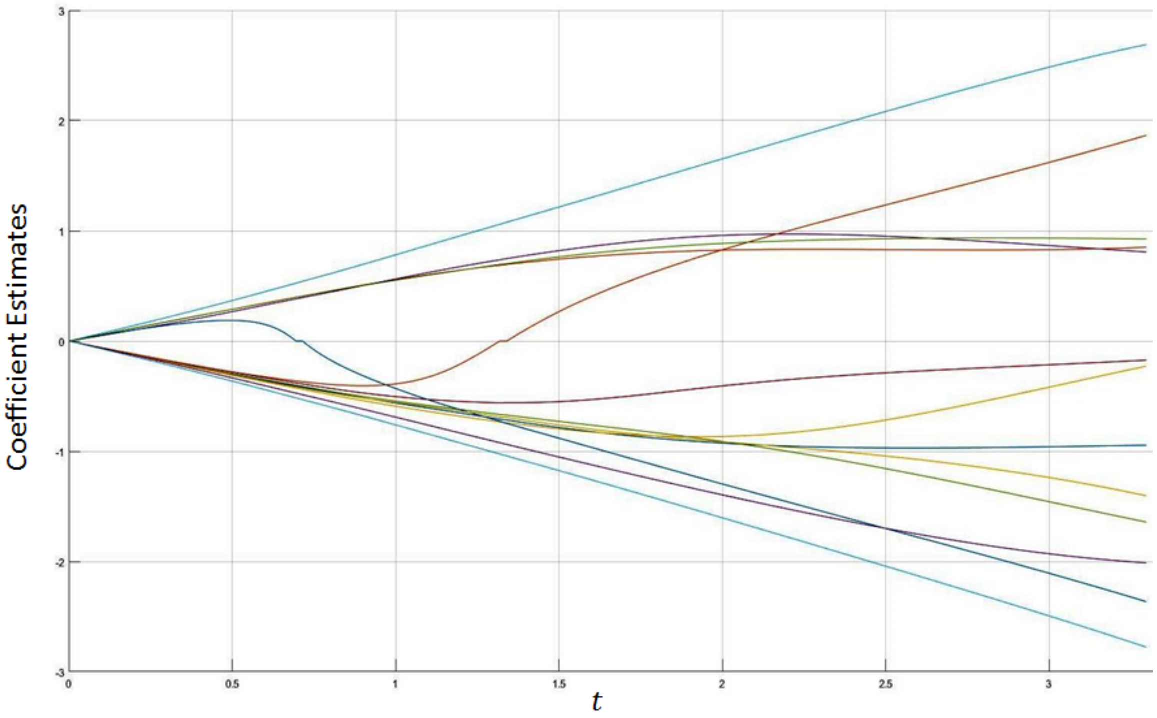

Figure 5.10: Trace plot of Boston dataset by the $L_1$-exponential norm with $c = 0.15$. The $x$-axis is the tuning parameter $t$. The $y$-axis indicates the values of the coefficient estimates.

## 5.2 Bayes Estimates

The penalized linear regression estimates can be interpreted as the mode of the Bayes posterior estimates under prior. The lasso corresponds to the Laplacian prior, as follows:

$$\boldsymbol{\beta}|\lambda \sim \prod_{j=1}^{p} \frac{\lambda}{2} e^{-\lambda|\beta_j|}. \tag{5.9}$$

Park and Casella [37] presented an extensive comparison between the Bayesian lasso and the ordinary lasso. The ridge regression corresponds to the Gaussian prior, as follows:

$$\boldsymbol{\beta}|\lambda \sim \prod_{j=1}^{p} \frac{\lambda}{2} e^{-\lambda\beta_j^2}. \tag{5.10}$$

The elastic net corresponds to a combination of the Laplacian and Gaussian, as follows:

$$\boldsymbol{\beta}|\lambda \sim \prod_{j=1}^{p} \frac{\lambda}{2} e^{-\lambda\left(\alpha|\beta_j|+\frac{(1-\alpha)}{2}\beta_j^2\right)}. \tag{5.11}$$

The $L_1$-exponential net corresponds to the new prior, as follows:

$$\boldsymbol{\beta}|\lambda \sim \prod_{j=1}^{p} \frac{\lambda}{2} e^{-\lambda\left(e^{\frac{|\beta_j|}{\sigma}}\right)} \tag{5.12}$$

## 5.3 Solution Algorithm for the $L_1$-exponential norm

A coordinate descent algorithm based approach is proposed to solve the $L_1$-exponential norm PLR. The algorithm is described in the following steps:

**Input**: The outer iteration counter $v$, the penalty parameter $\lambda^{(v)}$, and the shape parameter $c$.

**Step 1**: Set $r = 0$, the iteration counter. Initialize $\boldsymbol{\beta}^r_{exp}$ as follows: if $v = 1$, then $\boldsymbol{\beta}^r_{exp} = \hat{\boldsymbol{\beta}}_o$, otherwise $\boldsymbol{\beta}^r_{exp}|_v = \hat{\boldsymbol{\beta}}_{exp}|_{(v-1)}$. where $\boldsymbol{\beta}^r_{exp}$ is the $r^{th}$ update for the $L_1$-exponential norm estimate $\hat{\boldsymbol{\beta}}_{exp}$

**Step 2**: Set $j = 1$.

**Step 3**: Solve $t_j^{r+1}$ as per Equation (5.13), except when all $\beta^{j,exp}_{r+1} = 0$.

$$t_j^{r+1} = ct_j^{r+1}\, ln\left(\sum_{l=1}^{p}\left(e^{\frac{|\beta^*_{l,exp}|}{ct_j^{r+1}}} - 1\right) + 1\right) \tag{5.13}$$

where $\beta^*_{l,exp}$ is the current or recent updated value, i.e., for $1 \leq l < j$, $* = r + 1$ and for $j \leq l \leq p$, $* = r$.

**Step 4**: Find $\beta^{r+1}_{j,exp}$ as per the equation solver approach given by Equation (5.25) or the updater approach given by Equation (5.39)

**Step 5**: $j = j + 1$. If $j \leq p$, then repeat Step 3 and Step 4. If $j = p + 1$, then set $r = r + 1$. If termination criterion is met, then stop, otherwise go to Step 2.

The above algorithm is repeated by updating the value of $v$ as $v = v + 1$, and $\lambda^{(v+1)} > \lambda^{(v)}$.

The KKT conditions of the new formulation will be:

**Optimality condition:**

$$\nabla S(\boldsymbol{\beta}) + \lambda \nabla g(\boldsymbol{\beta}) = \mathbf{0}, \qquad \beta_j \neq 0 \tag{5.14}$$

$$\begin{bmatrix} -\sum_{i=1}^{N}\left(x_{i1}\left(y_i - \sum_{j=1}^{p}\beta_j x_{ij}\right)\right) \\ -\sum_{i=1}^{N}\left(x_{i2}\left(y_i - \sum_{j=1}^{p}\beta_j x_{ij}\right)\right) \\ \vdots \\ -\sum_{i=1}^{N}\left(x_{ip}\left(y_i - \sum_{j=1}^{p}\beta_j x_{ij}\right)\right) \end{bmatrix} + \lambda \begin{bmatrix} sign\big(\beta_1\big)\, e^{\frac{\beta_1}{\sigma}} \\ sign\big(\beta_2\big)\, e^{\frac{\beta_2}{\sigma}} \\ \vdots \\ sign\big(\beta_p\big)\, e^{\frac{\beta_p}{\sigma}} \end{bmatrix} = \mathbf{0}, \qquad \beta_j \neq 0 \tag{5.15}$$

**Dual feasibility:**

$$\lambda \geq 0 \tag{5.16}$$

**Primal feasibility:**

$$g(\boldsymbol{\beta}) \ \leq 0 \tag{5.17}$$

$$\sigma \sum_{j=1}^{p}\left(e^{\frac{|\beta_j|}{\sigma}} - 1\right) - \sigma\big(e^{\frac{t}{\sigma}} - 1\big) \leq 0 \tag{5.18}$$

**Complementary slackness:**

$$\lambda\, g(\boldsymbol{\beta}) \leq 0 \tag{5.19}$$

$$\lambda\left(\sigma \sum_{j=1}^{p}\left(e^{\frac{|\beta_j|}{\sigma}} - 1\right) - \sigma\big(e^{\frac{t}{\sigma}} - 1\big)\right) = 0 \tag{5.20}$$

The $L_1$-exponential norm requires the norm shape to be fixed. This is done in order to explore its full capacity. When $\lambda > 0$, then there is a corresponding $t$, such that the following constraint is active or binding, i.e.:

$$\sum_{j=1}^{p}\left(e^{\frac{|\beta_j|}{\sigma}}-1\right)=\left(e^{\frac{t}{\sigma}}-1\right) \tag{5.21}$$

$$t=\sigma\, ln\left(\sum_{j=1}^{p}\left(e^{\frac{|\beta_j|}{\sigma}}-1\right)+1\right) \tag{5.22}$$

The coordinate descent algorithm is still applicable since the loss function is convex & differentiable, and the penalized term is convex, continuous in its domain & separable. Since the function is not quadratic, the direct coordinate newton updates can have a different number of iterations than naive updates, covariant updates, or indirect coordinate newton updates (the one that updates $z_j^{r+1}$ by the coordinate newton update). Now the stationary point of Formulation (5.6) should satisfy the following:

$$-\sum_{i=1}^{N}x_{ij}\left(y_i-\sum_{l=1}^{p}\beta_l x_{il}\right)+\lambda\, sign\big(\beta_j\big)\, e^{\frac{|\beta_j|}{\sigma}}=0, \quad \beta_j\neq 0,\ \forall\, j. \tag{5.23}$$

In general, if $\beta_{j,exp}^{r+1}\neq 0$, the update $\beta_{j,exp}^{r+1}$ will be defined as:

$$\beta_{j,exp}^{r+1}=z_j^{r+1}-sign\left(\beta_{j,exp}\right)\frac{\lambda}{N}e^{\frac{|\beta_{j,exp}|}{\sigma}}. \tag{5.24}$$

where $\beta_{j,exp}$ on the right hand side of Equation (5.24) can be the $r^{th}$ or $(r+1)^{th}$ update depending on the approach of solving. There are two approaches proposed for solving the $L_1$-exponential norm: equation solver approach, and updater approach.

### 5.3.1 Equation Solver Approach

From Equation (5.24), $z_j^{r+1}$ can be obtained by any type of coordinate descent update mechanisms. Variable $\beta_{j,exp}^{r+1}$ cannot have an explicit mathematical expression. The equation-solver approach requires solving for the equation of one variable $\beta_{j,exp}^{r+1}$ by some line search technique, like golden-section search, or bisection search. The equation solver approach considers $\beta_{j,exp}^{r+1}$ on both sides of the equation as a variable for the $(r+1)^{th}$ update.

$$\beta_{j,exp}^{r+1} = sign(z_j^{r+1})\Bigg(|z_j^{r+1}| - \frac{\lambda}{N}e^{\frac{|\beta_{j,exp}^{r+1}|}{\sigma}}\Bigg)_+, \tag{5.25}$$

The sign and magnitude of $z_j^{r+1}$ will determine the sign and magnitude of $\beta_{j,exp}^{r+1}$. If $|z_j^{r+1}| \leq \frac{\lambda}{N}$, then the only feasible solution is $\beta_{j,exp}^{r+1}$ equal to 0. If $|z_j^{r+1}| > \frac{\lambda}{N}$ , then there will be two scenarios:

**Scenario 1**

If $z_j^{r+1} > \frac{\lambda}{N}$, then consider $R_j^+(\beta_{j,exp}^{r+1})$ that is defined as:

$$R_j^+(\beta_{j,exp}^{r+1}) = \beta_{j,exp}^{r+1} - z_j^{r+1} + \frac{\lambda}{N}e^{\frac{\beta_{j,exp}^{r+1}}{\sigma}} = 0 \tag{5.26}$$

Observe that $R_j^+$ is monotonically increasing w.r.t $\beta_{j,exp}^{r+1}$, since $\left.\dfrac{dR_j^+}{d\beta_j}\right|_{\beta_j=\beta_{j,exp}^{r+1}} = 1 + \dfrac{\lambda}{\sigma N} e^{\dfrac{\beta_{j,exp}^{r+1}}{\sigma}} > 0 \quad \forall\ \beta_{j,exp}^{r+1}$. Also, notice that $R_j^+(0) < 0$ and $R_j^+(z_j^{r+1}) > 0$. Thus, $\beta_{j,exp}^{r+1} \in (0,\ z_j^{r+1})$.

**Scenario 2**

If $z_j^{r+1} < -\dfrac{\lambda}{N}$, then consider $R_j^-(\beta_{j,exp}^{r+1})$ that is defined as:

$$R_j^-(\beta_{j,exp}^{r+1}) = \beta_{j,exp}^{r+1} - z_j^{r+1} - \frac{\lambda}{N} e^{\dfrac{-\beta_{j,exp}^{r+1}}{\sigma}} = 0 \tag{5.27}$$

Similarly, $R_j^-$ is monotonically increasing w.r.t. $\beta_{j,exp}^{r+1}$, since $\left.\dfrac{dR_j^-}{d\beta_j}\right|_{\beta_j=\beta_{j,exp}^{r+1}} = 1 + \dfrac{\lambda}{\sigma N} e^{\dfrac{-\beta_{j,exp}^{r+1}}{\sigma}} > 0 \quad \forall\ \beta_{j,exp}^{r+1}$. Also, notice that $R_j^-(0) > 0$ and $R_j^-(z_j^{r+1}) < 0$. Thus, $\beta_{j,exp}^{r+1} \in (z_j^{r+1},\ 0)$.

Iteratively, the algorithm will be excuted for $j = 1, 2, \ldots p, 1, 2, \ldots p, 1 \ldots$ until the convergence.

### 5.3.2 Updater Approach

This approach considers $\beta_{j,exp}$ on the left hand side of Equation (5.24) to be the new update $(r + 1)$. And $\beta_{j,exp}$ on the right hand side of Equation (5.24), which is inside the exponential term, to be the old update $\beta_{j,exp}^r$. Moreover, we assume that the coefficient update does not change the sign abruptly, i.e.:

If $\beta^r_{j,exp} > 0$, then $\beta^{r+1}_{j,exp} \geq 0$. If $\beta^r_{j,exp} < 0$, then $\beta^{r+1}_{j,exp} \leq 0$. If $\beta^r_{j,exp} = 0$, then $\beta^{r+1}_{j,exp} \in \mathbb{R}$.

One can rewrite Equation (5.24) for the updater approach as follows:

$$\beta^{r+1}_{j,exp} = sign(z^{r+1}_j)\Bigg(|z^{r+1}_j| - \frac{\lambda}{N}e^{\frac{|\beta^r_{j,exp}|}{\sigma}}\Bigg)_+, \tag{5.28}$$

$z^{r+1}_j$ can be obtained by naive, covariant, or indirect newton coordinate updates. However, this method is not guaranteed to converge when $\sigma$ becomes smaller and higher order norms are not negligible. This is due to abrupt increase in the step size as can be seen by the counterexample (See Figure (5.11)). The direct newton coordinate update has the capability to deal with the situation when $\sigma$ becomes smaller, which is described, when $\beta_j \neq 0$, as follows:

$$\beta^{r+1}_j = \beta^r_j - \frac{\dfrac{\partial}{\partial \beta_j}\Big(S(\boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta})\Big)}{\dfrac{\partial^2}{\partial {\beta_j}^2}\Big(S(\boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta})\Big)} \tag{5.29}$$

$$\frac{\partial}{\partial \beta_j}\Big(S(\boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta})\Big) = -\sum_{i=1}^{N} x_{ij}\Big(y_i - \sum_{l=1}^{p} \beta_l x_{il}\Big) + \lambda\, sign\big(\beta_j\big)\, e^{\frac{|\beta_j|}{\sigma}} \tag{5.30}$$

$$\frac{\partial^2}{\partial {\beta_j}^2}\Big(S(\boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta})\Big) = N + \frac{\lambda}{\sigma} e^{\dfrac{|\beta_j|}{\sigma}} \tag{5.31}$$

If $\beta_{j,exp}^{r+1} > 0$, then let $\beta_{j,exp}^{(+)r+1}$ the positive update be defined as:

$$\beta_{j,exp}^{(+)r+1} = \beta_{j,exp}^{r} + \frac{\sum_{i=1}^{N} x_{ij}\Big(y_i - \sum_{l=1}^{p} \beta_l^* x_{il}\Big) - \lambda e^{\dfrac{\beta_{j,exp}^{r}}{\sigma}}}{N + \dfrac{\lambda}{\sigma} e^{\dfrac{\beta_{j,exp}^{r}}{\sigma}}}. \tag{5.32}$$

If $\beta_{j,exp}^{r+1} > 0$, then let $\beta_{j,exp}^{(-)r+1}$ the negative update be defined as:

$$\beta_{j,exp}^{(-)r+1} = \beta_{j,exp}^{r} + \frac{\sum_{i=1}^{N} x_{ij}\Big(y_i - \sum_{l=1}^{p} \beta_l^* x_{il}\Big) + \lambda e^{\dfrac{-\beta_{j,exp}^{r}}{\sigma}}}{N + \dfrac{\lambda}{\sigma} e^{\dfrac{-\beta_{j,exp}^{r}}{\sigma}}}, \tag{5.33}$$

otherwise $\beta_{j,exp}^{r+1} = 0$. Then:

If $\beta_{j,exp}^{(+)r+1} > 0$, $\beta_{j,exp}^{(-)r+1} > 0$, then set $\beta_{j,exp}^{r+1} = \beta_{j,exp}^{(+)r+1}$.

If $\beta_{j,exp}^{(+)r+1} < 0$, $\beta_{j,exp}^{(-)r+1} < 0$, then set $\beta_{j,exp}^{r+1} = \beta_{j,exp}^{(-)r+1}$.

Otherwise, set $\beta_{j,exp}^{r+1} = 0$.

Looking closely at the direct coordinate newton updates in the $L_1$-exponential norm, it is not equivalent to naive, covariance, or indirect coordinate newton updates. This is because the $L_1$-exponential norm, unlike the lasso, ridge regression, elastic net, is not quadratic. Hence, one iteration of the direct newton coordinate will move to the improving direction but will not necessarily result $\dfrac{\partial f(\boldsymbol{\beta})}{\partial \beta_j} = 0 \quad \forall\, \beta_j \neq 0.$

The direct coordinate newton can also be expressed as:

If $\beta^{r+1}_{j,exp} > 0$, then let $\beta^{(+)r+1}_{j,exp}$ the positive update be defined as:

$$\beta^{(+)r+1}_{j,exp} = \beta^{r}_{j,exp} + \frac{\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{l\neq j}\beta^{*}_{l}x_{il}\right) - N\beta^{r}_{j,exp} - \lambda e^{\frac{\beta^{r}_{j,exp}}{\sigma}}}{N\left(1 + \frac{\lambda}{N\sigma}e^{\frac{\beta^{r}_{j,exp}}{\sigma}}\right)}, \tag{5.34}$$

$$\beta^{(+)r+1}_{j,exp} = \beta^{r}_{j,exp} + \frac{z^{r+1}_{j} - \beta^{r}_{j,exp} - \frac{\lambda}{N}e^{\frac{\beta^{r}_{j,exp}}{\sigma}}}{1 + \frac{\lambda}{N\sigma}e^{\frac{\beta^{r}_{j,exp}}{\sigma}}}, \tag{5.35}$$

$$\beta^{(+)r+1}_{j,exp} = \frac{z^{r+1}_{j} + \frac{\lambda}{N\sigma}e^{\frac{\beta^{r}_{j,exp}}{\sigma}}\beta^{r}_{j,exp} - \frac{\lambda}{N}e^{\frac{\beta^{r}_{j,exp}}{\sigma}}}{1 + \frac{\lambda}{N\sigma}e^{\frac{\beta^{r}_{j,exp}}{\sigma}}}, \tag{5.36}$$

$$\beta^{(+)r+1}_{j,exp} = \frac{z^{r+1}_{j} - \frac{\lambda}{N}e^{\frac{\beta^{r}_{j,exp}}{\sigma}}\left(1 - \frac{\beta^{r}_{j,exp}}{\sigma}\right)}{1 + \frac{\lambda}{N\sigma}e^{\frac{\beta^{r}_{j,exp}}{\sigma}}}, \tag{5.37}$$

If $\beta_{j,exp}^{r+1} < 0$, then let $\beta_{j,exp}^{(-)r+1}$ the negative update be defined as:

$$\beta_{j,exp}^{(-)r+1} = \frac{z_j^{r+1} + \dfrac{\lambda}{N} e^{\dfrac{-\beta_{j,exp}^{r}}{\sigma}} \left(1 + \dfrac{\beta_{j,exp}^{r}}{\sigma}\right)}{1 + \dfrac{\lambda}{N\sigma} e^{\dfrac{-\beta_{j,exp}^{r}}{\sigma}}}, \tag{5.38}$$

otherwise $\beta_{j,exp}^{r+1} = 0$. In general:

$$\beta_{j,exp}^{r+1} = \frac{sign\big(z_j^{r+1}\big)\left(|z_j^{r+1}| - \dfrac{\lambda}{N} e^{\dfrac{|\beta_{j,exp}^{r}|}{\sigma}} \left(1 - sign\big(z_j^{r+1}\big)\dfrac{\beta_{j,exp}^{r}}{\sigma}\right)\right)_+}{1 + \dfrac{\lambda}{N\sigma} e^{\dfrac{|\beta_{j,exp}^{r}|}{\sigma}}}, \tag{5.39}$$

Iteratively, the algorithm will be executed for $j = 1, 2, \dots p, 1, 2, \dots p, 1 \dots$ until the convergence. $z_j^{r+1}$ can be found by any kind of updates then instead of using Equation (5.28), Equation (5.39) can be utilized. Figure (5.11) shows the difference between using Equation (5.28) and Equation (5.39). The reason is that when $\sigma$ becomes very small in Equation (5.28), the term $e^{\dfrac{|\beta_{j,exp}^{r}|}{\sigma}}$ becomes large and hence the step size is too large, while Equation (5.39) normalizes the step as can be seen in the denominator.

Based on trial experiments, it is observed that the equation solver approach is faster than the updater approach, in terms of the number of iterations. See the example demonstrated in Figure (5.12). However, the efficiency of the equation solver approach is less, since it requires the solution for Equation (5.25) in every

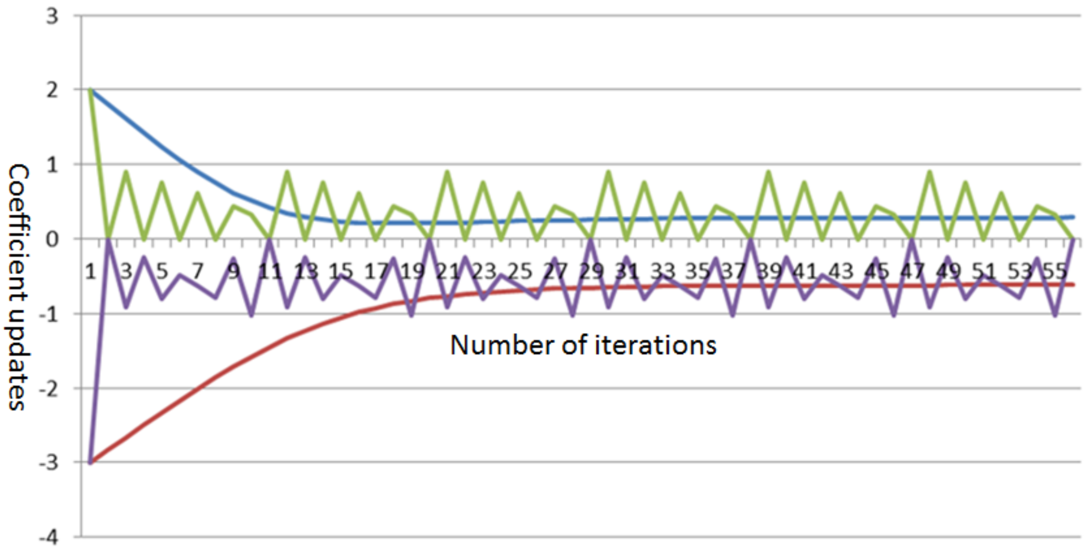

Figure 5.11: $y$-axis represents the value of the coefficient estimate. $x$-axis represents the number of iterations. Solving for two coefficient estimates, the two smooth curves are solved by updater approach; indirect coordinate newton updates, and the two zigzagging curves are solved by updater approach but direct update (not converging).

iteration.

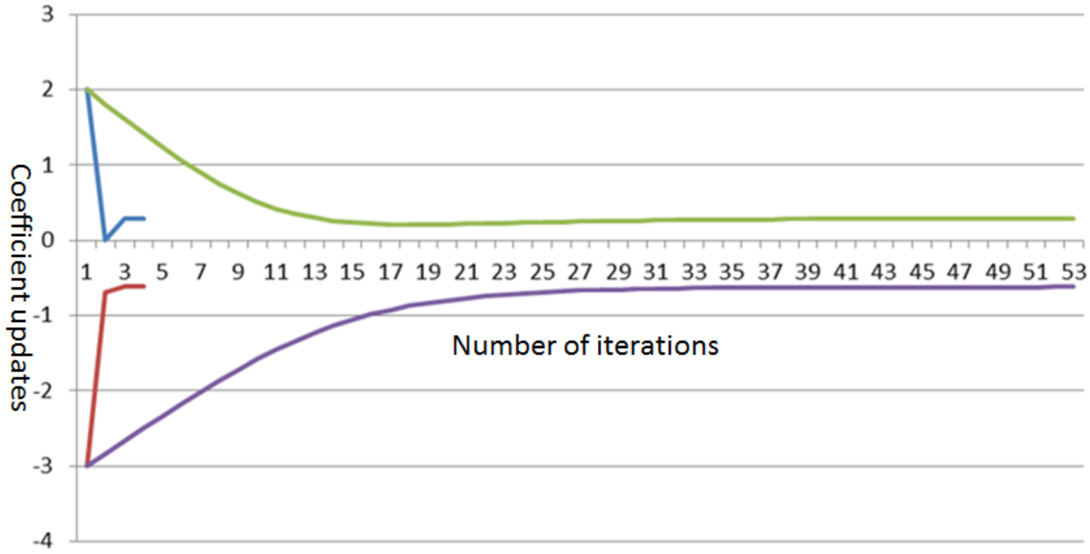

Figure 5.12: $y$-axis represents the value of the coefficient estimate. $x$-axis represents the number of iterations. Solving for two coefficient estimates, the two longer curves are solved by updater approach; indirect coordinate newton updates, and the short curves are solved by equation-solver approach.

## CHAPTER 6

# NUMERICAL EXPERIMENTATION

## 6.1 Boston Housing Data

**Data:** The dataset of Boston housing comes from a study by [38] that exploits the will of people to pay more for clean air. The response **y** is the mean value of owner occupied homes in \$1000's. There are 13 predictors (i.e., $p = 13$). There are 506 observations.

**Method:** **X** is standardized and **y** is centered before fitting linear models. A tenfold cross validation approach will be used on the 506 observations to determine the optimal tuning parameters. The coefficients will be estimated using the OLS, lasso, ridge regression, fixed-shape elastic net, and $L_1$-exponential norm methods.

**Results & Discussion:** The $L_1$-exponential norm outperforms all other methods in the example of Boston housing dataset. It is worthy to note that the fixed-shape elastic net in this example is always outperformed by the lasso, except when $c$ becomes very high and hence the fixed-shape elastic net becomes the lasso. It is a good example to show the effect of adding higher degree norms on the prediction error. Figure (6.1) shows the mean squared errors acquired from a tenfold cross validation of the 506 observations on the dataset. Both the fixed-shape elastic net and the $L_1$-exponential norm are behaving like the lasso for $c \to \infty$, but they set apart when $c \to 0$. The $L_1$-exponential norm becomes $L_\infty$-norm, while the fixed-shape elastic net becomes the ridge regression.

Table 6.1 indicates the coefficient estimates of each method. The best result is for the $L_1$-exponential norm at $c = 0.15$. The best result of the most sparse solution is for the $L_1$-exponential norm at $c = 0.4$. Although it is geometrically noticeable that the curves of such a norm with $\sigma = 0.4t$ has more exposure to the OLS function than the curves of $L_2$-norm, the $L_1$-exponential norm at $c = 0.4$ still can potentially give a sparse solution as good as the lasso due to the existence of the $L_1$-norm or, roughly speaking, the non-smooth corners. Figure (6.2) shows the norm shapes of the $L_1$-exponential norm at $c = 0.45$, and the ridge regression.

Table 6.1: Boston housing results

| Coeff.\Method | OLS | Lasso | Ridge | Exp (c=0.4) | Exp (c=0.15) | ENet (c=2) |
|---|---|---|---|---|---|---|
| 1. CRIM | -0.108 | -0.1 | -0.104 | -0.101 | -0.108 | -0.101 |
| 2. ZN | 0.046 | 0.042 | 0.043 | 0.042 | 0.041 | 0.042 |
| 3. INDUS | 0.021 | 0 | 0.006 | 0 | 0.003 | 0 |
| 4. CHAS | 2.687 | 2.689 | 2.745 | 2.732 | 2.858 | 2.693 |
| 5. NOX | -17.766 | -16.481 | -16.643 | -16.3 | -16.498 | -16.504 |
| 6. RM | 3.81 | 3.855 | 3.864 | 3.882 | 4.026 | 3.854 |
| 7. AGE | 0.001 | 0 | -0.0003 | 0 | -0.003 | 0 |
| 8. DIS | -1.476 | -1.412 | -1.414 | -1.387 | -1.335 | -1.414 |
| 9. RAD | 0.306 | 0.261 | 0.27 | 0.256 | 0.265 | 0.262 |
| 10. TAX | -0.012 | -0.01 | -0.011 | -0.01 | -0.01 | -0.01 |
| 11. PTRATIO | -0.953 | -0.933 | -0.935 | -0.93 | -0.95 | -0.933 |
| 12. B | 0.009 | 0.009 | 0.009 | 0.009 | 0.01 | 0.009 |
| 13. LSTAT | -0.525 | -0.522 | -0.516 | -0.514 | -0.461 | -0.522 |
| MSE | 23.854 | 23.804 | 23.828 | 23.797 | 23.645 | 23.805 |
| SE | 2.073 | 2.145 | 2.134 | 2.165 | 2.266 | 2.143 |
| Zero coeff. | 0 | 2 | 0 | 2 | 0 | 2 |

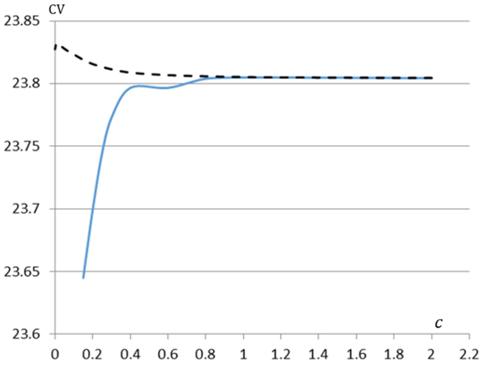

Figure 6.1: $y$-axis represents MSE from the tenfold CV, and $x$-axis represents $c$, which controls the shape of the norm.

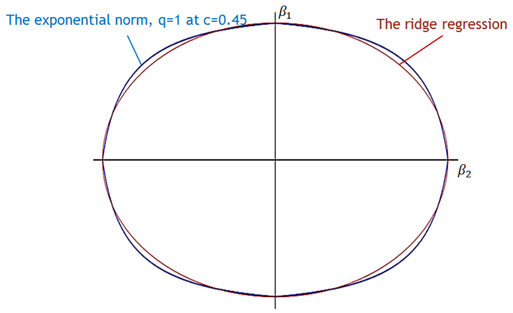

Figure 6.2: The red ball is the norm constraint of the ridge regression, and the blue ball is the $L_1$-exponential norm at c=0.45 for the same norm size.

## 6.2 Leukemia Classification Data

**Data:** Leukemia classification data by [39] is one of the well-known microarray dataset. There are two classes of Leukemia cancer: Acute Lymphoblastic Leukemia (ALL), and Acute Myeloid Leukemia (AML).

**Method:** There are two sets: training set and testing set. With 7129 genes for both sets, the training set has 38 observations ($N = 38$); 27 in ALL and 11 in AML. The testing set has 34 observations that will be used to check the prediction accuracy for models. To have an efficient solving time, 1000 genes that have the highest overall variance among 7129 genes in training set will be selected ($p = 1000$). Tenfold cross validation will be used to determine the determine the optimal tuning parameters. The coefficients will be estimated using the OLS, lasso, ridge regression, fixed-shape elastic net, and $L_1$-exponential norm methods.

**Results & Discussion:** Table 6.2 indicates the testing error for each method and each value of $c$ (if applicable). The fixed-shape elastic net and $L_1$-exponential norm have the same best prediction error. At the best prediction error, $L_1$-exponential norm selects 135 features, compared to 104 features selected by the fixed-shape elastic net. The fixed-shape elastic net, at $c = 0.5$ has the sparsest solution at the minimum testing error, i.e., the number of nonzero coefficient is 63. In any case, it is obvious that both the fixed-shape elastic net and the $L_1$-exponential norm can perform very well with $p \gg N$ case and provide various range of sparsity.

Microarray datasets usually consist of thousands of genes as predictors and much less number of observations, such that $p \gg N$. The ridge regression is not a good method for such datasets because it cannot perform variable selection. Thus, all coefficient estimates will be nonzero. On the other hand, the lasso can perform variable selection but has two limitations. The first limitation is that it cannot select features more than the number of observations, i.e. the maximum number of nonzero estimates can be selected is $N$. The second limitation is the fact that it does not perform well with correlated predictors, and thus may not select a group of correlated genes.

The novel methods: the fixed-shape elastic net and the $L_1$-exponential norm can have nonzero coefficient estimates more than $N$ for $p \gg N$ case. When $c \to \infty$, both methods will act like the lasso. When $c \to 0$, the fixed-shape elastic net will act like the ridge regression, and the $L_1$-exponential norm will act like $L_\infty$-norm. Both ridge regression and $L_\infty$-norm will have $p$ nonzero coefficient estimate, i.e., there will be no sparse solution. The shape parameter $c$ can control the level of sparsity form the lasso up to non-sparse case. They also can be solved efficiently by CDA.

The conventional elastic net is known to be a better option than the lasso, but

Table 6.2: Leukemia classification results

| Method | c | 10-fold CV error | Testing error | Number of genes |
|---|---|---|---|---|
| Golub | - | 3/38 | 4/34 | 50 |
| Lasso | - | 0/38 | 2/34 | 37 |
| Fixed-Shape EN | 1 | 0/38 | 2/34 | 49 |
| | 0.75 | 0/38 | 2/34 | 54 |
| | 0.50 | 0/38 | 1/34 | 63 |
| | 0.35 | 0/38 | 1/34 | 66 |
| | 0.20 | 0/38 | 1/34 | 104 |
| | 0.10 | 0/38 | 2/34 | 199 |
| | 0.05 | 0/38 | 2/34 | 328 |
| | 0.01 | 0/38 | 2/34 | 789 |
| | 0.05 | 0/38 | 2/34 | 889 |
| | 0.001 | 0/38 | 3/34 | 971 |
| $L_1$-Exponential Norm | 2 | 0/38 | 2/34 | 83 |
| | 1 | 0/38 | 1/34 | 87 |
| | 0.80 | 0/38 | 1/34 | 90 |
| | 0.60 | 0/38 | 1/34 | 91 |
| | 0.40 | 0/38 | 1/34 | 114 |
| | 0.30 | 0/38 | 1/34 | 135 |
| | 0.25 | 0/38 | 2/34 | 204 |
| | 0.20 | 0/38 | 2/34 | 321 |
| | 0.17 | 0/38 | 2/34 | 479 |
| | 0.15 | 0/38 | 3/34 | 639 |
| | 0.13 | 0/38 | 3/34 | 825 |
| | 0.10 | 0/38 | 3/34 | 974 |

the shape is influenced by the tuning parameter $t$, as shown earlier. The fused lasso can have different groups of variable selection ([16]). The fused lasso cannot be solved using CDA. Also, it requires "linkage hierarchical clustering" to make an order for the fusion before solving microarray dataset. However, the results of the proposed methods are better than the fused lasso's result.

## 6.3 Simulated Data

The aim of this study is to compare lasso, ridge regression, fixed-shape elastic net and the $L_1$-exponential norm with simulated data. Examples 1, 2, and 3 are taken from [5] and Example 4 is taken from [6]. Example 5 is created as a modification of Example 4, where every correlated group of predictors is partially correlated to another group.

**Data:** The data is generated from the following model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \omega\boldsymbol{\varepsilon}, \tag{6.1}$$

where $\boldsymbol{\varepsilon} \sim N(0, 1)$ and $\omega > 0$ sets the noise to signal ratio.

The details of the five examples are as follows:

1. For Example 1, let $\boldsymbol{\beta} = (3 \quad 1.5 \quad 0 \quad 0 \quad 2 \quad 0 \quad 0)^T$ and $\omega = 3$. The correlation between $x_j$ and $x_l$ is to be $0.5^{|l-j|}$ for $j = 1, 2, \ldots p$, and $l = 1, 2, \ldots p$. The number of observations for both training set and testing set is $N = 40$.

2. For Example 2, it is the same as the above one except that the correlation between any two different predictors is 0.85.

3. For Example 3, $\omega = 15$. The correlation between any two different predictors is 0.5. The number of observations for both training set and testing set is

$N = 100$. Let

$$\boldsymbol{\beta} = \Big(\underbrace{0\ldots0}_{10} \quad \underbrace{2\ldots2}_{10} \quad \underbrace{0\ldots0}_{10} \quad \underbrace{2\ldots2}_{10}\Big)^T, \tag{6.2}$$

4. For Example 4, $\omega = 15$. The number of observations for both training set and testing set is $N = 50$. Let

$$\boldsymbol{\beta} = \Big(\underbrace{3\ldots3}_{15} \quad \underbrace{0\ldots0}_{25}\Big)^T, \tag{6.3}$$

Let $\mathbf{X}$ be,

$$\mathbf{x}_j = \mathbf{Z}_1 + \boldsymbol{\varepsilon}_{\mathbf{x}_j}, \quad \mathbf{Z}_1 \sim N(0,1), \quad j = 1, \ldots, 5, \tag{6.4}$$

$$\mathbf{x}_j = \mathbf{Z}_2 + \boldsymbol{\varepsilon}_{\mathbf{x}_j}, \quad \mathbf{Z}_2 \sim N(0,1), \quad j = 6, \ldots, 10, \tag{6.5}$$

$$\mathbf{x}_j = \mathbf{Z}_3 + \boldsymbol{\varepsilon}_{\mathbf{x}_j}, \quad \mathbf{Z}_3 \sim N(0,1), \quad j = 11, \ldots, 15, \tag{6.6}$$

where $\mathbf{x}_j \sim N(0,1)$ is independent identically distributed such that $j = 16, \ldots, 40$, and $\boldsymbol{\varepsilon}_{\mathbf{x}_j}$, is independent identically distributed for $j = 1, \ldots, 15$. As can be seen that the first fifteen predictors are correlated as shown above, and the last 25 predictors are added noise.

5. For Example 5, it is the same as Example 4 except,

$$\mathbf{x}_j = \frac{\mathbf{Z}_1 + \mathbf{Z}_2}{2} + \boldsymbol{\varepsilon}_{\mathbf{x}_j}, \quad j = 1, \ldots, 5, \tag{6.7}$$

$$\mathbf{x}_j = \frac{\mathbf{Z}_2 + \mathbf{Z}_3}{2} + \boldsymbol{\varepsilon}_{\mathbf{x}_j}, \quad j = 6, \ldots, 10, \tag{6.8}$$

$$\mathbf{x}_j = \frac{\mathbf{Z}_1 + \mathbf{Z}_3}{2} + \boldsymbol{\varepsilon}_{\mathbf{x}_j}, \quad j = 11, \dots, 15, \tag{6.9}$$

**Method:** For each example, the simulation is replicated 50 times. Every replication will have two different sets: the training set, to fit the model; and the testing set to find the mean-squared prediction error. The median of the 50 mean-squared prediction error is the one that will be considered for performance comparison. The re-sampling method used for the simulation was tenfold cross validation.

**Results & Discussion:** The series of tables (Table (6.3) - Table (6.7)) show the results of the simulations. For Example 1 and Example 2, the $L_1$-exponential norm outperforms the other methods. Whether the correlation is low like the first example or high like the second example, $L_1$-exponential norm is better, and gives sparse solution.

The $L_1$-exponential norm outperforms all methods in Example 3. The $L_1$-exponential norm at $c = 0.4$ (with almost 20% weight of $L_1$-norm) showed to perform the best and its solution has the closest average of number of zeros compared to the actual model. The second best perform is the $L_1$-exponential norm at $c = 5$ (with almost 90% weight of $L_1$-norm).

For the last two examples; Example 4 and Example 5, the $L_1$-exponential norm outperformed all the other methods. These two examples are ideally made to check the ability of the methods to deal with the high correlations and noise. The

Table 6.3: Example 1 results

| Method | Example 1 | | | |
|---|---|---|---|---|
| | c | MSPE | SE | Ave 0's |
| Lasso | - | 9.84 | 2.06 | 5.3 |
| Ridge regression | - | 9.66 | 2.01 | - |
| Fixed-shape elastic net | 1.15 | 9.73 | 2.03 | 4 |
| $L_1$-exponential norm | 0.6 | 9.2 | 1.96 | 2.5 |
| | 2 | 9.33 | 2 | 4.4 |

Table 6.4: Example 2 results

| Method | Example 2 | | | |
|---|---|---|---|---|
| | c | MSPE | SE | Ave 0's |
| Lasso | - | 9.92 | 2.06 | 4.9 |
| Ridge regression | - | 9.74 | 1.99 | - |
| Fixed-shape elastic net | 0.01 | 9.76 | 2 | 0.5 |
| $L_1$-exponential norm | 5 | 8.97 | 1.83 | 4.6 |
| | 1 | 9.41 | 2 | 3.4 |

fixed shape elastic net came second in performance.

Table 6.5: Example 3 results

| Method | Example 3 | | | |
|---|---|---|---|---|
| | c | MSPE | SE | Ave 0's |
| Lasso | - | 243.73 | 33.85 | 30.5 |
| Ridge regression | - | 234.36 | 32.92 | - |
| Fixed-shape elastic net | 0.01 | 234.1 | 32.88 | 0.6 |
| $L_1$-exponential norm | 0.4 | 230 | 31.48 | 18.9 |
| | 5 | 231.15 | 31.68 | 29.8 |

Table 6.6: Example 4 results

| Method | Example 4 | | | |
|---|---|---|---|---|
| | c | MSPE | SE | Ave 0's |
| Lasso | - | 282.01 | 51.43 | 29.1 |
| Ridge regression | - | 299.07 | 53.85 | - |
| Fixed-shape elastic net | 2 | 268.5 | 49.1 | 20.4 |
| $L_1$-exponential norm | 0.6 | 264.9 | 48.8 | 17.3 |
| | 0.8 | 272.9 | 51.01 | 17.6 |

Table 6.7: Example 5 results

| Method | Example 5 | | | |
|---|---|---|---|---|
| | c | MSPE | SE | Ave 0's |
| Lasso | - | 263.154 | 50.7153 | 30.98 |
| Ridge regression | - | 258.7 | 50.3261 | - |
| Fixed-shape elastic net | 2 | 252.3 | 49.4514 | 22.38 |
| $L_1$-exponential norm | 0.4 | 248.9 | 47.5755 | 16.96 |
| | 1.5 | 266.4 | 52.8783 | 19.28 |

# CHAPTER 7

# DISCUSSION AND CONCLUSION

In this thesis, a brief literature review on the OLS method is presented. Its poor performance in terms of prediction accuracy and interpretation is highlighted. Some regularization and/or variable selection methods such as the ridge regression, the lasso, and the elastic net are illustrated. For the ridge regression, it can be concluded that it can overcome the problem of the prediction accuracy. However, its results are sometime very difficult to interpret. On the other hand, the lasso can overcome the interpretability issue with sparsity, but still it has its own drawbacks. For example, when the predictors are much higher than the number of observations, the sparsity is restricted. Also, when the predictors have a high correlation, only one of the correlated predictors is selected. The elastic net overcomes the above two drawbacks of the lasso, and have a higher flexibility than the previous mothods. This is due to the combination of the $L_1$-norm and

$L_2$-norm. Also, the elastic net has two parameters, which controls the size of the norm and the shape, by a weighted combination between the $L_1$-norm and $L_2$-norm.

The conventional elastic net was shown to fail at capturing all the combinations of norm shape and norm size. To overcome this, the concept of the fixed-shape elastic net was proposed. Basically, it controls the norm shape at different norm sizes. The same concept of the fixed-shape elastic net can be applied for other generalizations that use $\alpha$ as a combination weight.

Moreover, a new norm was introduced to the family of the regularized linear regressions. The norm is called as the exponential norm. The focus of this thesis is on the $L_1$-exponential norm. It can successfully compete with other existing methods. It was shown that the $L_1$-exponential norm is always convex, and can give a sparse solution. Moreover, it was shown that the elastic net is a two term approximation of the $L_1$-exponential norm.

The idea of the exponential norms started from Syed et al.[40]. The paper suggested to use the corrontropic function as a loss function instead of the first and second order loss function. Similarly, the intention here was to use exponential norms instead of the first and second order norms. There are two types of exponential norms, the first is the one that was introduced in the report:

$$\sum_{j=1}^{p}\left(e^{\dfrac{|\beta_j|^q}{q!\sigma^q}}-1\right)\leq e^{\dfrac{t^q}{q!\sigma^q}}-1. \tag{7.1}$$

This norm was shown to be flexible to have $L_q$-norm when $c \to \infty$, and $L_\infty$-norm when $c \to 0$. There is another type of exponential norm, (let it be called as the $L_q$-inverse exponential norm) that can be formulated as:

$$\sum_{j=1}^{p}\left(1-e^{\dfrac{-|\beta_j|^q}{q!\sigma^q}}\right)\leq 1-e^{\dfrac{-t^q}{q!\sigma^q}}. \tag{7.2}$$

This inverse exponential norm has a flexibility to have $L_q$-norm when $c \to \infty$, and $L_0$-norm when $c \to 0$, (see Figure (7.1)). This means that the norm will never be convex, since $L_0$-norm can exist $\forall q$.

The Taylor series expansion of the norm will be:

$$\sum_{j=1}^{p}\left(1-e^{\dfrac{-|\beta_j|^q}{q!\sigma^q}}\right)=\sum_{j=1}^{p}\sum_{m=1}^{\infty}\frac{(-1)^{m-1}|\beta_j|^{mq}}{(m!)(q!)^m\sigma^{mq}}=\sum_{j=1}^{p}\left(\frac{|\beta_j|^q}{(q!)\sigma^q}-\frac{|\beta_j|^{2q}}{2(q!)^2\sigma^{2q}}+\frac{|\beta_j|^{3q}}{3!(q!)^3\sigma^{3q}}-\dots\right). \tag{7.3}$$

Looking closely at a case where $q > 1$, say $q = 2$. Figure (7.2) shows us extreme nonconvex cases for $q = 2$, yet it cannot have a sparse solution due to lack of the contribution of $L_1$-norm.

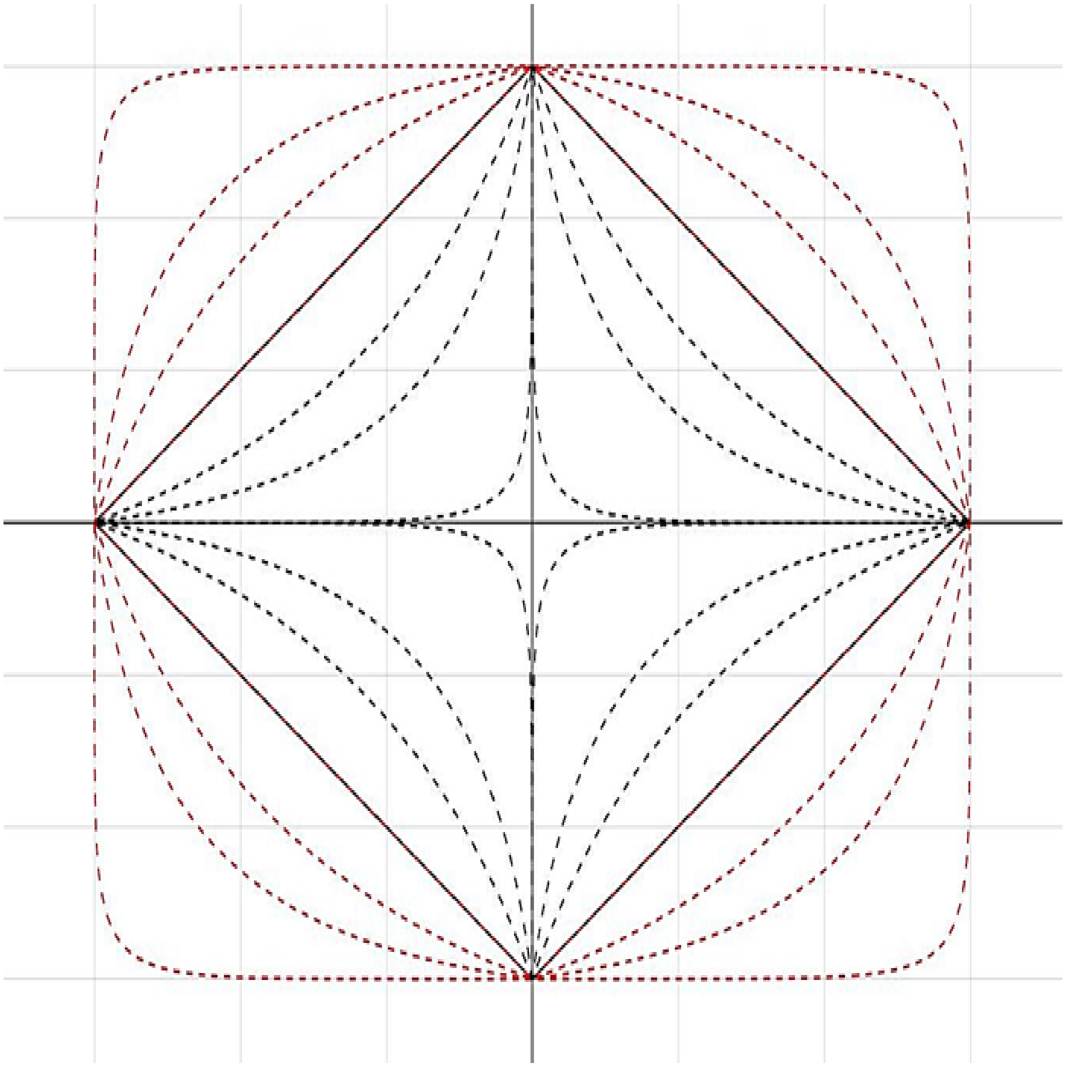

Figure 7.1: For two predictors, the exponential norms with $q = 1$ with $c =$ 0.1,0.5,1,1000, respectively from outside, and the inverse exponential norms with $c = 0.1$,0.5,1,1000, respectively from outside. They almost meet when $c = 1000$

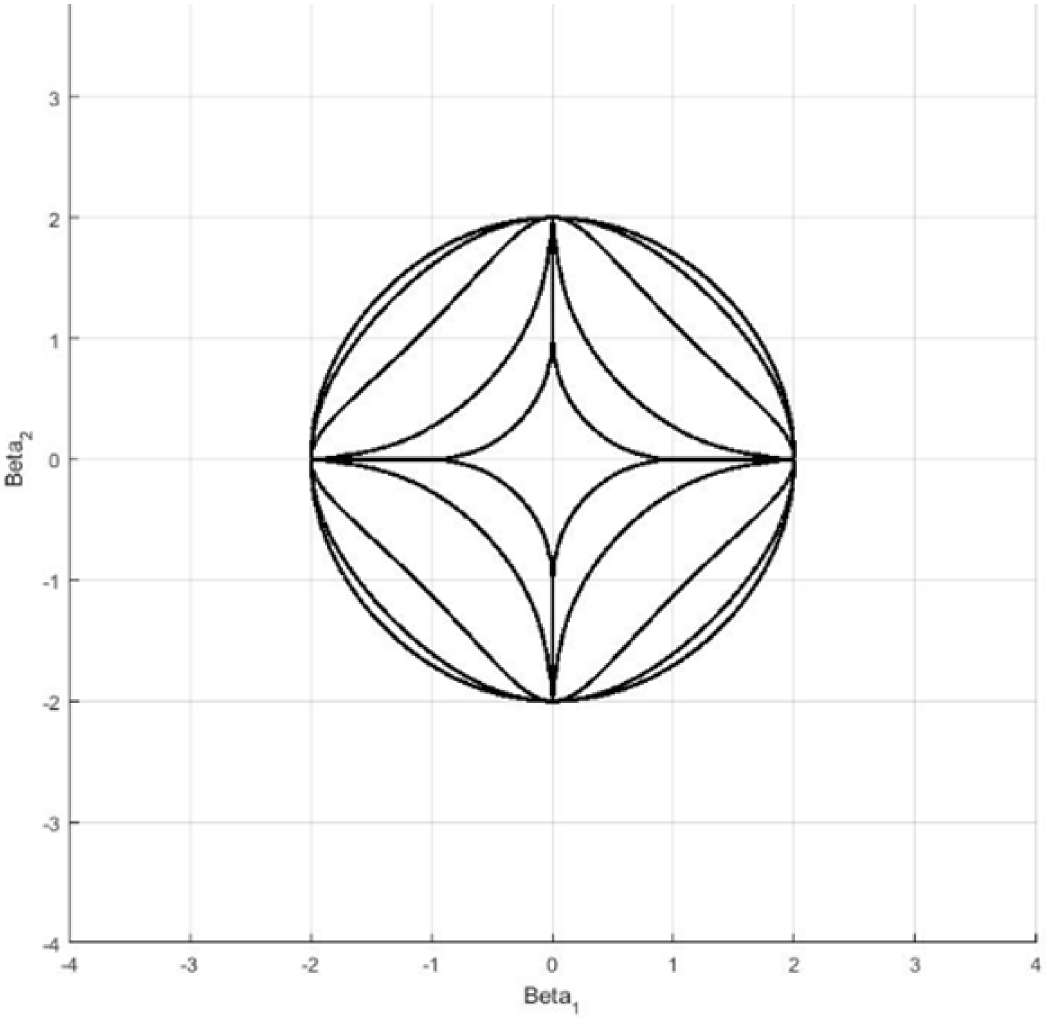

Figure 7.2: The sub-level sets of the $L_2$-inverse exponential norm

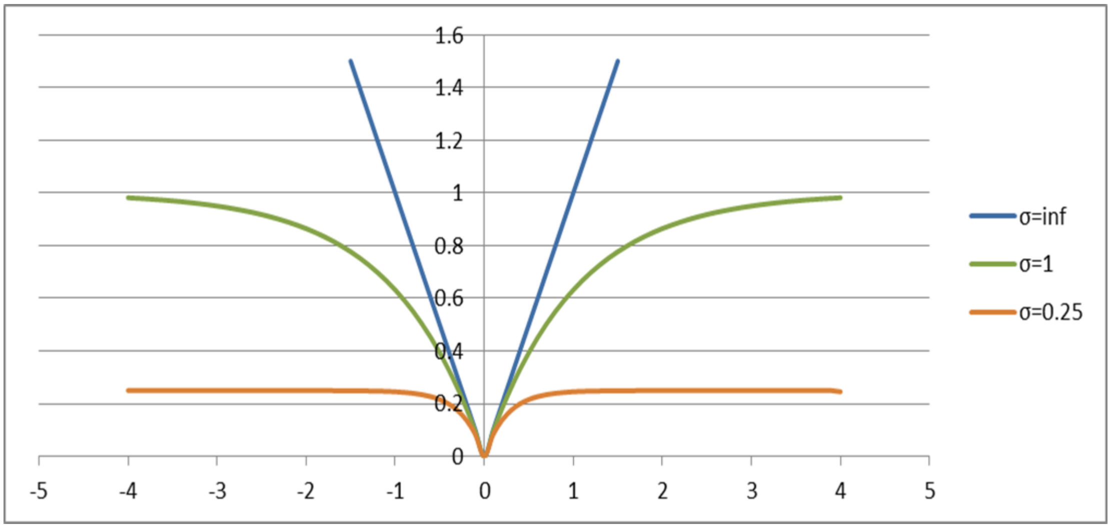

Figure 7.3: Function $1 - e\,\frac{|\beta|}{\sigma}$

Future study can be done on the nonconvex case which might give a very strong sparse property, whether it is the $L_q$-exponential norm where $q < 1$, or the $L_q$-inverse exponential norm. The $L_1$-inverse exponential norm can be considered as a novel method to the nonconvex PLR such as SCAD and MC+. The nonconvexity parameter in this case is $\sigma$. Figure (7.3) shows the penalty in $\mathbb{R}$ for different values of $\sigma$.

For solving the PLR, the coordinate descent algorithm was selected. three update mechanisms for the coordinate descent algorithm were presented to solve for PLR models. It was shown that the coordinate descent method is simple and efficient. Typically, the coordinate descent algorithm converges for $L_1$-norm penalty (such as the lasso and the elastic net) even though it is non-differentiable. The main element for convergence is convexity.

In addition to that, there is a contribution in the literature to solve for nonconvex PLR as described in Section 2.4. This can be applied for nonconvex exponential norms, such as the $L_1$-inverse exponential norm. Mazumder et al. [41] and Breheny and Huang [42] demonstrated coordinate descent algorithm to find a local optimal solution for nonconvex penalty.

Based on the numerical examples, it can be concluded that the proposed methods will excel when the predictors are highly correlated, and for $p \gg N$ case.

# REFERENCES

[1] E. B. (Firma), *Britannica book of the year.* Encyclopaedia britannica, 1973.

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

[3] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning.* Springer, 2013, vol. 112.

[4] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

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

[6] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[7] F. Bach, R. Jenatton, J. Mairal, G. Obozinski *et al.*, "Optimization with sparsity-inducing penalties," *Foundations and Trends® in Machine Learning*, vol. 4, no. 1, pp. 1–106, 2012.

[8] L. Breiman *et al.*, “Heuristics of instability and stabilization in model selection,” *The annals of statistics*, vol. 24, no. 6, pp. 2350–2383, 1996.

[9] L. Breiman, “Better subset regression using the nonnegative garrote,” *Technometrics*, vol. 37, no. 4, pp. 373–384, 1995.

[10] L. E. Frank and J. H. Friedman, “A statistical view of some chemometrics regression tools,” *Technometrics*, vol. 35, no. 2, pp. 109–135, 1993.

[11] J. Fan and R. Li, “Variable selection via nonconcave penalized likelihood and its oracle properties,” *Journal of the American statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[12] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[13] A. T. Puig, A. Wiesel, G. Fleury, and A. O. Hero, “Multidimensional shrinkage-thresholding operator and group lasso penalties,” *IEEE Signal Processing Letters*, vol. 18, no. 6, pp. 363–366, 2011.

[14] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, “A sparse-group lasso,” *Journal of Computational and Graphical Statistics*, vol. 22, no. 2, pp. 231–245, 2013.

[15] L. Jacob, G. Obozinski, and J.-P. Vert, “Group lasso with overlap and graph lasso,” in *Proceedings of the 26th annual international conference on machine learning.* ACM, 2009, pp. 433–440.

[16] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, “Sparsity and smoothness via the fused lasso,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.

[17] H. Zou, “The adaptive lasso and its oracle properties,” *Journal of the American statistical association*, vol. 101, no. 476, pp. 1418–1429, 2006.

[18] C.-H. Zhang *et al.*, “Nearly unbiased variable selection under minimax concave penalty,” *The Annals of statistics*, vol. 38, no. 2, pp. 894–942, 2010.

[19] L. Dicker, B. Huang, and X. Lin, “Variable selection and estimation with the seamless-l 0 penalty,” *Statistica Sinica*, pp. 929–962, 2013.

[20] N. Meinshausen, “Relaxed lasso,” *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 374–393, 2007.

[21] H. Wang, G. Li, and G. Jiang, “Robust regression shrinkage and consistent variable selection through the lad-lasso,” *Journal of Business & Economic Statistics*, vol. 25, no. 3, pp. 347–355, 2007.

[22] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard, “Wavelet shrinkage: asymptopia?” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 301–369, 1995.

[23] D. L. Donoho and I. M. Johnstone, “Adapting to unknown smoothness via wavelet shrinkage,” *Journal of the american statistical association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[24] W. K. Härdle and L. Simar, *Applied multivariate statistical analysis.* Springer Science & Business Media, 2012.

[25] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms.* John Wiley & Sons, 2013.

[26] A. Bagirov, N. Karmitsa, and M. M. Mäkelä, *Introduction to Nonsmooth Optimization: theory, practice and software.* Springer, 2014.

[27] M. R. Osborne, B. Presnell, and B. A. Turlach, "A new approach to variable selection in least squares problems," *IMA journal of numerical analysis*, vol. 20, no. 3, pp. 389–403, 2000.

[28] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani *et al.*, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[29] W. J. Fu, "Penalized regressions: the bridge versus the lasso," *Journal of computational and graphical statistics*, vol. 7, no. 3, pp. 397–416, 1998.

[30] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on pure and applied mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[31] A. J. Kooij *et al.*, *Prediction accuracy and stability of regression with optimal scaling transformations.* Child & Family Studies and Data Theory (AGP-D), Department of Education and Child Studies, Faculty of Social and Behavioural Sciences, Leiden University, 2007.

[32] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of statistical software*, vol. 33, no. 1, p. 1, 2010.

[33] P. Tseng *et al.*, "Coordinate ascent for maximizing nondifferentiable concave functions," 1988.

[34] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of optimization theory and applications*, vol. 109, no. 3, pp. 475–494, 2001.

[35] J. Friedman, T. Hastie, H. Höfling, R. Tibshirani *et al.*, "Pathwise coordinate optimization," *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.

[36] T. T. Wu and K. Lange, "Coordinate descent algorithms for lasso penalized regression," *The Annals of Applied Statistics*, pp. 224–244, 2008.

[37] T. Park and G. Casella, "The bayesian lasso," *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 681–686, 2008.

[38] D. Harrison and D. L. Rubinfeld, "Hedonic housing prices and the demand for clean air," *Journal of environmental economics and management*, vol. 5, no. 1, pp. 81–102, 1978.

[39] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri *et al.*, "Molec-

ular classification of cancer: class discovery and class prediction by gene expression monitoring," *science*, vol. 286, no. 5439, pp. 531–537, 1999.

[40] M. N. Syed, P. M. Pardalos, and J. C. Principe, "On the optimization properties of the correntropic loss function in data analysis," *Optimization Letters*, vol. 8, no. 3, pp. 823–839, 2014.

[41] R. Mazumder, J. H. Friedman, and T. Hastie, "Sparsenet: Coordinate descent with nonconvex penalties," *Journal of the American Statistical Association*, vol. 106, no. 495, pp. 1125–1138, 2011.

[42] P. Breheny and J. Huang, "Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection," *The annals of applied statistics*, vol. 5, no. 1, p. 232, 2011.

# Vitae

- Name: Abdulrahman Mohyiddin Mahmood Khan
- Nationality: Bahrain
- Date of Birth: Nov 11, 1989
- Email: *abd898@gmail.com*
- Permenant Address: Saudi Arabia, Khobar