# High Dimensional Variable Selection

Daniel Guetta
*Part III Essay*

(Dated: April 26, 2010)

'Model selection' is a statistical learning problem in which we use a set of input vectors $\boldsymbol{X}$ and a set of matching output numbers $Y$ to 'learn' something about the relationship between the inputs and outputs. The simplest, and most commonly used, model to represent this relationship is the *linear model*, in which $Y$ is estimated using a linear combination of the entries in $\boldsymbol{X}$.

A key characteristic of any model we might choose is its *dimension.* A high-dimensional model will use many of the variables in $\boldsymbol{X}$ to estimate $Y$. A low-dimensional model will use few of them. Surprisingly, we will see that low-dimensional models often perform better than high-dimensional ones.

This raises the question of *how* the choose the 'right' variables to include in our model, especially if the original vector $\boldsymbol{X}$ with which we are provided contains many observations (ie: is high dimensional). This essay is a review of the methods that have been developed to answer this question, with a particular focus on high-dimensional input data.

We first cover classical model selection theories, and explain why these are often not appropriate for high-dimensional models. We then consider methods that have been designed to deal with high-dimensional models. Finally, we explore some recent improvements over traditional methods.

### Contents

# Part I
# Preliminaries

## I. MODEL SELECTION

Consider a real valued input vector $\boldsymbol{X} \in \mathbb{R}^p$ (also called the vector of *predictors*) and a real valued output number $Y \in \mathbb{R}$ (also called the *response*). We restrict our attention to statistical models in which $Y$ is related to $\boldsymbol{X}$ as follows

$$Y = f(\boldsymbol{X}) + \epsilon$$

where $\mathbb{E}(\epsilon) = 0$. This assumption amounts to saying that all departures from the deterministic relationship $Y = f(\boldsymbol{X})$ can be captured in an additive error $\epsilon$.

In the quintessential statistical learning problem, we are given a 'training set' $\mathcal{T}$, consisting of $N$ pairs $(\boldsymbol{X_1}, Y_1), \cdots, (\boldsymbol{X_N}, Y_N)$, often grouped together into a matrix and a vector $\mathcal{T} = (\mathbf{X}, \boldsymbol{Y})$. Our aim is to use this training set to find the function $\hat{f}(\boldsymbol{X})$ that best estimates $f(\boldsymbol{X})$. We label our estimate $\hat{Y}$

$$\hat{Y} = \hat{f}(\boldsymbol{X})$$

One key property of any model we might choose is its *complexity*. More complex models use *many* of the components in $\boldsymbol{X}$ to estimate $Y$ – they are therefore high dimensional, and require many parameters. Less complex models ony use a few of the components of $\boldsymbol{X}$ to estimate $Y$ – they are low dimensional, and require fewer parameters. Surprisingly, we will find that if $\boldsymbol{X}$ contains many components, the second option usually gives rise to a better model.

Of course, finding the 'best' estimate implies we need a measure of how 'good' an estimate is. This measure is called the *loss function*, denoted $L(Y, \hat{Y})$. Given an input vector $\boldsymbol{X}$ and the *true* output $Y$, this function tells us how 'good' our prediction $\hat{Y} = \hat{f}(\boldsymbol{X})$ is.

The most common and convenient loss function is the *squared error loss*:

**Definition 1** (Squared Error Loss).

$$L\left(Y, \hat{Y}\right) = \left(Y - \hat{Y}\right)^2$$

In terms of this loss function, our aim is to find the $\hat{f}(\boldsymbol{X})$ that minimizes the *expected generalisation error*

**Definition 2** (Expected Generalisation Error).

$$\text{Err} = \mathbb{E}_{(x,y)}\left\{L\left(y, \hat{y}\right)\right\}$$

(In this paper, we use several different kinds of expectations, $\mathbb{E}$, each with respect to slightly different probability spaces. Their meanings should be self-evident, but we state them explicitely in Appendix A).

This poses a difficulty. We only have $N$ observations (those in $\mathcal{T}$) to 'test' our model, but we need to minimize the expected generalisation error over *all* possible observations.

Traditionally, the solution to this problem has been to consider only data in the training set. In other words, instead of minimizing the expected generalisation error, we minimise the *training error*

**Definition 3** (Training Error).

$$\begin{aligned}\overline{\text{err}} &= \mathbb{E}_{(X,Y)\in\mathcal{T}}\left\{L\left(y, \hat{y}\right)\right\} \\ &= \frac{1}{N}\sum_{(X,Y)\in\mathcal{T}} L\left(Y, \hat{Y}\right)\end{aligned}$$

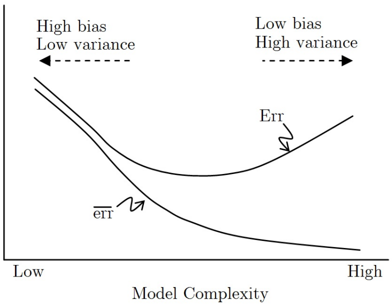

FIG. 1: The behaviour of the training error and expected generalisation error with varying model complexity. The training error does not take the variance of the model into account, and therefore consistently underestimates the expected generalisation error, especially for very complex models. Adapted from [Hastie Tibshirani and Friedman 2009].

### A. The Problem

For low-dimensional problems (in which $\boldsymbol{X}$ is a small vector), the training error is a fair estimate of the expected generalisation error. However, as we consider increasingly large models, the situation quickly deteriorates. Figure 1 illustrates two problems that develop.

- The training error consistently underestimates the expected generalisation error.
- More worringly, the behaviours of the two quantities differ for more complex models; the training error decreases indefinitely as complexity increases, whereas the expected generalisation error reaches a minimum.

  This means that whereas the 'best' model lies at some intermediate level of complexity, a naive minimisation of the training error would lead us to choose the most complex model possible instead.

We quantise both these effects by defining the *expected optimism*, which asses the extent to which the training error underestimates the expected generalisation error[1]

**Definition 4** (Expected Optimism).

$$\omega = \mathbb{E}_{\mathcal{T}}\left\{\text{Err} - \overline{\text{err}}\right\}$$

The expectation is taken over all possible training sets $\mathcal{T}$.

Before moving on, we consider the two issues above in more detail.

The first problem is intuitively understandable. Because the training set $\mathcal{T}$ is used to fit the model $\hat{f}(\boldsymbol{X})$, the model is 'tailored' to data in the training set. The model therefore performs much better with points in $\mathcal{T}$ (low $\overline{\text{err}}$) than with points outside it (high Err). More technically, the training error uses the same data to fit the model and to asses the goodness of the fit. The expected generalisation error uses new data to asses the fit.

To understand the second problem, we need to explore the nature of the expected generalisation error in more detail.

**Theorem 1.** *The expected generalisation error under squared error loss can be writen as*

$$\begin{aligned}\mathbb{E}_{(x,y)}\left\{L(y,\hat{f}(\boldsymbol{x}))\right\} &= \mathbb{V}\text{ar}(\epsilon) + \mathbb{V}\text{ar}\left(\hat{f}(\boldsymbol{x})\right) \\ &\quad + \left[\mathbb{E}\left(\hat{f}(\boldsymbol{x})\right) - f(\boldsymbol{x})\right]^2 \quad (1)\end{aligned}$$

*Proof.* See Appendix C □

Consider the three terms in equation 1

- $\mathbb{V}\text{ar}(\epsilon)$ is called the *irreducible error* in $Y$ – it comes from the underlying, true model $f$. Even if our statistical model $\hat{f}$ *exactly* represented the real underlying model, it would still contain this error.
- $\mathbb{V}\text{ar}\left(\hat{f}(\boldsymbol{X})\right)$ is the variance of our model $\hat{f}$. This is a *random* source of error in our model.
- $\left[\mathbb{E}\left(\hat{f}(\boldsymbol{X})\right) - f(\boldsymbol{X})\right]^2$ is the *bias* of our model $\hat{f}$. This is a *systematic* source of error in our model.

How do the bias and variance change as we vary the complexity of our model?

- Fitting a model with many components leads to a model with low bias. This is because the model will be more complex and will therefore better fit the data.

  However, this also means that we have more parameters to estimate with the same amount of data, and this means that the *variance* of each parameter will be very large.
- Fitting a simple model with few components leads to a model with low variance, because many data points will be available per parameter to estimate. However, the model will be less complex, and this will lead to greater bias.

This behaviour of the expected generalisation error is called the *bias-variance tradeoff*. The training error reflects none of this subtlety. It only considers the bias and not the variance. The more complex a model we choose,

[1] This definition is slightly at odds with that in the literature (see, for example, [Hastie Tibshirani and Friedman 2009] and [Efron 1986]), which calls for the definition of the *in-sample error* instead of the expected generalisation error. The distinction is mostly technical, and we relegate it to Appendix B, with the proof of theorem 2.

the better it will fit the points in the training set, and the lower the training error. This explains the differing behaviours of $\overline{\text{err}}$ and Err in Figure 1.

We will formalise this qualitative discussion in the context of the linear model in the next section, but we can gain some insight using a theorem which we can prove in full generality.

**Theorem 2.** *For a wide class of loss functions including squared error loss (definition 1)*

$$\omega = \frac{2}{N}\sum_{i=1}^{N} \mathbb{C}\mathrm{ov}\left(\hat{Y}_i, Y_i\right)$$

*where $N$ is the number of items in the training set $\mathcal{T}$, $Y_i$ is the ith output in the training set and $\hat{Y}_i = \hat{f}(\boldsymbol{X}_i)$, our model's prediction of what $Y_i$ should be.*

*Proof.* See Appendix B. □

The theorem effectively says that the optimism of the training error is greater when our model does a *good* job of predicting $y$ *on average*. This is perfectly consistent with our observation that the less biased the model, the greater the variance and the more inaccurate the training error.

The discussion above reveals the startling fact that even though minimising the training error will lead to the model with the least bias, it might sometimes be advantageous to intentionally fit a biased model, so as to achieve a drop in variance. Furthermore, it seems that any judicious way to bias our model will also reduce its complexity, since variance seems to be related to model complexity.

There is one final point to take into account, and that is that in some very high dimensional models, we have *prior reason* to believe that some of the variables should not be included in the model. For example, many biological microarray experiments, include thousands of genes as potential predictors, but only a few of these are expected to influence results. In these situations, we have an additional motivation to reduce the complexity of the model.

The question remains, however – *how* should we reduce the complexity of our model? Which variables should be drop, and how many? The methods in this paper provide an answer to these questions.

### B. The way forward

The way forward is clear – we must somehow find an expression that accurately estimates Err for any model. This will then allow us to choose the model that minimises that expression.

The rest of this essay explores such methods. We will see that there are broadly two approaches we can take

- Methods like *cross validation* and *bootstrapping* attempt to estimate Err directly, from the training set $\mathcal{T}$. We will briefly consder these methods in section III.

- Methods like Mallow's $C_p$, the Akaike Information Criterion, the Bayesian Information Criterion, Penalised Least Squares, etc... estimate Err by estimating the expected optimism $\omega$ and then adding it to $\overline{\text{err}}$ (which can easily be calculated using data in $\mathcal{T}$). We will consider these methods at length in the rest of this paper.

Our discussion has thus far been very general. The time has come to focus our discussion on a particular type of model $\hat{f}$. Of the many models used in statistical learning, the most fundamental and most often used is the linear model. This model will be our focus in this paper.

## II. THE LINEAR MODEL

In the *linear model*, we restrict our search to functions of the form

$$\hat{f}(\boldsymbol{X}) = \boldsymbol{X\beta} \tag{2}$$

In its most general form, the linear model takes the form $\hat{f}(\boldsymbol{X}) = \beta_0 + \boldsymbol{X\beta}$ (where $\beta_0$ may be 0). The data can, however, be normalised to eliminate the need for a constant term $\beta_0$, and this considerably simplifies notation. We will also find it useful to normalise the vectors $\boldsymbol{X}$ to have a mean of 0 and a variance of 1. In the rest of this paper, we will assume these normalisations have been carried out, and we summarise them in the following definition

**Definition 5** (Normalisation conditions)**.** We will always assume that the inputs $\mathbf{X}$ and outputs $\boldsymbol{Y}$ are normalised as follows

$$\sum_{i=1}^{N} Y_i = 0 \qquad \sum_{i=1}^{N} X_{ij} = 0, \forall j \qquad \sum_{i=1}^{N} X_{ij}^2 = 1, \forall j$$

In the context of the linear model

- A simple model is one that does not use every component of $\boldsymbol{X}$ to estimate $Y$. Thus, in a simple model a number of components of $\boldsymbol{\beta}$ will be equal to 0.

- In a complex model, more components of $\boldsymbol{\beta}$ will be nonzero, reflecting the fact more variables are used.

Our aim is to find the $\boldsymbol{\beta}$ that minimizes the expected generalisation error (definition 2).

We first take the naive approach of minimising the training error (definition 3). We call this method *ordinary least squares* (OLS)

$$\begin{aligned}\overline{\mathrm{err}}(\boldsymbol{\beta}) &= \frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \hat{f}(\boldsymbol{X}_i)\right)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}(Y_i - \boldsymbol{X}_i\boldsymbol{\beta})^2 \\ &= \frac{1}{N}\|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 \qquad (3)\end{aligned}$$

We differentiate and set to 0. Solving, we find that

$$\hat{\boldsymbol{\beta}}^{\mathrm{OLS}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\boldsymbol{Y} \qquad (4)$$

Given an input vector $\boldsymbol{x}$, our corresponding estimate for $y$ is then

$$\hat{y}^{\mathrm{OLS}} = \hat{f}(\boldsymbol{x}) = \boldsymbol{x}^T\hat{\boldsymbol{\beta}}^{\mathrm{OLS}} = \boldsymbol{x}^T\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\boldsymbol{Y} \qquad (5)$$

We define a so-called 'hat matrix'

**Definition 6** (Hat Matrix).

$$\mathbf{H}^{\mathrm{OLS}} = \boldsymbol{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$$

This allows us to relate the outputs in the training set $\boldsymbol{Y}$ and the outputs of our model $\hat{f}(\boldsymbol{X})$ as follows

$$\hat{\boldsymbol{Y}}^{\mathrm{OLS}} = \mathbf{H}^{\mathrm{OLS}}\boldsymbol{Y}$$

### A. Distribution of $\hat{\beta}$

If the errors $\epsilon$ are Gaussian, then

$$\boldsymbol{Y} \sim N_n\left(X\boldsymbol{\beta}, \sigma_\epsilon^2\mathbf{I}\right)$$

where $N_d$ is the d-dimensional multivariate normal.

We also know that $\hat{\boldsymbol{\beta}}^{\mathrm{OLS}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\boldsymbol{Y}$ – in other words, $\hat{\boldsymbol{\beta}}^{\mathrm{OLS}}$ is simply a linear transformation of $\boldsymbol{Y}$. Therefore[2]

$$\begin{aligned}\hat{\boldsymbol{\beta}}^{\mathrm{OLS}} &\sim N_p\Big(\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T X\boldsymbol{\beta}, \\ &\qquad \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{I}\mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\sigma_\epsilon^2\Big) \\ &\sim N_p\left(\boldsymbol{\beta}, \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\sigma_\epsilon^2\right) \qquad (6)\end{aligned}$$

Notice that

- Assuming the normalisation conditions in defintion 5 are met, the covariance matrix of the input data, $\mathbf{X}$ is given by

  $$\mathbf{S} = \frac{\mathbf{X}^T\mathbf{X}}{N}$$

  The covariance matrix of $\hat{\boldsymbol{\beta}}^{\mathrm{OLS}}$ is therefore the inverse of $\mathbf{S}$ (up to a factor of $N$).

  This is very sensible. What does a high $\mathbf{S}$ in a certain direction mean? It means that the data points in $\mathcal{T}$ are very spread out along that particular direction. The nature of the model along that direction is therefore very well defined, and the variance of our estimate of the gradient of that line will therefore be very low. (See figure 2 for an illustration of this phenomenom. The caption of the picture refers to material in the next section).

- $\mathbb{E}(\hat{\boldsymbol{\beta}}) = \beta$, and therefore $\mathbb{E}(\hat{Y}) = Y$. In other words, the linear model is *unbiased.*

  In fact, by the Gauss Markov Theorem, the ordinary least squares estimate has the smallest variance amongst all least-squares estimates that are unbiased. However, as we saw in the last section, the variance can be reduced much further by introducing some bias.

### B. The geometric interpretation of ordinary least squares

Consider once again the least squares estimate of $Y$

$$\hat{\boldsymbol{Y}}^{\mathrm{OLS}} = \mathbf{X}\boldsymbol{\beta}^{\mathrm{OLS}}$$

Clearly the right-hand-side of this equation is a linear combination of the columns of $\mathbf{X}$. In other words, our estimate $\hat{\boldsymbol{Y}}$ lies in the *vector space* spanned by the columns of $\mathbf{X}$. Ordinary least squares, then, simply consists of finding the closest vector to $\boldsymbol{Y}$ that lies in that vector space – in other words, the *projection* of $\boldsymbol{Y}$ onto that space.

We can gain greater mathematical insight into this statement using *singular-value decomposition* of the matrix $\mathbf{X}$ – this can be thought of as a generalisation of eigenvalue decomposition for non-square matrices.

Our matrix $\mathbf{X}$ is an $N \times p$ matrix, which can be thought of as converting a vector from '$\boldsymbol{\beta}$-space' (dimension $p$) to '$\hat{\boldsymbol{Y}}$-space' (dimension $N$) – effectively, it takes a $\boldsymbol{\beta}$ and transforms it into the correct $\hat{\boldsymbol{Y}}$. The singular value decomposition of $\mathbf{X}$ is[3]

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

[2] It is a well known property of the multivariate normal that if $\boldsymbol{Y} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\boldsymbol{X} = \mathbf{A}\boldsymbol{Y}$, then $\boldsymbol{X} \sim N(\mathbf{A}\boldsymbol{\mu}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^T)$

[3] The definition of singular value decomposition that we use here is that in [Hastie Tibshirani and Friedman 2009, pp. 64-66]. It is slightly at odds with other definitions in the literature

where

- $\mathbf{V}$ is a $p \times p$ matrix, whose columns are orthonormal and span the *row* space of $\mathbf{X}$.

  It can be thought of as converting whatever vector $\mathbf{X}$ is acting on from '$\boldsymbol{\beta}$-space' into the 'eigenspace' of $\mathbf{X}$.

  Note that $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$.

- $\mathbf{D}$ is a $p \times p$ diagonal matrix containing entries $d_1, \cdots, d_p$, called the *singular values* of the matrix $\mathbf{X}$. These are analogous to the 'eigenvalues' of $\mathbf{X}$.

- $\mathbf{U}$ is an $N \times p$ matrix, whose columns are orthonormal and span the *column* space of $\mathbf{X}$.

  It can be thought of as converting our vector back from the 'eigenspace' of $\mathbf{X}$ into '$\boldsymbol{Y}$-space'.

  Note that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, but because the matrix is not square, $\mathbf{U}\mathbf{U}^T \neq \mathbf{I}$.

Now, recall that

$$\hat{\boldsymbol{Y}}^{\text{OLS}} = \mathbf{H}^{\text{OLS}}\boldsymbol{Y}$$

According to our discussion above, the action of $\mathbf{H}$ should be to project $\boldsymbol{Y}$ onto the space spanned by the columns of $\mathbf{X}$. To see how this is the case, consider the singular-value form of $\mathbf{H}$

$$\begin{aligned}\mathbf{H}^{\text{OLS}} &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \\ &= \mathbf{U}\mathbf{D}\mathbf{V}^T\left(\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{U}\mathbf{D}\mathbf{V}^T\right)^{-1}\mathbf{V}\mathbf{D}\mathbf{U}^T\boldsymbol{Y} \\ &= \mathbf{U}\mathbf{D}\mathbf{V}^T\left(\mathbf{V}\mathbf{D}\mathbf{D}\mathbf{V}^T\right)^{-1}\mathbf{V}\mathbf{D}\mathbf{U}^T\boldsymbol{Y} \\ &= \mathbf{U}\mathbf{U}^T\boldsymbol{Y} \qquad (7)\end{aligned}$$

In this form, we immediately see that $\mathbf{H}^{\text{OLS}}$ is indeed a projection matrix. The matrix $\mathbf{U}^T$ first projects $\boldsymbol{Y}$ onto the 'eigenspace' of $\mathbf{X}$ (notice that $\mathbf{U}^T\boldsymbol{Y}$ contains the coordinates of $\boldsymbol{Y}$ expressed in the basis $\mathbf{U}$), and then retransforms the resulting vector back into '$\hat{\boldsymbol{Y}}$-space' (using $\mathbf{U}$). The result is an approximation of $\boldsymbol{Y}$ using only the columns of $\mathbf{U}$.

Before we conclude this section, we spend a short time discussing the meaning of the singular values (the entries in the matrix $\mathbf{D}$) – this will come in useful when we discussion Ridge regression in section VII E. We first write the covariance matrix of $\mathbf{X}$ in singular-value form

$$\begin{aligned}\mathbf{S} &= \frac{\mathbf{X}^T\mathbf{X}}{N} \\ &= \frac{\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{U}\mathbf{D}\mathbf{V}^T}{N} \\ &= \frac{\mathbf{V}\mathbf{D}^2\mathbf{V}^T}{N}\end{aligned}$$

This is none other than the eigen-decomposition of $\mathbf{X}^T\mathbf{X}$. Thus, the singular values $d_j$ are the (square root of) the eigenvalues of $\mathbf{S}$, and the vectors $\boldsymbol{v}_j$ that form the columns of the matrix $\mathbf{V}$ are the eigenvectors of $\mathbf{S}$.

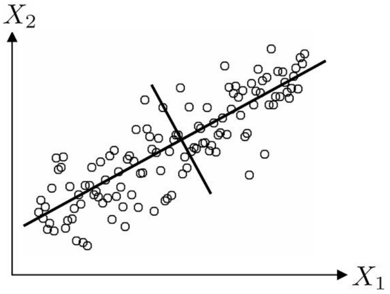

FIG. 2: Principal components of a matrix $\mathbf{X}$ containing two columns – one for $X_1$ and one for $X_2$. The $Y$ axis is out of the page, and the dots are projections of the data points on the $X_1$-$X_2$ plane. The heavy lines indicate the principal components of the matrix – the first principal component has the largest singular value, and the second is orthogonal to the first. Ordinary least squares projects the vector $\boldsymbol{Y}$ onto these components.

Let $d_1$ be the largest singular value. This means that $\boldsymbol{v}_1$ is the linear combination of columns of $\mathbf{X}$ that has a larger sample variance than any other. $\boldsymbol{v}_2$ is then the next largest, subject to being orthogonal to $\boldsymbol{v}_1$ (remember that $\mathbf{V}$ is an orthonormal matrix), and so on. These directions are called the *principal components* of the matrix $\mathbf{X}$. Ordinary least squares projects $\boldsymbol{Y}$ onto these components. The concept is illustrated in figure 2.

### C. Expected Optimism in Action

We are now able to re-examine the results of section I in the more concrete context of the linear model.

We first prove two theorems

**Theorem 3.** *For a prediction method satisfying* $\hat{\boldsymbol{Y}} = \mathbf{H}\boldsymbol{Y}$*, the expected optimism is given by*

$$\omega = \frac{2}{N}\mathbb{T}\text{r}(\mathbf{H})\sigma_\epsilon^2$$

*Proof.* See Appendix B □

This theorem is in perfect agreement with our discussion in section I. Indeed, the trace of $\mathbf{H}$ is directly related to the complexity of the model. Recall that $\mathbf{H}$ is effectively a projection of $\mathbf{Y}$ onto a smaller space (section II B). Expressing $\boldsymbol{Y}$ in diagonal form results in a matrix whose trace is equal to the number of dimensions onto which $\mathbf{H}$ projects.

This leads us to define the *effective number of parameters*

**Definition 7** (Effective number of parameters)**.**

$$d_{\text{eff}} = \mathbb{T}\text{r}\left(\mathbf{H}\right)$$

**Theorem 4.** *If a linear model is fit using ordinary least squares (with $\hat{f}(\boldsymbol{X}) = \boldsymbol{X}\hat{\boldsymbol{\beta}}^{OLS}$)*

$$\omega = \frac{2p}{N}\sigma_\epsilon^2$$

*where $p$ is the number of covariates in $\boldsymbol{X}$ and $\sigma_\epsilon^2 = \mathbb{V}\text{ar}(Y_i)$, the irreducible error in the underlying model.*

*Proof.* See Appendix B □

### D. Orthonormal design

In what follows, it will often be enlightening to consider cases in which $\mathbf{X}^T\mathbf{X} = \mathbf{I}$ – this is called the *orthonormal design* case, and it results in a particularly simple expression for equation 3.

**Theorem 5.** *In the orthonormal design case, in which $\mathbf{X}^T\mathbf{X} = \mathbf{I}$, equation 3 can be written as*

$$Q(\boldsymbol{\beta}) = \frac{1}{N}\left\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}^{OLS}\right\|^2 + \frac{1}{N}\left\|\hat{\boldsymbol{\beta}}^{OLS} - \boldsymbol{\beta}\right\|^2 \qquad (8)$$

*where $\hat{\boldsymbol{\beta}}^{OLS}$ is the ordinary least squares estimate of $\boldsymbol{\beta}$ and $\hat{\boldsymbol{Y}}^{OLS}$ is the ordinary least squares estimate of $\boldsymbol{Y}$.*

*Proof.* See Appendix D. □

This Theorem seems almost trivial – and indeed, in the case of ordinary least squares, it is. The reason it is so useful is because the first term in equation 8 does not depend on $\boldsymbol{\beta}$, the argument of the function $Q$. It only depends on $\hat{\boldsymbol{\beta}}^{\text{OLS}}$, which in turns only depends on our input data. This means that in minimising $Q(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$, we only need to consider the second term, which can be written as $\sum_{j=1}^{p}(\hat{\beta}_j^{\text{OLS}} - \beta_j)^2$. In this form, it is clear that it is sufficient to minimise the term for each component of $\boldsymbol{\beta}$ separately.

In other words, minimising $Q(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$ is equivalent to minimising

$$q(\beta_j) = \left(\hat{\beta}_j^{\text{OLS}} - \beta_j\right)^2$$

for each $\beta_j$.

In this particular case, it is obvious that the solution is $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}^{\text{OLS}}$. In some more complex models we will be consdering, this will not be the case.

## III. ESTIMATING Err DIRECTLY

We now return to the estimation of the expected generalisation error, and begin by briefly considering methods that directly estimate Err from the data in the training set $\mathcal{T}$

### A. Cross-validation & Generalised Cross-validation

We saw above that the reason $\overline{\text{err}}$ underestimates Err is because the same data is used to fit the model and to assess its ‘goodness’.

Cross-validation [Stone 1974], [Allen 1971, where it appears under the name of PRESS] solves that problem by splitting the data into $K$ separate segments, fitting the model using $K - 1$ of these segments and assessing the goodness of the fit using the last segment. More formally, we define the *cross-validation score* as an estimate to the expected generalisation error

**Definition 8** (Cross-Validation Score)**.** We divide our training set $\mathcal{T}$ into $K$ equal segments or folds. We write $\hat{f}^{-\kappa(i)}$ to represent the model fitted to $\mathcal{T}$ *minus* the segment containing data point $i$. The cross-validation score is then

$$\text{CV}\left(\hat{f}\right) = \frac{1}{N}\sum_{i=1}^{N} L\left(Y_i, \hat{f}^{-\kappa(i)}(X_i)\right)$$

It remains to decide how to choose $K$, the number of folds to break our data into. Once again, we come against the bias-variance tradeoff. A large value of $K$ will result in a low bias (because each set will contain many data points) but a large variance (because the sets will be very similar). At this extreme, $K = N$ fits the model $N$ times, each time ommitting a single data point – this is called ‘leave-one out’ cross validation. By contrast, a small value of $K$ will result in a high bias but a low variance.

On balance, a value of $K = 5 - 10$ is often recommended [Kohavi 1995] ([Breiman and Spector 1992] also show that in some cases, 5-fold cross-validation performs better than ‘leave-one out’ cross validation).

To fit our model using cross validation, we would select the model with the lowest cross validation score (we will have more to say about the mechanics of the process in section V). The computational burden of cross-validation is considerable, especially when $K$ is large and the model needs to be fitted many times. In certain special problems, however, this computation can be done quickly. For example, when fitting a linear model $\hat{\boldsymbol{Y}} = \mathbf{H}\boldsymbol{Y}$, it can be shown [Hastie and Tibshirani 1990, pp 47] that the cross-validation score takes the form

$$\mathrm{CV}\left(\hat{f}\right) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{Y_i - \hat{f}(X_i)}{1 - H_{ii}}\right)^2$$

In some cases, it is easier to calculate the trace of $\mathbf{H}$ than it's $i$th diagonal element, and this leads to an estimate of the cross-validation score called the *generalised cross-validation score*

$$\mathrm{GCV}\left(\hat{f}\right) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - \hat{f}(X_i)}{1 - \mathbb{T}\mathrm{r}(\mathbf{H})/N}\right)^2 \tag{9}$$

### B. Bootstrapping

Bootstrapping was developed by [Efron 1979]. The basic idea is to randomly draw datasets *with* replacement from the training set, with each sample being the same size as the original training set (possibly containing repetitions). This is done $B$ times, producing $B$ bootstrap datasets.

How can we use these bootstrap datasets to estimate prediction error?[4] One approach would be to fit our model to each bootstrap dataset and keep track of how well it predicts the original training set. Let $\hat{f}^{*b}$ be the model obtained by fiting the data in the $b$th bootstrap dataset. Then our estimation of Err would be

$$\frac{1}{B}\frac{1}{N}\sum_{b=1}^{B}\sum_{i=1}^{N} L\left(Y_i, \hat{f}^{*b}(X_i)\right)$$

The problem is that there is overlap between the bootstrap datasets and the training samples on which they are tested – and this is precisely the reason $\overline{\mathrm{err}}$ was a poor estimate of Err in the first place.

By mimicking cross-validation, a better bootstrap estimate can be obtained by only testing our models on data *not* contained in the relevant dataset.

**Definition 9** (Bootstrap estimate of Err)**.** Collect $B$ samples, with replacement, from the training set $\mathcal{T}$. Let $\hat{f}^{*b}$ be the model fitted to the $b$th bootstrap dataset. Let $C^{-i}$ be the set of indices of the bootstrap samples $b$ that *do not* contain observation $i$, and $|C_i|$ be the size of that set.

Then the bootstrap estimate of Err is defined by

$$\hat{\mathrm{Err}}^{b} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|C^{-i}|}\sum_{b\in C^{-i}} L(Y_i, \hat{f}^{*b}(X_i))$$

This solves the overfitting problem created by cross-validation. However, a problem still remains, and to appreciate it more fully, we first prove the following theorem.

**Theorem 6.** *The probability of a given observation appearing in any given bootstrap sample is roughly* $0.632$*. As such, the number of distinct observations in each bootstrap sample is roughly* $0.632N$*, where* $N$ *is the number of items in our training set.*

*Proof.* The probability of choosing any item in the training set is $1/N$. Thus, the probability of *not* choosing an item is $(1 - \frac{1}{N})$. Since each bootstrap sample consists of $N$ items, we would need to *not* choose an item $N$ times for it not to apppear in the sample. Thus:

$$\mathbb{P}\left(\text{observation } i \notin \text{sample } b\right) = \left(1 - \frac{1}{N}\right)^N$$

And therefore

$$\mathbb{P}\left(\text{observation } i \in \text{sample } b\right) = 1 - \left(1 - \frac{1}{N}\right)^N$$

For large $N$

$$\mathbb{P}\left(\text{observation } i \in \text{sample } b\right) \approx 1 - e^{-1} = 0.632$$

It follows that the average number of distinct observations in each sample is about $0.632N$. □

Theorem 6 implies that our bootstrap estimate of Err behaves roughly like twofold cross validation ($K = 2$), because each boostrap sample contains roughly half of the data points. We saw, however, that $K = 5 - 10$ is optimal. The bootstrap samples are therefore smaller than might be ideal, the resulting models will therefore be less complex and the predicted error will be *biased upwards.*

To remedy to this, [Efron 1983] proposed the .632 estimator of Err

$$\hat{\mathrm{Err}}^{(.632)} = 0.368 \cdot \overline{\mathrm{err}} + 0.632 \cdot \hat{\mathrm{Err}}^{b} \tag{10}$$

Effectively, this model takes a combination of points that are too close to the training set (used to work out $\overline{\mathrm{err}}$) and points that are too far from the training set (used to work out $\hat{\mathrm{Err}}^{b}$) to get a balanced average. The derivation of the coefficients is somewhat heuristic, and we leave the interested reader to consult the paper for more details.

It is worth noting that some improvements are available over the .632 bootstrap estimator. In particular, the .632+ bootstrap estimator [Efron and Tibshirani 1997] adapts the coefficients in equation 10 to the data in the training set.

[4] It is interesting to node that [Efron 1983] considers the bootstrap as an estimation of the expected optimism ($\omega$) rather than the expected generalisation error Err. The distinction is purely semantic, and we prefer to take the view of [Hastie Tibshirani and Friedman 2009] – namely that the bootstrap estimates Err directly.

## IV. ESTIMATING Err USING THE EXPECTED OPTIMISM

We now consider methods that estimate Err by first estimating the expected optimism $\omega$.

### A. Mallow's $C_p$ and the AIC

We saw, in theorem 4, that for ordinary least squares, the expected optimism is given by

$$\omega = \frac{2p}{N}\sigma_\epsilon^2$$

The $C_p$ statistic [Mallows 1973] simply adds this $\omega$ to the training error to obtain an estimate for the expected generalisation error.

**Definition 10** (Mallows' $C_p$)**.**

$$C_p = \frac{1}{N}\left\|\boldsymbol{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right\|^2 + 2\frac{p}{N}\sigma_\epsilon^2$$

where $N$ is the number of observations in the training set and $p$ is the number of variables in our model.

The variance $\sigma_\epsilon^2$ is estimated by using a low-bias method (ordinary least squares, for example) to fit the data. We then calculate $C_p$ for every candidate model, and choose the one with the lowest value (see section V for details).

The Akaike Information Criterion (AIC) [Akaike 1974] works in a very similar way, but uses a somewhat different loss function, which makes it more applicable to a more general class of models. In the case of the linear model with Gaussian errors $\epsilon$, AIC and $C_p$ are equivalent, and we refer interested readers to Akaike's paper.

Before concluding this section, we take a small detour to prove a very satisfying result – that minimising the AIC and minimising the generalised cross-validation score (section III A) are two asymptotically equivalent methods. To do this, consider the generalised cross validation score in equation 9

$$\text{GCV}\left(\hat{f}\right) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - \hat{f}(x_i)}{1 - \mathbb{T}\mathrm{r}(\mathbf{H})/N}\right)^2$$

Using the approximation $1/(1-x)^2 \approx 1 + 2x$, we have

$$\text{GCV}\left(\hat{f}\right) \approx \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{f}(x_i)\right)^2\left(1 + 2\frac{\mathbb{T}\mathrm{r}(\mathbf{H})}{N}\right)$$

In the linear model, we have seen that $\mathbb{T}\mathrm{r}(\mathbf{H}) = p$, and so

$$\text{GCV}\left(\hat{f}\right) = \frac{1}{N}\|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \frac{2d}{N}\left\{\frac{\|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2}{N}.\right\} \quad (11)$$

The term in curly brackets is simply an estimate of the variance $\sigma_\epsilon^2$, and so expression 11 is none other than the AIC! This confirms that these are ultimately different ways to do the same thing – namely, estimating the expected generalisation error using only the data in the training set. This result also extends to cross-validation itself (rather than *generalised* cross-validation) – see [Stone 1977]. See also [Efron 1986] for a discussion of this correspondence.

### B. The BIC

The Bayesian Information Criterion (BIC) [Schwarz 1978] is very similar to $C_p$ and the AIC

**Definition 11** (The Bayesian Information Criterion)**.** For the linear model and Gaussian errors $\epsilon$

$$\text{BIC} = \frac{N}{\sigma_\epsilon^2}\left[\frac{1}{N}\|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + (\log N)\frac{p}{N}\sigma_\epsilon^2\right]$$

where $N$ is the number of observations in the training set and $p$ is the number of variables in our model.

Notice that the only difference between definition 10 and definition 11 is the factor of $\log N$ instead of the factor of 2. For $N > e^2 \approx 7.4$, $\log N > 2$, and the BIC penalises complex model more severely than the AIC.

As with the AIC, we calculate the BIC for every candidate model, and choose the model with *lowest* BIC (see section V for details).

Despite its similarity with the AIC, the BIC is derived from entirely different, Bayesian, princples. We give a short heuristic derivation here. For more details, see [Schwarz 1978] ([Kass and Raftery 1995] also provide a more accessible account of the derivation, and [Cavanaugh 2009] provides a similar but less pedantic derivation).

We suppose we have a set of candidate models $\mathcal{M}_m$, $m = 1, 2, \cdots, M$, each with corresponding model parameters $\theta_m$ and with $p_m$ parameters. We wish to choose the best model among them, based on the data in our training set $\mathcal{T}$. The Bayesian approach to this problem is to find the model $\mathcal{M}_m$ that maximises

$$\mathbb{P}(\mathcal{M}_n|\mathcal{T}).$$

Using Bayes' Equation, this can be written

$$\begin{aligned}\mathbb{P}(\mathcal{M}_n|\mathcal{T}) &= \mathbb{P}(\mathcal{M}_n)\cdot\mathbb{P}(\mathcal{T}|\mathcal{M}_n)\\ &= \mathbb{P}(\mathcal{M}_n)\int\mathbb{P}(\mathcal{T}|\theta_n,\mathcal{M}_n)\,\mathbb{P}(\theta_n|\mathcal{M}_n)\,\mathrm{d}\theta_n.\end{aligned} \quad (12)$$

(Here, $\mathbb{P}(\mathcal{M}_n)$ is a prior distribution over all the possible models $\mathcal{M}_m$).

Now, consider the integral:

- The first term in the integral is simply the likelihood – the probability of obtaining the data in our training set given a certain model and associated parameter.

- The second term is the probability of getting a particular parameter given a certain model. In a way, it's a kind of "prior probability" for parameters in a given model.

Now, if we were to assume that the only tennable parameter for any given model is the maximum likelihood parameter (MLE)[5] $\hat{\theta}_n$, then we would have $\mathbb{P}\left(\theta_n|\mathcal{M}_n\right) = \mathbb{1}_{\theta_n=\hat{\theta}_n}$, and the integral above would simply be equal to the likelihood at the maximum likelihood estimator $\mathbb{P}(\mathcal{T}, \hat{\theta}_n|\mathcal{M}_n)$.

However, this is, of course, *not* the case. The maximum likelihood estimator has non-zero variance, and so we can't be *absolutely sure* that $\theta = \hat{\theta}$. Other nearby parameters are also tennable. This tends to make the integral smaller (because some weight is given to smaller likelihoods).

This heuristic argument can be formalised using a so-called Laplace Expansion (see the references above for details), and the integral can be written as

$$\log \mathbb{P}\left(\mathcal{T}|\mathcal{M}_n\right) = \log \mathbb{P}\left(\mathcal{T}|\hat{\theta}_n, \mathcal{M}_n\right) - \frac{p_n}{2}\log N + O(1).$$

Returning to equation 12, we have

$$\begin{aligned}\log \mathbb{P}\left(\mathcal{M}_n|\mathcal{T}\right) = \log\left(\mathbb{P}\left(\mathcal{M}_n\right)\right)\\ + \log \mathbb{P}\left(\mathcal{T}|\hat{\theta}_n, \mathcal{M}_n\right) - \frac{p_n}{2}\log N.\end{aligned}$$

Now, assuming a uniform prior over all models, the first term can be dropped, and we are left with the task of maximising

$$\log \mathbb{P}\left(\mathcal{T}|\hat{\theta}_n, \mathcal{M}_n\right) - \frac{p_n}{2}\log N.$$

This problem is equivalent to the *minimisation* of the *Schwarz criterion*

$$-\log \mathbb{P}\left(\mathcal{T}|\hat{\theta}_n, \mathcal{M}_n\right) + \frac{d_n}{2}\log N.$$

In the case of the linear model with Gaussian errors $\epsilon$, this reduces to the BIC (definition 11).

It is often difficult to decide whether to use the BIC or AIC. The BIC has the advantage of being asymptotically consistent – as $N \to \infty$, BIC will select the correct model. However, with finite sample sizes, the BIC often chooses models that are too small due to its heavy penalties on complex models.

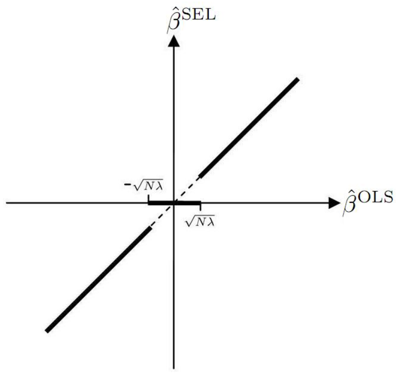

FIG. 3: The behaviour of models like the $C_p$, the AIC and the BIC. The ordinary least-squares estimates is calculated, and any small components are shrunk to 0. This is called *hard thresholding* in the literature.

### C. General analysis

The AIC and BIC have involved taking the training error and correcting it to obtain an estimate of the expected generalisation error. These are only two of a very large number of estimators of their kind (see, for example, [Hocking 1976]). Most of these estimators have in common the aim to minimise a quantity that is monotonically increasing with $\overline{\text{err}}$. The AIC and BIC, for example, both call for the minimisation of

$$\begin{aligned}Q(\boldsymbol{\beta}) &= \overline{\text{err}} + \lambda \left\|\boldsymbol{\beta}\right\|_0 && (13)\\ &= \frac{1}{N}\left\|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\right\|^2 + \lambda\left\|\boldsymbol{\beta}\right\|_0 && (14)\end{aligned}$$

(Where $p = \|\boldsymbol{\beta}\|_0$ denotes the $L_0$ norm of the vector $\boldsymbol{\beta}$ – in other words, the number of non-zero components in that vector). In the case of $C_p$ and AIC, we had $\lambda = 2\sigma_\epsilon^2/N$, and in the case of BIC, we had $\lambda = \sigma_\epsilon^2 \log N/N$

It is helpful to analyse this aim in the case of an orthonormal design matrix (section II D). Using the result of theorem 5 and the ensuing discussion, we find that when $\mathbf{X}^T\mathbf{X} = \mathbf{I}$, the problem of minimising equation 14 is equivalent to the problem of minimising

$$q(\beta_j) = \frac{1}{N}\left(\hat{\beta}_j^{\text{OLS}} - \beta_j\right)^2 + \lambda\mathbb{1}_{\beta_j\neq 0}$$

For each $\beta_j$ separately. The solution to this minimisation problem for each $\beta_j$ is clearly

$$\hat{\beta}_j^{\text{SEL}} = \hat{\beta}_j^{\text{OLS}}\mathbb{1}_{|\hat{\beta}_j^{\text{OLS}}|>\sqrt{N\lambda}}$$

In other words, this method find the ordinary least-squares estimate of $\boldsymbol{\beta}$ and considers each component of the estimate. Any component of magnitude larger than $\sqrt{N\lambda}$ is not shrunk. Other components are shrunk to 0. This behaviour is illustrated in figure 3.

It is informative to consider this in terms of the bias-variance tradeoff. By shrinking some components of $\boldsymbol{\beta}$,

[5] The maximum likelihood parameter $\hat{\theta}_n$ is parameter $\theta$ that maximises the likelihood $\mathbb{P}\left(\mathcal{T}, \theta|\mathcal{M}_n\right)$

we introduce some bias – because the unbiased ordinary-least squares estimate clearly indicates that these components are *not* 0. However, in fixing these components to 0, we also reduce their *variance* to 0 – and we therefore *increase* our prediction accuracy in that way.

If $\beta_j$ is very large, shrinking it to 0 will reduce in a very large increase in bias – the resulting decrease in variance will probably not be sufficient to offset this. If, on the other hand, $\beta_j$ is small, the decrease in variance is likely to offset the increase in variance. The methods above set the "cut-off" point at $\beta_j = \sqrt{N\lambda}$. In the case of $C_p$ and the AIC, this is $\beta_j = 2\sigma_\epsilon^2$ and in the case of BIC, this is $\beta_j = \sigma_\epsilon^2 \log N$.

### D. Penalised Least Squares

The form of equation 13 suggests a more general class of methods which minimise the quantity

$$\begin{aligned} Q(\boldsymbol{\beta}) &= \overline{\mathrm{err}} + \sum_{j=1}^{d} p_\lambda(|\beta_j|) \\ &= \frac{1}{N}\left\|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\right\|^2 + \sum_{j=1}^{d} p_\lambda(|\beta_j|) \end{aligned} \tag{15}$$

Where $p_\lambda$ is a function of $|\beta_j|$, called the *penalty function*. $p_\lambda$ denotes the dependence of the function on a *regularisation parameter* $\lambda$ – a way for the strength of the penalty to 'tweaked'.

At this juncture, penalised least squares does not seem like a particularly desirable procedure – unlike the other procedures we have discussed, there does not seem to be any good theoretical motivation for its use. Despite this shortcoming, penalised least squares has been one of the most successful methods for high dimensional model selection.

## V. PRACTICALITIES & THE PROBLEM OF HIGH DIMENSIONS

To understand the need for the method of penalised least squares, we must consider the way methods like $C_p$, BIC, etc. are applied. Each of these measures are an estimate of Err, which we are trying to minimise, and are all in the form

$$\overline{\mathrm{err}} + \lambda p$$

This means that once we have chosen *which* variables will be included in our model (ie: will be non-zero in $\boldsymbol{\beta}$), $d$ is fixed and we simply need to minimize $\overline{\mathrm{err}}$ by finding the ordinary least-squares solution of equation 4.

This suggests the following method

- Build every possible combination of variables in $\boldsymbol{\beta}$
- For each of these combinations, find the ordinary least-squares estimate, work out $\overline{\mathrm{err}}$, and the appropriate estimate of Err.
- Pick the combination with the lowest estimate of Err.

The issue with this method, of course, is that when the model is high-dimensional, the first step entails an enormous computational burden (if the original input vectors $\boldsymbol{X}$ contain $d$ components, there are $2^d - 1$ possible combination of variables!). This makes these methods highly impractical.

That said, it should be noted that a significant amount of work has gone into reducing this burden, in two ways

- Reducing the time it takes to calculate the ordinary least-squares estimate for each model, by updating the hat matrix $\mathbf{H}$ rather than re-calculating it for each model, and by regularly removing superfluous data from these matrices. See [Furnival 1971] and papers referenced therein for details.
- Finding clever ways to exclude some combinations as infeasible without examining them. Most such methods rely on the fact that it is impossible to reduce $\overline{\mathrm{err}}$ by removing variables from our model (indeed, figure 1 clearly shows that $\overline{\mathrm{err}}$ is monotonously decreasing as $d$ increases).

  The concept is best illustrated by example. Imagine we are trying to minimise a quantity

  $$Q = \overline{\mathrm{err}} + d$$

  Imagine further than during the course of our calculations, we have already found a 1-variable combination $\mathcal{A}$ with $Q_{\mathcal{A}} = 2$. We now come accross a 3-variable combination $\mathcal{B}$ with $Q_{\mathcal{B}} = 5$. The method allows us to ignore all 7 'subsets' of $\mathcal{B}$ – because even if $\overline{\mathrm{err}}$ does not increase as we consider these subsets, the best we'll ever be able to get is $Q = Q_{\mathcal{B}} - 2 = 3$, which does not 'beat' $Q_{\mathcal{A}} = 2$. We have therefore reduced our search space by 7. Such methods are an example of branch and bound methods.

  This example is, of course, very simplistic. See [LaMotte and Hocking 1970] for a more rigorous discussion.

These two methods are combined in the 'leaps and bounds' procedure of Furnival and Wilson [Furnival and Wilson 1974]. This procedure is able to cope with 30-40 variables, but this falls grossly short of the requirements of modern high-dimensional problems, which often include thousands of variables.

Some heuristic algortihms also exist to aid this process. For example

- Forward selection is a greedy algortihm, which starts with an empty model, and successively adds the variable that is most heavily correlated with the response.

- Backwards elimination starts with a model containing *all* variables, and removes the least significant ones one-by-one (this model is not applicable were the number of variables is greater than $N$, in which no ordinary least-squares estimate exists).

- Stepwise selection is basically forward selection, but with the possiblity of deleting a single variable – backwards-selection-style – at each forward step. This heuristic procedure has recently been formalised and thoroughly analysed by [Zhang 2008], under the name of FOBA, and promises to rival the other techniques quoted in this paper.

These methods, of course, all require 'stopping rules'.

Despite these methods, classical model selection remains a very difficult problem computationally, especially for models that lie in higher dimensions. How can penalised least squares help? The answer is that when $p$ is a convex function, the problem of minimising equation 15 becomes a convex optimisation problem, which is exactly solvable in polynomial time. (See, for example, [Boyd and Vandenberghe 2004] for an introduction to the theory of convex optimisation). These expressions are not as theoretically attractive as measures such as the AIC (and we shall see in section VI that they often lack some of the desirable properties of these measures), but at least they are tractable, and can often lead to remarkably good solutions.

In the next part of this paper, we consider these methods in more detail.

# Part II

# Penalised Least Squares

In this section, we consider the method of penalised least squares (introduced in section IV D) in more detail. This method suggests that we should choose a $\beta$ that minimises

$$\frac{1}{N}\left\|\boldsymbol{Y}-\mathbf{X}\boldsymbol{\beta}\right\|^2+\sum_{j=1}^{d}p_\lambda(|\beta_j|) \qquad (16)$$

In this section, we will consider a wide range of penalties $p_\lambda$. In analysing them, it will be useful to consider the orthonormal design case (see section II D). In the orthonormal design case, we can use Theorem 5 to show that minising equation 16 is equivalent to minimising

$$q(\beta_j)=\frac{1}{N}\left(\hat{\beta}_j^{\mathrm{OLS}}-\beta_j\right)^2+p_\lambda(|\beta_j|) \qquad (17)$$

for each $\beta_j$.

Before we begin, however, we first consider the properties that we would want from our $p_\lambda$.

## VI. GENERAL CONSIDERATIONS

Fan and Li [Fan and Li 2001] suggest that any penalty function $p_\lambda$ should result in an estimator $\boldsymbol{\beta}$ that fulfils the following properites

**Sparsity** – we have already seen that in some ultra-high dimensional models, we have prior reasons to believe that some of the components of $\boldsymbol{\beta}$ should be 0. We would therefore hope that any $p_\lambda$ we choose will result in a method that sets some components of $\boldsymbol{\beta}$ to 0.

Sparsity is also advantageous in shrinking the "effective number of parameters" in the model (definition 7) and reducing the optimism $\omega$, but less critical. Indeed, we will see that even if parameters are not shrunk to *exactly* 0, a reduction in the effective number of parameters still occurs.

**Unbiasedness** – as we saw in the previous section, penalised least squares and similar methods *introduce* bias into $\boldsymbol{\beta}$ in the hope of reducing variance. However, this phenomenon mostly occurs where $\beta_j$ is small and our method shrinks it to 0. We would therefore hope that any $p_\lambda$ we choose will result in an estimator that is approximately unbiased for *large* components of $\boldsymbol{\beta}$.

**Continuity** – to prevent instabilities in prediction, we would like our predictor to be continuous in $\hat{\boldsymbol{\beta}}^{\mathrm{OLS}}$ (ie: in the data in our training set).

We consider each of these conditions in detail in the orthonormal design case. Recall that in that case, our aim is to minimise equation 17 for each component $\beta_j$. Now, let $\Delta = 2/N$ and let $\mathrm{sgn}(x)$ be the sign of $x$. The derivative[6] of equation 17 with respect to $\beta_j$ is then

$$\begin{aligned}q'(\beta_j)&=-\Delta\hat{\beta}_j^{\mathrm{OLS}}+\Delta\beta_j+\mathrm{sgn}(\beta_j)p'_\lambda\left(|\beta_j|\right)\\&=\mathrm{sgn}(\beta_j)\left\{\Delta|\beta_j|+p'_\lambda\left(|\beta_j|\right)\right\}-\Delta\hat{\beta}_j^{\mathrm{OLS}}\end{aligned}$$

'Minimising' $q(\beta_j)$ is equivalent to finding the point at which $q'(\beta_j)=0$.

### A. Sparsity

For sparsity, we require our method to at least have the *ability* to set some of the $\beta_j$ to 0. Or in other words, we require the existence of some cases in which $q'(0)=0$.

[6] Strictly speaking, this derivative only exists when the penalty function $p_\lambda$ is differentiable everywhere. In other cases, the formalism of subgradients must be applied – see Appendix E for details.

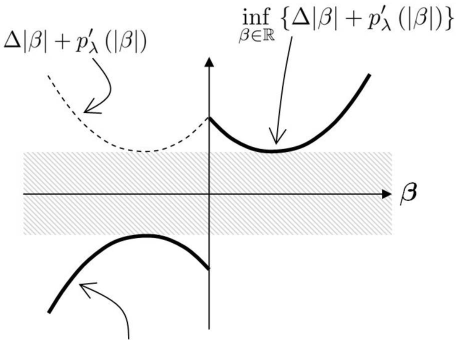
$\Delta|\beta| + p'_\lambda(|\beta|)$
$\inf_{\beta\in\mathbb{R}} \{\Delta|\beta| + p'_\lambda(|\beta|)\}$
$\beta$
$\mathrm{sgn}(\beta)\{\Delta|\beta| + p'_\lambda(|\beta|)\}$

FIG. 4: An illustration of the sparsity condition. The heavy line is the function $\mathrm{sgn}(\beta)\{\Delta|\beta| + p'_\lambda(|\beta|)\}$ and the dashed line is the function $\Delta|\beta| + p'_\lambda(|\beta|)$. If $\hat{\beta}^{\mathrm{OLS}}$ falls in the zone shaded in grey above, it is clear that $q'(\beta) = $ *heavy line* $- |\hat{\beta}^{\mathrm{OLS}}|$ will be positive for positive $\beta$ and negative for negative $\beta$. However, if the minimum of $\Delta|\beta| + p'_\lambda(|\beta|)$ (dashed line) is not positive (ie: condition 19 fails), there is no grey zone for this to occur.

Now, consider a situation in which

$$\Delta|\hat{\beta}^{\mathrm{OLS}}| < \inf_{\beta\in\mathbb{R}} \{\Delta|\beta| + p'_\lambda(|\beta|)\} \tag{18}$$

In that case, $q'(\beta)$ is positive for $\beta > 0$, and negative for $\beta < 0$, and therefore $q'(0) = 0$, precisely as required.

For condition 18 to be applicable, however, we require the infimum on the RHS to be positive. Thus, our condition for sparsity is

$$\boxed{\inf_{\beta\in\mathbb{R}} \{\Delta|\beta| + p'_\lambda(|\beta|)\} > 0} \tag{19}$$

This concept is illustrated in Figure 4.

### B. Unbiasedness

We know that the ordinary least-squares estimator $\hat{\boldsymbol{\beta}}^{\mathrm{OLS}}$ is unbiassed (see section II A). Thus, we simply require our estimator to be equal to $\hat{\boldsymbol{\beta}}^{\mathrm{OLS}}$ for high values of $\beta$.

Consider that if $p'_\lambda(|\beta|) = 0$, $q'(\beta) = \beta - \hat{\beta}^{\mathrm{OLS}}$ and therefore clearly has a root at $\beta = \hat{\beta}^{\mathrm{OLS}}$, as required.

Thus, our requirement for unbiasedness is

$$\boxed{p'_\lambda(|\beta|) \to 0 \text{ as } |\beta| \to \infty} \tag{20}$$

### C. Continuity

For our estimator $\beta$ to be continuous, it must never 'jump' from one value to another discountinuously as the input data is changed.

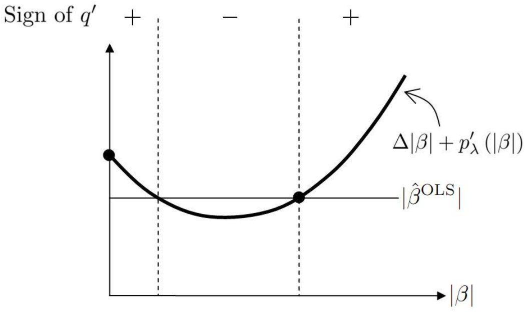
Sign of $q'$
+
−
+
$\Delta|\beta| + p'_\lambda(|\beta|)$
$|\hat{\beta}^{\mathrm{OLS}}|$
$|\beta|$

FIG. 5: And illustration of what happens when the continuity condition (equation 21) is not met. The heavy line is $\Delta|\beta| + p'_\lambda(|\beta|)$, and the normal line is $\hat{\beta}^{\mathrm{OLS}}$. Thus, $q'(\beta) =$ *Solid line - normal line*. The sign of $q'(\beta)$ at various points is indicated in the diagram, and clearly shows that $q(\beta)$ first increases, then decreases and then increases again. This means that the minimum of $q$ must occur at one of the points indicated by black dots. However, *which* of those two points it occurs at depends on exactly where $\hat{\beta}^{\mathrm{OLS}}$ is, and there is therefore a point at which our solution discontinuously jumps from one point to the other.

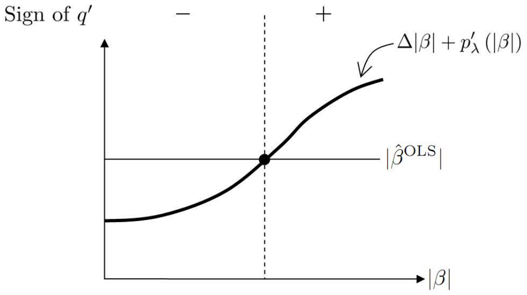
Sign of $q'$
−
+
$\Delta|\beta| + p'_\lambda(|\beta|)$
$|\hat{\beta}^{\mathrm{OLS}}|$
$|\beta|$

FIG. 6: And illustration of what happens when the continuity condition (equation 21) is met. Details as in figure 5. In this case, $q(\beta)$ decreases to a minimum and then increases again. There is only one possible solution, indicated by a black dot, which varies continuously as $\hat{\beta}^{\mathrm{OLS}}$ changes.

We have already seen that if the condition in equation 18 is fulfilled, $\beta$ is shrunk to 0. Therefore, let us consider the other possibility

$$\Delta|\hat{\beta}^{\mathrm{OLS}}| > \inf_{\beta\in\mathbb{R}} \{\Delta|\beta| + p'_\lambda(|\beta|)\}$$

In this situation, a necessary and almost sufficient[7] condition for continuity is that the minimum of $\Delta|\beta| +$

[7] The condition is 'almost' sufficient because we also require that the function $\Delta|\beta| + p'_\lambda(|\beta|)$ be *unimodal* for all $\beta \in \mathbb{R}$. In other words, we require there to be a $t^*$ such that the function is decreasing for all $t < t^*$ and increasing for all $t > t^*$. If this were not the case, the curve in figure 6 might 'come down' and cross the line a second time, creating the possibility for a discontinuity. This subtlety was omitted by [Fan and Li 2001], and I am grateful to Richard Samworth for pointing it out to me.

$p'_\lambda(|\beta|)$ be attained at 0. In other words

$$\boxed{\arg\inf_{\beta\in\mathbb{R}}\left\{\Delta|\beta| + p'_\lambda(|\beta|)\right\} = 0} \qquad (21)$$

Figure 5 explains why we lose continuity when this is *not* the case, and figure 6 explains how this condition fixes things.

### D. Classical model selection

To conclude our discussion of the general conditions on $p_\lambda$, it is instructive to ask whether these conditions are met for classical model selection procedures (AIC, BIC, etc...), for which we saw

$$p_\lambda(|\beta_j|) = \lambda\mathbb{1}_{(|\beta_j|\neq 0)}$$

First, we note that the derivative of $p_\lambda$ is the Dirac delta function

$$p'_\lambda(|\beta_j|) = \lambda\delta(|\beta_j|)$$

Now, consider our three properties

**Sparsity** The sparsity condition in this case is

$$\inf_{\beta\in\mathbb{R}}\left\{\Delta|\beta| + \lambda\delta(|\beta|)\right\} > 0$$

Clearly, this infimum cannot be negative, because both its components are positive. Furthermore, at $\beta = 0$, the function is clearly not 0 because the Dirac delta function spikes. Thus, the infimum is greater than 0, and the condition is fulfilled.

**Unbiasedness** Clearly, for large $|\beta|$, the Dirac delta function vanishes. Thus, the unbiasedness condition is fulfilled.

**Continuity** The continuity condition in this case is

$$\arg\inf_{\beta\in\mathbb{R}}\left\{\Delta|\beta| + \lambda\delta(|\beta|)\right\} = 0$$

Unfortunately, it is clear that the continuity condition is *not* met – we could hardly claim that a minimum occurs at $\beta = 0$, where the Dirac delta function spikes.

Classical model selection, therefore, results in solutions that are sparse and unbiassed, but which are unstable with respect to input conditions. See [Breiman 1996] for an extended discussion of this phenomenon. Clearly, therefore, this is another advantage of penalised least squares over classical model selection, as well as tractability.

We are now ready to begin our discussion of various specific forms of $p_\lambda$.

## VII. BRIDGE REGRESSION

Two of the most important penalty functions we will consider – the Ridge penalty and the LASSO penalty are special cases of the *bridge regression penalty* [Frank and Friedman 1993]

$$p_\lambda(\beta_j) = \lambda|\beta_j|^\gamma \qquad \gamma \neq 0 \qquad (22)$$

The $\gamma = 0$ case can be thought of as corresponding to classical model selection. Since we usually define $0^0 = 1$, definition 22 does not include this special case.

In this section, we consider a number of general properties of bridge regression. We will then discuss the Ridge and the LASSO in more detail.

### A. Conditions on $p_\lambda$

We begin by examining which of the conditions in section VI the bridge penalty satisfies. We note that in all cases for which $\gamma \neq 0$

$$p'_\lambda(\beta_j) = \lambda\gamma|\beta_j|^{\gamma-1}$$

(The case $\gamma = 0$ corresponds to classical model selection and was dealt with at the end of section VI).

**Sparsity** The sparsity condition in this case is

$$\inf_{\beta\in\mathbb{R}}\left\{\Delta|\beta| + \lambda\gamma|\beta_j|^{\gamma-1}\right\} > 0$$

We then have the following behaviour:

- For $\gamma > 1$, the function is well defined at $\beta = 0$, and so the infimum occurs $\beta = 0$, taking a value of 0. Thus, sparsity is *not* fulfilled.
- For $\gamma \leq 1$, the function blows up as $\beta \to 0$, and the infimum *is* therefore greater than 0. Sparsity *is* therefore fulfilled.

**Unbiasedness** For large $|\beta|$, $p'_\lambda$ only vanishes if $\gamma < 1$. Thus, the Bridge penalty is only unbiased if $\gamma < 1$.

**Continuity** The continuity condition in this case is

$$\arg\inf_{\beta\in\mathbb{R}}\left\{\Delta|\beta| + \lambda\gamma|\beta_j|^{\gamma-1}\right\} = 0. \qquad (23)$$

We then have the following behaviour

- For $\gamma > 1$, the function takes the value of 0 at $\beta = 0$ – this is clearly the infimum. Thus, continuity *is* fulfilled.
- For $\gamma = 1$, $p_\lambda$ is not differentiable, and we need to invoke the concept of subgradients (see Appendix E). The subgradient of $|\beta|$ is

$$\partial|\beta| = \begin{cases} -1 & \beta < 0 \\ \{\theta : \theta \in [-1, 1]\} & \beta = 0 \\ 1 & \beta > 0 \end{cases}$$

Clearly, the smallest value the subgradient can have (-1) is realised that $\beta = 0$. Thus, the infimum does indeed occur at $\beta = 0$.

- For $\gamma < 1$, $p_\lambda$ is neither differentiable nor convex, so even the concept of subgradients (as defined in Appendix E) is of no assistance.

  In practice, however, it is found that method with $\gamma < 1$ are *not* continuous. This is intuitively understandable – a careful look at equation 23 indicates that the derivative of the penalty tends to infinity as $\beta \to 0$. It seems unlikely, therefore, that 23 would be minimised *at* $\beta = 0$.

Another important aspect to consider is the *convexity* of the penalty function. A problem involving a convex penalty function can easily be optimised. Others are much more difficult. In this case, only penalty functions with $\gamma \geq 1$ are convex.

This information is summarised in table I. Clearly, no single penalty satisfies all conditions. The LASSO has the advantage of being convex, continuous and sparse. In section VIII, we will meet a penalty function that is sparse, continuous, and unbiased.

### B. Penalty plots

Before we dive into the mathematics of Bridge regression, it is useful to consider the form of the bridge penalty functions for various values of $\gamma$. These are illustrated in figure 7.

These figures go a long way towards explaining the behaviour of these ridge penalties:

- For $\gamma \leq 1$, it is clear that the penalties favour the directions along the coordinate axes more than others – in other words, the penalties favour directions for which one of the coordinates is *small.* Thus, the penalties lead to sparse $\boldsymbol{\beta}$ vectors.

- For $1 < \gamma < 2$, the penalties still favour the directions along the coordinate axes, but less so. Furthermore, the distributions no longer have 'corners' at the coordinate axes – this, it turns out, is the reason these penalties do not lead to sparse solutions. We motivate this fact informally in Appendix E, in the context of sub-gradients, and see [Tibshirani 1996] for a geometric explanation.

- For $\gamma = 2$, the penalty does not favour any particular direction. Shrinkage still occurs, of course, because the norm of the vector $\boldsymbol{\beta}$ is shrunk, but in no particular direction.

- For $\gamma > 2$, the penalty favours the directions *away* from the coordinate axes – in other words, it favours *large* parameters! This is clearly not desirable in the context of shrinkage.

These observations go a long way to explain the formal results in the previous section, and those we will discuss when we consider the orthonormal design case.

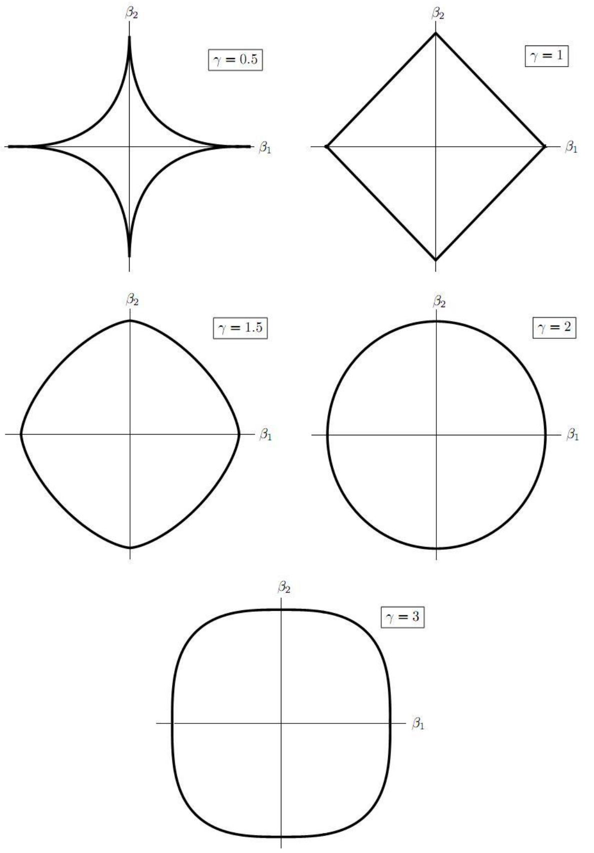

FIG. 7: The Bridge penalties for various values of $\gamma$, in the case in which the vector $\boldsymbol{\beta}$ only has two components – $\beta_1$ and $\beta_2$. The lines drawn are contours of equal $|\beta_1|^\gamma + |\beta_2|^\gamma$.

### C. The Bayesian Approach

It is also very insightful to look at Bridge regression from a Bayesian perspective. Once again, this gives us some intuition behind the formal results.

The minimisation of the Bridge penalty can also be thought of as the maximisation[8] of a log posterior distribution of $(\boldsymbol{\beta}|\boldsymbol{Y})$ given by

$$(\boldsymbol{\beta}|\boldsymbol{Y}) \sim C \exp\left(-\frac{1}{N}\left\|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\right\|^2 - \lambda' \sum |\beta_j|^\gamma\right)$$

It will be more convenient to write this as follows

$$(\boldsymbol{\beta}|\boldsymbol{Y}) \sim C \exp\left(-\frac{1}{2}\left\|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\right\|^2 - \lambda \sum |\beta_j|^\gamma\right)$$

[8] Note that in a Bayesian context it is more usual to use the posterior *mean* rather than the posterior *mode* (which is the quantity we use here). In the case of Ridge regression ($\gamma = 2$), the posterior is Gaussian and so these two measures coincide. This is not the case for other values of $\gamma$.

| | Classical<br>model selection | Bridge regression<br>$\gamma < 1$ | (LASSO)<br>$\gamma = 1$ | (Ridge)<br>$\gamma > 1$ | Elastic<br>net | SCAD |
|---|---|---|---|---|---|---|
| Sparsity | ✓ | ✓ | ✓ | | ✓ | ✓ |
| Unbiasedness | ✓ | ✓ | | | | ✓ |
| Continuity | | | ✓ | ✓ | ✓ | ✓ |
| Convexity | | | ✓ | ✓ | ✓ | |

TABLE I: Properties of the penalty functions in this paper.

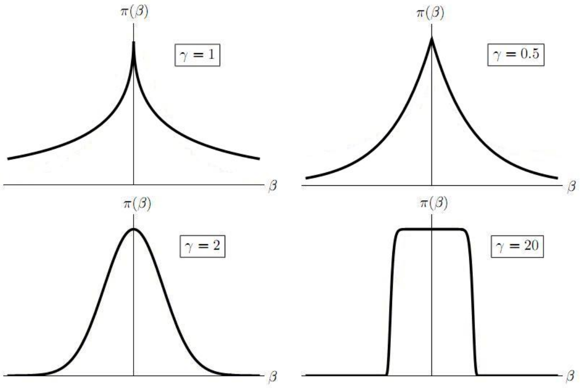

FIG. 8: The implied prior $\pi(\beta)$ on $\beta$ in Bridge regression, for various values of $\gamma$. Note that these plots were all made with the same value of $\lambda$ – in reality, Bridge regression at these values of $\gamma$ would use different values of $\lambda$ to optimise the fit (see section IX).

This modification does not imply any loss of generality, because the changes can be absorbed into the arbitrary constants $C$ and $\lambda$.

Now, Bayes' Theorem relates the prior and posterior distributions as follows

$$f(\boldsymbol{\beta}|\boldsymbol{Y}) \propto f(\boldsymbol{Y}|\boldsymbol{\beta})\pi(\boldsymbol{\beta})$$

(where $\pi((\boldsymbol{\beta})) = f(\boldsymbol{\beta})$ is the prior distribution of $\boldsymbol{\beta}$). Furthermore, if the errors $\epsilon$ are normally distributed, then

$$f(\boldsymbol{Y}|\boldsymbol{\beta}) \propto \exp\left(-\frac{1}{2}\left\|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\right\|^2\right)$$

By simple algebra, it follows that the priors on each components of $\boldsymbol{\beta}$ are given by

$$\pi(\beta) \propto \exp\left(-\beta^{\gamma}\right)$$

This prior is plotted for a few values of $\gamma$ in figure 8, and provides an invaluable insight into the way Bridge regression works.

- Ordinary least squares has a uniform prior.
- Bridge regression, however, involves a prior that places greater weight on small values of $\beta$ than large ones, and this causes shrinkage.
- For $\gamma > 1$, the distribution is slightly 'flat' at its maximum, which means that variables tend not to be shrunk all the way to $\beta = 0$.
- As $\gamma$ increases, the tails of the prior shrink, and this introduces bias into the model for large values of $\beta$. To the extent that for very high values of $\gamma$, low values of $\beta$ are not shrunk at all, but high ones are.

### D. Orthonormal Design

To gain further insight into the behaviour of these penalties, we consider them in the orthonormal design case of section II D. In that situation, the penalties take the form of equation 17

$$q(\beta_j) = \frac{1}{N}\left(\hat{\beta}_j^{\mathrm{OLS}} - \beta_j\right)^2 + \lambda|\beta_j|^{\gamma}$$

We would like to examine the behaviour of this function in the following cases

- $0 < \gamma < 1$
- $\gamma = 1$
- $1 < \gamma < 2$
- $\gamma = 0$
- $\gamma > 2$

The cases $\gamma = 1$ and $\gamma = 2$ admit analytical solutions. Some other special cases do as well, but the details are cumbersome and unenlightening, and we resort to simulations instead. We present our findings for the behaviour of Bridge regression penalties in figure 9.

We now consider each case above, and give details of the analytic solution (if it exists) and comment on the results.

Once again, we write $\Delta = 2/N$ and to simplify notation, we also write $\tilde{\lambda} = \lambda/\Delta$.

#### *1.* $0 < \gamma < 1$

Unfortunately, it is not possible to find analytic solution for $q(\beta)$ for any value of $\gamma$ in this range. We used

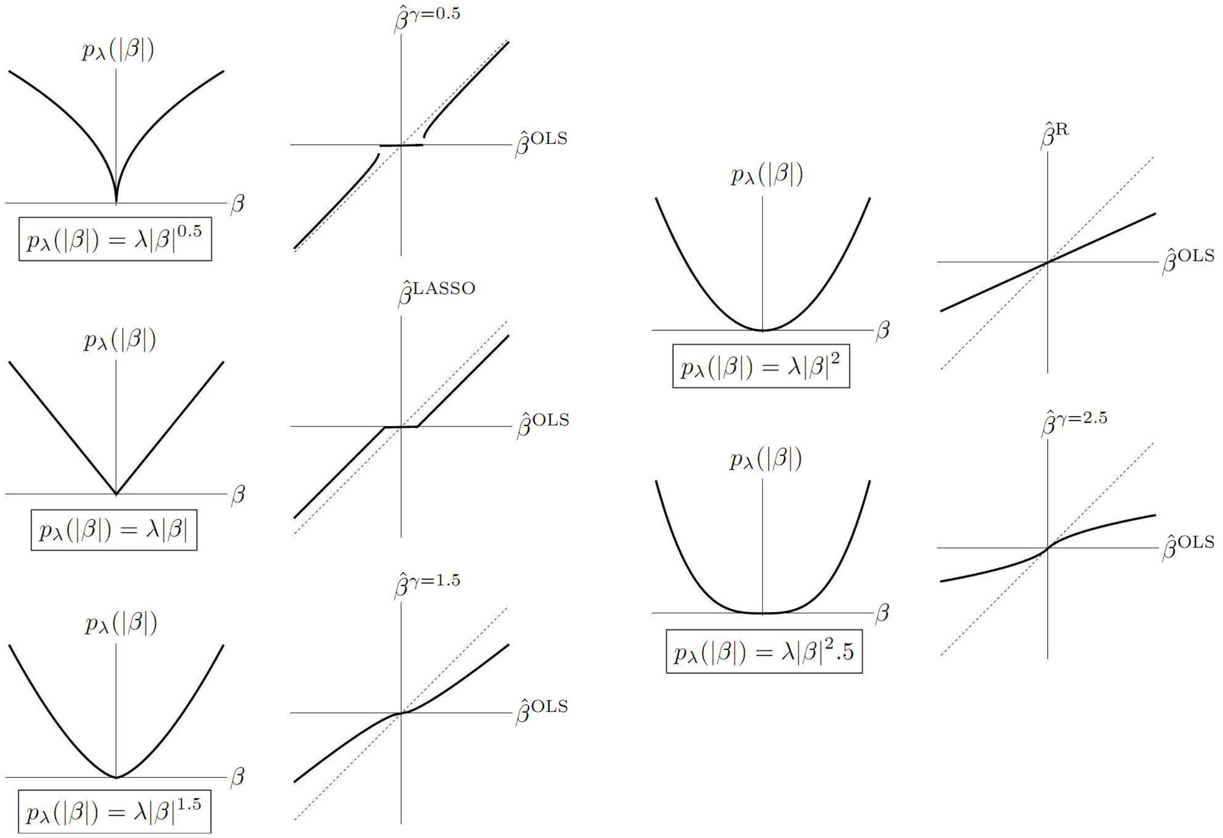

FIG. 9: The Bridge penalties for various values of $\gamma$, and the resulting estimators $\hat{\beta}$. In each diagram, the dotted line is $\hat{\beta}^{\gamma} = \hat{\beta}^{\text{OLS}}$, drawn for reference. See text of the paper for comments.

simulations, however, to examine the behaviour of $q$. Due to the singularity at the origin, $q$ always has a minimum at the origin. For large enough $\lambda$, $q$ also has a minimum elsewhere. This leads to a non-linear thresholding behaviour, as illustrated in figure 9 for $\gamma = 0.5$. The estimator is unbiased for large $\beta$, *and* successfully performs variable selection.

### *2.* $\gamma = 1$

This is the LASSO. In this case

$$q(\beta_j) = \frac{1}{N}\left(\hat{\beta}_j^{\text{OLS}} - \beta_j\right)^2 + \lambda|\beta_j|$$

Minimising such a non-differentiable function requires the use of subgradients – see Appendix E for details. The solution is

$$\hat{\beta}^{\text{LASSO}} = \text{sgn}(\hat{\beta}^{\text{OLS}})\left(|\hat{\beta}^{\text{OLS}}| - \tilde{\lambda}\right)_+$$

Where $x_+ = \max(x, 0)$.

This kind of function is called *soft thresholding* in the literature. It is very clear from the diagram how the LASSO biases the estimates of higher values of $\beta$.

### *3.* $1 < \gamma < 2$

The case $\gamma = 1.5$ admits an analytic solution here, but the algebraic details are extremely involved, and completely uninteresting. We therefore omit details, and simply sketch the result.

The diagram clearly shows that in this range of $\gamma$, we shrink small variables components of $\beta$ by a large amount, and large components of $\beta$ by a lesser amount.

### *4.* $\gamma = 2$

This is Ridge regression. In this case

$$q(\beta_j) = \frac{1}{N}\left(\hat{\beta}_j^{\text{OLS}} - \beta_j\right)^2 + \lambda|\beta_j|^2$$

The analytical solution in this case is simple

$$\hat{\beta} = \frac{1}{2\tilde{\lambda} + 1}\hat{\beta}^{\text{OLS}}$$

In other words, Ridge regression performs *proportional shrinkage*; the larger the variable, the more it is shrunk.

### 5. $\gamma > 2$

Once again, we resort to simulations and produce a plot of the solution.

In this case, we clearly retain small components of $\beta$ and shrink larger components, as we predicted from the form of the penalty. Clearly, this behaviour is not particularly useful in the context of this paper.

### 6. *Summary*

We see, therefore, that Bridge regression encompasses a wide range of behaviours, both in terms of small and large components.

Small components are either thresholded or not. Whether this happens depends on whether the penalty has a corner at the origin ($\gamma \leq 1$) or not ($\gamma > 1$).

The shrinakge of non-thresholded components depends on the concavity of the function. Less concavity leads to more shrinkage of small components, and large concavity leads to more shrinkage of large components.

## E. Additional Points on Ridge Regression

Ridge regression is simply Bridge regression with $\gamma = 2$. It was originally proposed by [Hoerl and Kennard 1970], with a very different motivation to that in this paper. They were looking to deal with the problem of co-linearity in the columns of $\mathbf{X}$. If the columns of $\mathbf{X}$ are colinear, then the determinant of $\mathbf{X}^T\mathbf{X}$ is likely to be very small (because it is the square of the volume of the parallelepiped whose edges are the columns of $\mathbf{X}$). In turns, this means that $(\mathbf{X}^T\mathbf{X})^{-1}$ is likely to have some very large eigenvalues. This is problematic, because we saw in section II A that these eigenvalues are the variances of the components of $\boldsymbol{\beta}$. (Alternatively, in terms of principal components, small eigenvalues of $\mathbf{X}^T\mathbf{X}$ implies that the variance is small along some principal components, meaning the model is hard to specify there). This problem is especially likely to happen in high dimensional problems, where $\mathbf{X}$ has many rows and few columns.

Ridge regression aims to solve this problem by minimising

$$Q_2(\boldsymbol{\beta}) = \frac{1}{2}\left\|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\right\|^2 + \tilde{\lambda}\|\boldsymbol{\beta}\|^2$$

(Remember that $\tilde{\lambda} = \Delta\lambda/2$).

The solution to this equation is unique among solutions to penalised least squares problems, in that it can be writen in closed form. Differentiating and setting to 0 leads to

$$\hat{\boldsymbol{\beta}}^{\mathrm{R}} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\boldsymbol{Y}$$

The resulting hat matrix is

$$\mathbf{H}^{\mathrm{R}} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T \tag{24}$$

Intuitively, we see that Ridge regression simply increases the eigenvalues of $\mathbf{X}^T\mathbf{X}$.

We can gain a much deeper understanding of what happens by considering the singular-value decomposition of $\hat{\boldsymbol{Y}}^{\mathrm{R}}$ (section II B). Recall that the singular-value decomposition of $\mathbf{X}$ is $\mathbf{X} = \mathbf{UDV}^T$. We then have

$$\begin{aligned}\hat{\boldsymbol{Y}}^{\mathrm{R}} &= \mathbf{UDV}^T\left(\mathbf{VDU}^T\mathbf{UDV}^T + \lambda\mathbf{I}\right)^{-1}\mathbf{VDU}^T\boldsymbol{Y}\\ &= \mathbf{UDV}^T\left(\mathbf{VD}^2\mathbf{V}^T + \lambda\mathbf{I}\right)^{-1}\mathbf{VDU}^T\boldsymbol{Y}\\ &= \mathbf{UD}\left(\mathbf{D}^2 + \lambda\mathbf{I}\right)^{-1}\mathbf{DU}^T\boldsymbol{Y}\\ &= \sum_{j=1}^{p}\boldsymbol{u}_j\frac{d_j^2}{d_j^2+\lambda}\boldsymbol{u}_j^T\boldsymbol{y}\end{aligned}$$

Comparing this to the analogous expression for ordinary least squares (equation 7), we see that just like ordinary least squares, ridge regression first expresses $\boldsymbol{Y}$ in terms of the principal components of $\boldsymbol{Y}$. But then, before converting back to '$\hat{\boldsymbol{Y}}$-space', it multiplies the components by a factor $d_j^2/(d_j^2+\lambda)$. It seems, therefore, that Ridge regression *shrinks* our projection along the principal components, but shrinks it much more along those principal components with *small* values of $d_j$. These, however, are precisely the components along which our observations are not spread out (ie: along which $\mathbf{X}$ has a low variance) – in other words, these are the very components through which it is difficult to fit a straight line, and that would therefore have a high variance.

Ridge regression, therefore, shrinks our projection precisely along those axes for which the variance of our predictor would be very high otherwise.

We can also use this singular value decomposition to work out the effective number of parameters in ridge regression (definition 7)

$$\begin{aligned}d_{\text{eff}}^{\mathrm{R}} &= \mathbb{T}\mathrm{r}(\mathbf{H})\\ &= \mathbb{T}\mathrm{r}\left(\sum_{j=1}^{p}\boldsymbol{u}_j\frac{d_j^2}{d_j^2+\lambda}\boldsymbol{u}_j^T\right)\\ &= \sum_{j=1}^{p}\frac{d_j^2}{d_j^2+\lambda}\end{aligned}$$

This result is smaller than $p$, which was the effective number of parameters for ordinary least squares. We have therefore succeeded in reducing the optimism of the training error in this case, despite the fact we have not reduced the number of nonzero parameters.

It can be shown that Ridge regression leads to a better estimate (in the mean squared error sense) than ordinary least squares for sufficiently small $\lambda$ (see [Hoerl and Kennard 1970]). Unfortunately, choosing $\lambda$ too large can significantly reduce the quality of the model. See section IX for a discussion on choosing the correct parameter.

### F. Additional comments on the LASSO

A few miscellaneous points on the LASSO

- Unlike the Ridge, the LASSO does not admit analytical solutions. However, it is still possible to make some deductions about the effective number of variables in the LASSO.

  One might be tempted to simply say that $d_{\text{eff}}^{\text{LASSO}}$ is equal to the number of parameters in our model. This however, is too simplistic, because it fails to take into account the search for the correct variables. That said, [Zou Hastie and Tibshirani 2007] show that the number of variables selected is an *unbiased estimator* of the real value of $d_{\text{eff}}^{\text{LASSO}}$.

- The LASSO is convex, and can therefore be calculated with relative ease. In addition, [Efron Hastie Johnstone and Tibshirani 2004] introduced an even faster algorithm, called *least angle regression* (LARS). The technique is useful in its own right, and with a slight modification can also find LASSO solutions.

  Due to space constraints, we will only give a qualitative description of the LARS algorithm. LARS is very similar to forward selection (see section V) in that it starts with an empty solution and adds the *most* correlated variable to the response $\boldsymbol{Y}$ – for arguments' sake, call that variable $X_1$. However, it is unlike forward selection in the way it chooses the *coefficient* for this variable

  - Forward variable selection would choose the coefficient in the most obvious way, using ordinary least squares. This would result in an estimate $\hat{\boldsymbol{Y}}$, with the property that the residuals $\boldsymbol{Y} - \hat{\boldsymbol{Y}}$ are totally uncorrelated with the variable $X_1$.

    This is because ordinary least squares works by projecting the vector $\boldsymbol{Y}$ onto the space spanned by $X_1$ – the resulting residual $\boldsymbol{Y} - \hat{\boldsymbol{Y}}$ is therefore necessarily perpendicular to $X_1$.

  - Least angle regression, on the other hand, takes a more 'democratic' approach. It produces an estimate $\hat{\boldsymbol{Y}}^{\text{LA}}$ by adding as much of the variable $X_1$ as needed to ensure that the correlation of $\hat{\boldsymbol{Y}}^{\text{LA}}$ with $X_1$ is equal to its correlation with the next-most-significant variable (say $X_2$). At which point it stops, and repeats the process for the next most correlated variable (say $X_3$).

- [Zhao and Yu 2006] showed that the LASSO only performs consistent variable selection (finds the correct model with probability 1 as $n$ tends to infinity) if the model satisfies a so-called *irrepresentable condition.* Some methods have been devised to deal with the situation in which this condition is not met. We discuss one of them, the randomised LASSO, in section XII.

### G. Bridge regression in its own right

Bridge Regression can also be used in its own right, with the methods of section IX used to select *both* $\lambda$ and $\gamma > 1$. This was, indeed, the context in which [Frank and Friedman 1993] originally introduced the method. [Fu 1998] analyses the performance of the Bridge and notes that it does sometimes outperform the LASSO (which makes sense, given that the LASSO is a subset of the Bridge), but that the method is frought with difficulty, because the non-linearity of the Bridge estimates makes generalised-cross validation an inappropriate tool for the selection of $\gamma$ and $\lambda$.

## VIII. SCAD

As discussed in sections VII A and VI D, neither classical model selection nor any of the Bridge penalties satisfy *all* three conditions in section VI. As a result, [Fan and Li 2001] suggest the Smoothly Clipped Absolute Deviation Penalty (SCAD), which does fulfil all three conditions. SCAD is defined by its derivative

$$p_\lambda'(|\beta|) = \lambda \left\{ \mathbb{1}_{|\beta|<\tilde{\lambda}} + \frac{(a\tilde{\lambda} - |\beta|)_+}{(a-1)\tilde{\lambda}} \mathbb{1}_{|\beta|>\tilde{\lambda}} \right\} \tag{25}$$

Even in the orthonormal design case, the solution is somewhat difficult to find. Details are provided in Appendix E, and the result is

$$\hat{\beta}^{\text{SCAD}} = \begin{cases} \text{sgn}(\hat{\beta}^{\text{OLS}}) \left( |\hat{\beta}^{\text{OLS}}| - \tilde{\lambda} \right)_+ & |\hat{\beta}^{\text{OLS}}| \leq 2\tilde{\lambda} \\ \frac{(a-1)\hat{\beta}^{\text{OLS}} - \text{sgn}(\hat{\beta}^{\text{OLS}})a\tilde{\lambda}}{(a-2)} & 2\tilde{\lambda} < |\hat{\beta}^{\text{OLS}}| \leq a\tilde{\lambda} \\ \hat{\beta}^{\text{OLS}} & |\hat{\beta}^{\text{OLS}}| > a\tilde{\lambda} \end{cases}$$

The SCAD penalty and this result are plotted in figure 10, together with the Bayesian prior on $\beta$ implied by SCAD (see section VII C).

The diagram clearly illustrates how SCAD resolves the bias of the LASSO. The estimator shrinks $\boldsymbol{\beta}$ towards the mean for small values of $\boldsymbol{\beta}$, but then returns to the ordinary least squares estimate. In terms of the Bayesian framework, SCAD retains the 'sharpness' of the maximum observed in LASSO, but keeps the tails constant and therefore reduces bias for large values of $\beta$.

The only issue with SCAD is that it is not convex, and is therefore very difficult to optimize. [Zou and Li 2008] suggested the following local linear approxmation near the point $\beta_0$

$$\begin{aligned} p_\lambda(|\beta|) &= p_\lambda(|\beta_0|) + p_\lambda'(|\beta_0|)\left(|\beta| - |\beta_0|\right) \\ &= p_\lambda'(|\beta_0|)|\beta| + \text{Constants} \end{aligned}$$

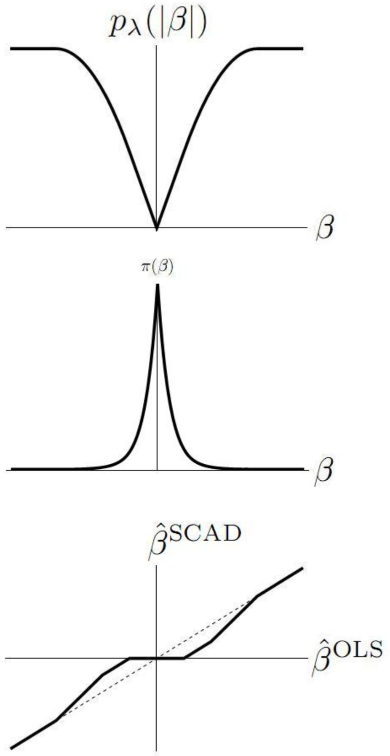

FIG. 10: Top: The SCAD penalty $p_\lambda$. Middle: The prior on $\beta$ implied by the SCAD penalty (see section VII C). Bottom: The SCAD estimator $\hat\beta^{\text{SCAD}}$. The dotted line is $\hat\beta^{\text{SCAD}} = \hat\beta^{\text{OLS}}$, drawn for reference.

We can then solve SCAD by choosing a sensible starting point $\beta_0$ and iterating as follows

$$\hat{\boldsymbol\beta}^{k+1} = \text{argmin}_{\boldsymbol\beta}\left(\frac{1}{N}\|\boldsymbol{Y}-\mathbf{X}\boldsymbol\beta\| + \sum_{j=1}^{p} p'_\lambda(|\hat\beta_j^k|)|\beta_j|\right)$$

Remarkably, provided the initial estimator is reasonably good (obtained using the LASSO, for example), it turns out that one step of the procedure is as good as the fully iterative procedure. Furthermore, under certain regularity conditions, this estimator can be shown to fulfil the oracle property.

[Fan and Li 2001] also propose a local *quadratic* approximation to SCAD.

A further complication comes from the fact that the SCAD penalty function contains *two* undetermined constants; $\lambda$ and $a$. Both could be estimated using the methods in section IX, but this is very computationally intensive. Instead, [Fan and Li 2001] suggest the following Bayesian argument to fix $a$

- Set $\lambda = \sqrt{2\log(p)}$. This is called *universal thresholding*, and was suggested by [Donoho and Johnstone 1994].
- Assume $\boldsymbol\beta$ has a normal prior with mean 0 and variance $a\lambda$.
- Computer the Bayes' Risk for each value of $a$ by numerical integration. The Bayes' Risk is simply

$$\mathbb{E}\left\{L(\boldsymbol\beta, \hat{\boldsymbol\beta})\right\}$$

- Carry out the simulation for a number of values of $p$.

In all cases, the minimum risk occurs at $a \approx 3.7$, and in simulations this value performs similarly to values of $a$ obtained by cross-validation (see section IX).

## IX. CHOOSING THE REGULARISATION PARAMETER $\lambda$

Every method we have considered so far has involved a regularisation parameter $\lambda$. This parameter regulates the severity of the penalty $p_\lambda$. $\lambda$ also indictes how far we are along the bias-variance spectrum. Small values of $\lambda$ reduce the penalty and therefore allow more complex models – this reduces the bias but increases the variance.

Let $\hat{\mathcal{M}}^\lambda$ represent the model obtained from a certain method, with a regularisation parameter $\lambda$, and let $\Lambda$ be the set of possible parameters. Our task is to choose one model (the 'best') out of the set of all possible models

$$\left\{\hat{\mathcal{M}}^\lambda; \lambda \in \Lambda\right\}$$

In this section, we consider methods to do this.

### A. Cross-valdiation & Generalised Cross-validation

By far the most common approach to the choice of $\lambda$ is cross-validation and generalised cross validation.

Cross validation was described in detail in section III A. Effectively, the set $\Lambda$ is filled with a number of discrete potential values for $\lambda$. For each of these values, we work out the cross-validation score, defined by

**Definition 12** (Cross-Validation Score)**.** We divide our training set $\mathcal{T}$ into $K$ equal segments or folds. We write $\hat Y^{-\kappa(i),\lambda}$ to represent the fitted value of $y$ when our model is fitted to $\mathcal{T}$ *minus* the segment containing data point $i$, with regularisation parameter $\lambda$. The cross-validation score is then

$$\text{CV}\left(\hat f\right) = \frac{1}{N}\sum_{i=1}^{N} L\left(Y_i, \hat Y^{-\kappa(i),\lambda}\right)$$

Finally, we choose the value of $\lambda \in \Lambda$ that produces the lowest cross-validation score.

In some cases, it is possible to calculate a *generalised cross-validation score*, which usually requires less computation. For example, for the specific case of ridge regression, [Golub Heath and Wahba 1979] show that minimising the following generalised-cross validation score (much easier to evaluate than the cross-validation score above) works just as well as minimising the cross validation score.

$$\mathrm{GCV}\left(\hat{f}\right) = N \frac{\left\| \boldsymbol{Y} - \mathbf{H}^R \boldsymbol{Y} \right\|^2}{\left(1 - \mathbb{T}\mathrm{r}\left(\mathbf{H}^R\right)\right)^2}$$

Where $\mathbf{H}^R$ is the hat matrix for the ridge estimate, defined in equation 24.

### B. Stability selection

Cross-validation entails the selection of a single model out of the set

$$\left\{ \hat{\mathcal{M}}^\lambda ; \lambda \in \Lambda \right\}$$

This might not, however, be the best approach, because it is possible that the 'best' model is not even in this set.

Stability selection takes a different approach, somewhat reminiscent of the bootstrap (section III B). It repeatedly perturbs the data and looks for variables that occur in a large fraction of the resulting models produced. These are the variables that are then chosen to form part of our final model $\hat{\mathcal{M}}^{\text{stable}}$

We now state this process formally.

**Definition 13** (Selection probabilities)**.** Let $I$ be a random sample from our data of size $\lfloor n/2 \rfloor$ (where $\lfloor x \rfloor$ is the largest integer smaller than or equal to $x$) drawn without replacement. Let $\hat{\mathcal{M}}^\lambda(I)$ be the result of using a method with regularisation parameter $\lambda$ on the data $I$.

Now, consider any variable $k$ in our problem. We denote the probability of this variable being selected by

$$\hat{\Pi}_k^\lambda = \mathbb{P}\left( k \subseteq \hat{\mathcal{M}}^\lambda(I) \right)$$

The probability $\mathbb{P}$ is taken over *all* possible subsamples $I$.

We are now ready to define our stable model

**Definition 14** (Stable model)**.** Our stable model $\hat{\mathcal{M}}^{\text{stable}}$ is chosen as follows

$$\hat{\mathcal{M}}^{\text{stable}} = \left\{ k : \max_{\lambda \in \Lambda} \hat{\Pi}_k^\lambda \geq \pi \right\}$$

where $\pi$ is a cutoff value with $0 < \pi < 1$

In other words, we keep variables with high selection probability and discard those with low selection probability.

It remains to discuss how to choose $\pi$ and $\Lambda$. Before we do, however, it is worth noting that [Meinshausen and Büehlmann 2010] show empirically that the results vary little for sensible choices in a range of the cutoff – say $\pi \in (0.6, 0.9)$.

In choosing $\pi$, we are trying to reduce the *per-family error rate V*

**Definition 15** (Per-family error rate)**.** The *per-family error rate*, $V_2$ is defined as the number of 'noise' variables selected in $\hat{\mathcal{M}}^{\text{stable}}$ – ie: the number of variables selected that, in the real underlying model, do not affect $Y$.

We also make the following definition

**Definition 16** (Selection range of $\Lambda$)**.** The *selection range* of $\Lambda$, $q_\Lambda$, is the total number of variables that our method is *capable* of selecting from subsamples as our model takes every value in the set $\Lambda$.

$$q_\Lambda = \mathbb{E}\left( \left| \cup_{\lambda \in \Lambda} \hat{\mathcal{M}}^\lambda(I) \right| \right)$$

Depending on the method used for fitting, it should be relatively simple to relate $q_\Lambda$ to $\Lambda$. For example, for the LASSO, smaller $\lambda$ means less variables, so if $\lambda_{\min}$ is the smallest member of $\Lambda$, $q_\Lambda$ is simply the number of variables in $\hat{\mathcal{M}}^\lambda_{\min}$.

The following theorem then holds

**Theorem 7.** *Under certain assumptions (see [Meinshausen and Büehlmann 2010, p 7]), the per-family error rate is bounded as follows*

$$\mathbb{E}(V) \leq \frac{1}{2\pi - 1} \frac{q_\Lambda^2}{p} \tag{26}$$

*where p is the total number of variables in our model.*

*Proof.* See [Meinshausen and Büehlmann 2010, §6.2]. □

Unsurprisingly, the theorem predicts that as we increase our threshold $\pi$ and as we reduce the total number of variables our method could possible select $q_\Lambda$, the probability of choosing a noise variable decreases.

Our tactic is then to set one of $q_\Lambda$ and $\pi$, and then use equation 26 to set the other to achieve the desired expected per-family error rate. A typical approach would be to set $\pi = 0.9$, and then to set $q_\Lambda$ (and therefore $\Lambda$) accordingly.

Note also that in some cases, the particular method used for fitting is so computationally intensive that it becomes impractical to apply stability selection for a large number of values of $\lambda$. In such case, it is possible to choose a single value of $\lambda$ and set $\Lambda = \lambda$. [Meinshausen and Büehlmann 2010] empirically show this method to be very successful provided that this single value of $\lambda$ is chosen such that some overfitting occurs (ie: such that the model $\hat{\mathcal{M}}^\lambda$ is too large).

# Part III
# Improvements

There have been a very large number of improvements on the methods we have discussed in this paper, designed to cope with various non-standard cases. Due to space constraints, we will only consider three here – the elastic net, Sure Independence Screening and the randomized LASSO. Others, of relevance to the topics in this paper are the adaptive LASSO [Zou 2006] and the Dantzig selector [Candes and Tao 2007].

## X. DEALING WITH CORRELATED VARIABLES – THE ELASTIC NET

### A. Introduction

The elastic net was proposed by [Zou and Hastie 2005] to solve two problems

1. LASSO is never able to construct a model with more than $n$ variables, where $n$ is the number of data points available (see [Efron Hastie Johnstone and Tibshirani 2004]). If the dimension of the problem, $p$, is much greater than $n$ (as is, for example, the case in genetic analyses where thousands of genes are screened using less than ten microarray experiments), this can be problematic.

   The Ridge is able to select larger models, but does not perform variable selection (ie: is not sparse).

2. If a model contains a number of correlated variables, the LASSO is most likely to pick any *one* of these correlated variables (we will examine this behaviour in more detail shortly). In many applications, this is desirable. For example, in ultra-high dimensional models, it is likely that every significant variable will have a number of 'noise' variables correlated to it (see section XI) – in that case, we prefer to only select the *most* significant variable.

   However, this can also be problematic in applications where we have reason to believe that correlated variables form a 'group'. This occurs, for example, in genetic studies where genes regulating a particular metabolic pathways are strongly correlated, and are *all* needed in the model if the metabolic pathway affects the response.

   The Ridge does not exhibit this behaviour – in fact, it tends to pick correlated variables *together* in a model. However the Ridge does not perform variable selection.

Our aim is to find a penalty that is both sparse, *and* groups correlated variables together.

Before we proceed, however, we spend some time considering the behaviour of penalised least squares with respect to correlated variables.

### B. Correlated variables

[Zou and Hastie 2005] and [Tibshirani 1996] use analytical solutions to examine the way LASSO and Ridge solution paths behave when faced with correlated variables. Their analysis is limited to the two-variable case. The LARS algorithm [Efron Hastie Johnstone and Tibshirani 2004] also provides some intuition as to why the LASSO only picks one variable out of a group of correlated variables. Indeed – once the 'best' variable in the group has been added to a LARS path, the residuals are unlikely to be correlated to the remaining variables in the correlated group.

We choose to provide a novel geometrical explanation of this behaviour, in terms of the LASSO penalisation function itself. Consider the penalised least squares problem

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \sum_{i=1}^{p} \lambda p(|\beta_i|)$$

The theory of convex optimisation (see, for example, [Boyd and Vandenberghe 2004]) imply that this problem is identical to the optimisation problem

$$\hat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta}} \|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 \text{ subject to } \sum_{i=1}^{p} p(|\beta_i|) \leq t$$

for some $t$. The $\lambda$ in the first formulation can be viewed as a Lagrange multiplier.

This optimisation problem effectively tries to get as close as possible to the ordinary least squares estimate $\hat{\boldsymbol{\beta}}^{\text{OLS}}$ while staying within the constraint. Thus, the solution will lie at the intersection of the contours of $\|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\|^2$ and the contour corresponding to $\sum_{i=1}^{p} p(|\beta_i|) = t$. Let us consider the form of these contours

- Contours of constant $\|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\|^2$ are clearly ellipses, and the minimum (ie: the centre of the ellipse) is clearly at the ordinary least squares estimate of $\boldsymbol{\beta}$

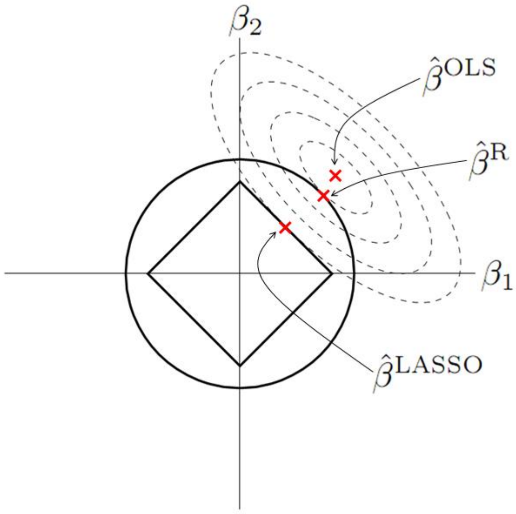

FIG. 11: Penalised least squares as a constrainted optimisation problem. The dotted lines are the contours of $\|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\|^2$ (centred at the ordinary least squares estimate), and the solid lines are the contours over which the Ridge and LASSO penalty functions are equal to a fixed constant $t$. The solutions occurs at the intersection of these contours.

- Contours of constant $\sum_{i=1}^{p} p(|\beta_i|) \leq t$ depend on the type of penalty used. For the Ridge penalty ($p(|\beta|) = |\beta|^2$), the contours are circles centred at the origin, and for the LASSO penalty ($p(|\beta|) = |\beta|$), the contours are diamonds, centred at the origin.

These concepts are illustrated in figure 11.

We now consider the geometry of the $\|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\|^2$ contours in more detail. First, notice that

$$\|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 = \boldsymbol{Y}^T\boldsymbol{Y} - 2\boldsymbol{Y}^T\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}$$

Notice, however, that $\mathbf{X}^T\mathbf{X}$ is simply the covariance matrix of $\mathbf{X}$. The geometry ellipses implies that the semi-major axes of the elliptical contours are inversely related to the *eigenvalues* of this matrix (see Appendix F).

Let us consider the two-variable case. Assuming the normalisation conditions in definition 5 are met, the covariance matrix is given by

$$\mathbf{X}^T\mathbf{X} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

The eigenvalues of this matrix are

$$1 + \rho \text{ and } 1 - \rho$$

And the semi-axes of the ellipse are therefore proportional to

$$\frac{1}{1+\rho} \text{ and } \frac{1}{1-\rho}$$

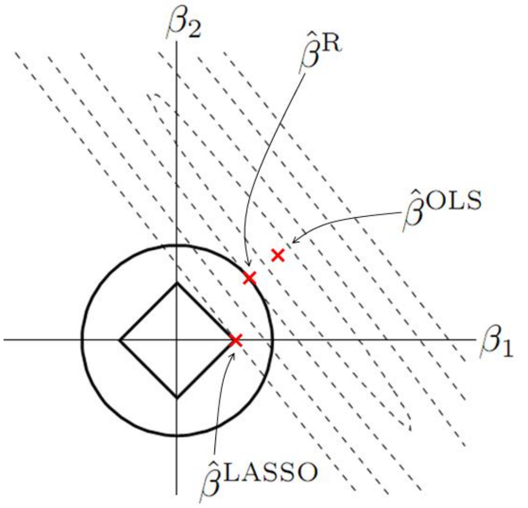

FIG. 12: The LASSO and the Ridge for two highly correlated variables ($\rho = -0.99$). The dotted lines are the elongated contours of $\|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\|^2$. The Ridge bulges outwards, and picks a solution involving both variables. The LASSO does not, and as a result only picks one of the two variables (in this case, $\beta_1$). [Note that this diagram is slightly misleading in that the design matrix $\mathbf{X}$ used to produce it does *not* satisfy the normalisation conditions in definition 5. See Appendix G for a discussion of why this was necessary.]

This means that as the variables become more correlated (ie: as $\rho \to 1$), the ellipse becomes more and more elongated.

We are now able to understand the behaviour of the LASSO and the Ridge when presented with a very long and thin ellipse, due to correlated variables:

- The Ridge (or indeed, any other strictly convex penalty) 'bulges out' to 'meet' the ellipse – thus, the solution occurs far from the axes and *both* variables are included.

- The LASSO (or any other concave penalty) does not 'bulge out' to meet the ellipse.

  In the case of non-correlated variables, this does not make a difference, because the ellipse itself will 'bulge out', and the LASSO is likely to pick both variables (like it does, for example, in figure 11).

  If the variables are correlated, however, the ellipse will also be very elongated, and depending on the inclination of the ellipse, the LASSO will choose one or the other of the correlated variables.

This phenomenom, in the case of high correlation, is illustrated in figure 12.

In summary, therefore, we see that a necessary condition for group selection of correlated variables is *strict* convexity of the penalty function.

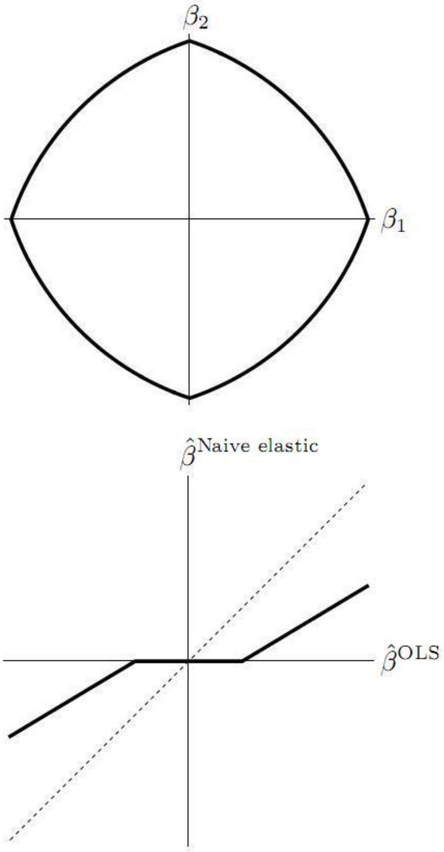
$\beta_2$
$\beta_1$
$\hat{\beta}^{\text{Naive elastic}}$
$\hat{\beta}^{\text{OLS}}$

FIG. 13: Top: Contours of constant naive elastic net penalty (in the two-variable case). Bottom: the effect of the naive elastic net in the orthonormal design case.

### C. The Solution

At first sight, it would appear that this problem can be solved by using Bridge regression with $1 < \gamma < 2$ – these functions being strictly convex. This, however, is not the case – as we saw in section VII A, Bridge regression is only sparse for $\gamma \leq 1$.

[Zou and Hastie 2005] proposed, instead, to use a linear combination of the LASSO and the Ridge

$$p_\lambda(|\beta|) = \lambda_1|\beta| + \lambda_2|\beta|^2 \tag{27}$$

They call this the *elastic net*.

The resulting solution is easily calculated from the results for the LASSO and the Ridge, and we omit the algebraic details. The form of the penalty function and of the elastic net estimator are shown in figure 13. Notice that the penalty function still has 'corners' at the origins, but is also strictly convex. This is what gives it both the properties we desire.

The diagram makes it clear that the elastic net is equivalent to performing Ridge shrinkage followed by LASSO thresholding. Unfortunately, this double-shrinkage produces in sub-optimal results, because it pushes us too far along the bias-variance tradeoff.

As a result, [Zou and Hastie 2005] call the method above the 'naive elastic net', and suggest a corrected estimator given by

$$\hat{\boldsymbol{\beta}}^{\text{Elastic}} = (1 + \lambda_2)\hat{\boldsymbol{\beta}}^{\text{Naive elastic}}$$

The factor of $1 + \lambda_2$ simply has the effect of 'undoing' some of the shrinkage, and is strongly motivated in a number of ways in [Zou and Hastie 2005].

## XI. DEALING WITH ULTRA-HIGH DIMENSIONAL PROBLEMS – SURE INDEPENDENCE SCREENING

In this section, we consider ultra-high dimensional problems – that is, problems in which the input vectors $\boldsymbol{X}$ are so large that even the methods we have discussed thus far are unable to fit the model. This can occur for a number of reasons. The two most obvious ones are

- Convex optimization algorithms usually take longer in very high dimensional spaces.
- When the number of variables grows extremely large, it becomes very probable that a number of variables will be correlated with each other (see [Fan and Lv 2008, p. 5] for a simulation demonstrating this). This makes it very difficult to know which variables *actually* affects $Y$, and which variables are just spuriously correlated to $Y$. This makes the model less *identifiable*.

For a more thorough discussion of the problems of high dimensionality, see [Fan and Lv 2008], [Donoho 2000] and [Fan and Li 2006].

Sure independence screening [Fan and Lv 2008] is a rapid and simple procedure for 'pruning' some of the variables in ultra-high dimensional problems. The resulting model is then small enough to allow the application of the methods we have discussed earlier in this paper.

### A. Basic SIS

The method used by sure indpendence screening is deceptively simple. It simply selects the variables that are most heavily correlated with the response $Y$. More formally, we let

$$\boldsymbol{\omega} = \mathbf{X}^T\boldsymbol{Y}$$

Since the columns of $\mathbf{X}$ and $\boldsymbol{Y}$ are scaled to ensure their mean is 0, $\boldsymbol{\omega}$ effectively contains the *correlation* of each variable with each variable. SIS then simply chooses variable $i$ if $\omega_i$ is among the highest of all components of $\omega$.

The number of variables we choose to retain in the model depends on our aims. A sensible suggestion is to retain $n - 1$ variables, to bring our model into the realm of more classical methods.

Despite its apparent simplicity, it can be shown that SIS possess the *sure screening property.* Let $\mathcal{M}_*$ be the 'real' underlying model, and let $\mathcal{M}_\gamma$ be the model chosen by SIS. Then

$$\mathbb{P}\left(\mathcal{M}_* \subset \mathcal{M}_\gamma\right) \to 1 \text{ as } n \to \infty \tag{28}$$

The proof is very involved – see [Fan and Lv 2008] for details.

### B. Iterated SIS (ISIS)

SIS may break down if[9]

- A variable is *jointly correlated* with the $Y$ (ie: when considered in conjunction with other variables, it does affect the response) but marginally uncorrelated (ie: by itself, it doesn't affect the response much). In that case, SIS would not rank the variable highly enough, and simply discard it.

  As an example, consider a true underlying model involving $J+1$ variables $X_1 \cdots X_{j+1}$, such that $\mathbb{C}\text{ov}(X_i, X_j) = \rho$. Imagine our true model is

  $$Y = X_1 + \cdots + X_J - J\rho X_{j+1}$$

  Clearly, if $J$ is large, $X_{j+1}$ heavily affects $Y$. However

  $$\begin{aligned}\mathbb{C}\text{ov}(X_{j+1}, y) = &\sum_{i=1}^{J} \mathbb{C}\text{ov}(X_{j+1}, X_i) \\ &- \mathbb{C}\text{ov}(X_{j+1}, J\rho X_{j+1}) = 0\end{aligned}$$

  SIS would therefore rank this important variable *last* in the list of potential variables!

- A 'spurious variable' is correlated to $Y$ only by virtue of its correlation to other variables which are genuinly correlated to $Y$. These 'spurious variables' may be ranked higher than other, genuine variables.

  As an example, consider a true underlying model involving three variables, $X_0$, $X_1$ and $X_2$, such that $X_0$ is uncorrelated to the other two. Imagine our true model is

  $$Y = \rho X_0 + X_1 + X_2$$

  Now imagine a third variable, $X_3$, is correlated to $X_1$ and $X_2$ such that $\mathbb{C}\text{ov}(X_3, X_1) = \mathbb{C}\text{ov}(X_3, X_2) = \rho$, but uncorrelated to $X_0$. We then have

  $$\mathbb{C}\text{ov}(X_0, Y) = \rho$$

  $$\mathbb{C}\text{ov}(X_3, Y) = \mathbb{C}\text{ov}(X_3, X_1 + X_2) = 2\rho$$

  $X_3$, the spurious correlated, is more strongly correlated to $Y$ than $X_0$. Thus, SIS would rank $X_3$ higher than $X_0$.

Iterated SIS (ISIS) is a method that seeks to overcome these difficulties. It was first proposed in [Fan and Lv 2008], and an improved version was proposed in [Fan Samworth and Wu 2009]. We give an account of the latter version.

1. Begin with an ultrahigh dimensional problem containing $p$ variables.
2. Using SIS, select $k_1$ of these variables. Let $\mathcal{A}_1$ be the resulting set of variables.
3. Use the LASSO (or any other similar method) to select a subset of these variables. We let
   - $\mathcal{M}_1$ be the resulting set of variables
   - $\hat{\boldsymbol{\beta}}_{\mathcal{M}_1}$ be the corresponding vector of fitted coefficients
   - $\boldsymbol{x}_{i,\mathcal{M}_1}$ be the sub-vector of $\boldsymbol{x}_i$ containing only those elements in $\mathcal{M}_1$.
4. Look at each variable $j$ that was *not* selected in $\mathcal{M}_1$, and calculate the following quantity[10]

   $$L_j^{(2)} = \min_{\boldsymbol{\beta}_{\mathcal{M}_1}, \beta_j} \frac{1}{n} \sum_{i=1}^{n} L\left(Y_i, \boldsymbol{x}_{i,\mathcal{M}_1}^T \boldsymbol{\beta}_{\mathcal{M}_1} + X_{ij}\beta_j\right)$$

   This quantity looks somewhat daunting, but is in fact quite simple. It does the following
   - Consider a model containing all the variables in $\mathcal{M}_1$ as well as variable $j$.
   - Find the minimum $\overline{\text{err}}$ for that model

   Effectively, it asks "if I were to add variable $j$ to my model as well as those variables in $\mathcal{M}_1$, how low would I be able to get $\overline{\text{err}}$?"
5. Select those variables $j$ with the *least* value of $L_j^{(2)}$, and put then into a set $\mathcal{A}_2$.
6. Now consider the set $\mathcal{M}_1 \cup \mathcal{A}_2$ and use the LASSO (or any other similar method) to select a subset of these variables. Call the resulting set of variables $\mathcal{M}_2$.
7. Return to step 4 with $\mathcal{M}_2$ instead of $\mathcal{M}_1$.

[9] The examples given here were inspired by similar examples given in [Fan and Lv 2010, pp. 127-128], a review paper on high dimensional variable selection.

[10] This is where the ISIS algorithm in [Fan and Lv 2008] differs from that in [Fan Samworth and Wu 2009]. The former advocates using the residuals from the previous step of fitting instead of $L_j^{(2)}$

The process can be repeated *either* until $\mathcal{M}_\ell = \mathcal{M}_{\ell-1}$ or until we have reached a set containing the prescribed number of variables $d$. [Fan Samworth and Wu 2009] chose $k_1 = \lfloor 2d/3 \rfloor$ (where $\lfloor x \rfloor$ is the largest integer smaller than or equal to $x$) and $k_r = d - \|\mathcal{M}_{r-1}\|$ thereafter, to ensure that ISIS takes at least two iterations to terminate.

How does ISIS deal with the shortcomings of SIS mentioned above?

- Even if a variable is not very highly ranked in the first stage of ISIS, it is very likely to be selected at a later stage if it is indeed jointly correlated with $Y$.
- Even if a variable is highly correlated with $Y$, it will not be selected unless it *also* significantly improves the prediction accuracy of the model. Thus, in the example above, once $X_1$ and $X_2$ are selected, $X_3$ is unlikely to also be selected, because it won't significantly improve the model. This allows $X_0$ to be selected.

These improvements are borne out by empirical studies in the aforementioned papers.

### C. Variants on ISIS

A number of attempts exist to further improve the performance of ISIS. We very briefly consider two of them here

- [Fan and Lv 2008] suggest transformation of variables as a way of dealing with correlation. For example, weights $w_1$, $w_2$ and $w_3$ at 2, 9 and 18 years are clearly positvely correlated. Considering, instead, the variables $w_1$, $w_2 - w_1$ and $w_3 - w_2$ can significantly weaken the correlation. This is an example of a *subject related transformation.*
- [Fan Samworth and Wu 2009] suggest the following method
  - Partition the $n$ data points into two halves at random.
  - Perform SIS or ISIS separately to the data in each partition. This will gives two sets of variables $\mathcal{A}_1$ and $\mathcal{A}_2$.
  - Both of these sets will satisfy the sure-screening property (equation 28) and will therefore contain many variables that are *truly* in the model. They will also, however, contain many variables that are *not* in the underlying model (ie: the false discovery rate (FDR) for these sets will be high).
  - Construct the set
$$\mathcal{A} = \mathcal{A}_1 \cap \mathcal{A}_2$$
    This set will also satisfy the sure screening property, but will also contain many less 'false positives' (this statement is formalised in the paper).

## XII. DEALING WITH FAILURES IN REGULARITY CONDITIONS – RANDOMIZED LASSO

We mentioned above that [Zhao and Yu 2006] showed that the LASSO only performs consistent variable selection if the model satisfies a so-called *irrepresentable condition.* [Zou 2006] proposed a solution to this problem in the form of the adaptive LASSO, a two-stage procedure. We consider an alternative algortihm here, due to [Meinshausen and Büehlmann 2010] – the *randomised LASSO.* Despite its simplicity, it is consistent for variable selection even if the irrepresentable condition is violated.

The basic idea of the randomised LASSO is simply to change the regularisation parameter $\lambda$ for every component of $\beta$.

**Definition 17** (Randomised LASSO)**.** Choose $\alpha \in (0,1]$ (the *weakness* of our algorithm), and let $W_k$ be independent identically distributed variables in $[\alpha, 1]$, for $k = 1, \cdots, p$. The randomised LASSO estimator is then

$$\hat{\boldsymbol{\beta}}^{\text{LASSO},\lambda,W} = \text{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^p} \|\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{k=1}^{p} \frac{|\beta_k|}{W_k}$$

Of course, it is nonsensical to hope that this random perturbation will uniformly lead to an improvement – and indeed, if applied once, the randomised LASSO is not very useful. However, applying the randomised LASSO many times and looking for variables that are often chosen turns out to be a very powerful procedure.

# Part IV

# Conclusions

This paper has considered a number of methods of dealing with statistical learning problems that lie in very high dimensions. Before we conclude, we list these methods, and their salient points

**Bootstrapping and cross-validation** are very powerful methods, especially when large amounts of data are available.

**Classical model selection** is arguably the most 'accurate' method, but is unfortunately extremely slow for more than about 30 variables.

**Ridge regression** shrinks the contribution of all variables to a certain extent, but no contributions to 0. It is well suited for models in which we have reason to believe every variable affects the response in a small way.

**The LASSO** performs simultaneous fitting and model selection, by shrinking the coefficient of some variables to 0. Unfortunately, it also introduces bias into larger coefficients. It is well suited for models in which we have reason to believe most variables have no effect on the response.

**SCAD** works like the LASSO, but without the bias issue of the LASSO. It is, however, a non-convex linear program, and therefore entails some implementation difficulties.

**Elastic net regression** is useful in situations in which we have reason to believe many variables do not affect the response, and those that do are part of a number of correlated groups.

**Randomised LASSO regression** is useful in situations in which regularity conditions on **X** are broken.

## XIII. CONCLUDING REMARKS

This paper has reviewed the various methods currently available for high dimensional variable selection. The story, however, is far from complete, and the field is still very much at the forefront of research.

New inovative techniques continue to be proposed – the Dantzig selector [Candes and Tao 2007], the adaptive LASSO [Zou 2006] and the minimum concavity penalty (MCP) [Zhang 2007] are only a few examples of recent methods which we did not considered in this paper. We consider the elastic net as an attempt to combine a number of existing method together, and it is one of several such methods.

Another area of interest, which we did not consider in this paper, is a critical assessment of the statistical properties of these techniques under various conditions. We briefly mentioned the sure screening property in the context of sure independence screening. A number of other such properties are used to asses the performance of variable selection techniques (the oracle property, for example [Fan and Li 2001]). There is also a tremendous amount of recent work on the *distribution* of penalised likelihood estimators, and associated confidence sets – see, for example, [Pötscher and Leeb 2009], [Pötscher and Schneider 2010] and [Pötscher and Schneider 2009]. In addition, the methods considered in this paper sometimes perform poorly for certain kinds of design matrices **X** – it is of interest to characterise the situations in which this happens, and to devise methods that avoid these problems.

Our focus in this paper has mostly been on the linear model. There are, however, a number of methods designed to deal with high dimensional variable selection beyond the linear model. See [Hastie Tibshirani and Friedman 2009] and references therein (and, for example, [Fan Samworth and Wu 2009]). This is also an active area of current research.

Finally, opportunities abound to design robust and user-friendly algorithms and software to quickly and easily implement the theoretical ideas we have considered in this paper.

## XIV. ACKNOWLEDGEMENT

I am very grateful to Dr. Richard Samworth for introducing me to the wonders of statistics in his 'Statistical Theory' course, taught as part of the Certificate of Advanced Study in Mathematics.

My thanks also to Claude Schneider, for rescuing this essay from the clutches of digital oblivion when my laptop expired halfway through my writing.

# Part V
# Appendices

### Appendix A: Expectations

**Definition 18** (Expectations)**.** In this paper, we use several different kinds of expectations

- $\mathbb{E}_{(x,y)}$ represents taking an average over *every* possible pair of input and output that could possibly arise.
- $\mathbb{E}_{(X,Y)\in\mathcal{T}}$ represents taking an average over those pairs of $(X, Y)$ in our training set $\mathcal{T}$ only.
- $\mathbb{E}_{\mathcal{T}}$ represents taking an average over all possible training sets.
- $\mathbb{E}_{Y^{\text{NEW}}}$ is used in the definition of the in-sample error (definition 19). There, $Y^{NEW}$ represents a *new* observation of $Y$ for a *given* input $\boldsymbol{X}$. This expectation represents taking an average over all such new instances.

### Appendix B: Expected Optimisim

In this appendix, we prove theorem 2.

We will be using a slightly different version of the expected generalisation error, called the *in-sample error*

**Definition 19** (In-sample error).

$$\begin{aligned}\text{Err}_{\text{in}} &= \mathbb{E}_{y^{\text{NEW}}}\mathbb{E}_{(X,Y)\in\mathcal{T}}\left\{L\left(y^{\text{NEW}}, \hat{f}(\boldsymbol{X})\right)\right\} \\ &= \frac{1}{N}\sum_{i=1}^{N} L\left(\overline{Y}_i, \hat{Y}_i\right)\end{aligned}$$

where $Y^{\text{NEW}}$ represents a *new* observation at the *same* point $\boldsymbol{X}_i$, $\overline{Y}_i = \mathbb{E}\{f(\boldsymbol{X}_i)\}$ represents the *average* of all such observations, and $\hat{Y}_i = \hat{f}(\boldsymbol{X})$, as usual.

The similarity to the expected generalisation error (definition 2) is obvious. Both consider *new* data points outside the training set – the only difference is that the in-sample error constrains these new data points to be at the *same* $\boldsymbol{X}$ coordinates as those in the training set.

Our new definition of the expected optimism is then

**Definition 20** (Expected Optimism (reviewed)).

$$\omega = \mathbb{E}_{\mathcal{T}}\{\text{Err}_{\text{in}} - \overline{\text{err}}\}$$

We first develop a lemma necessary for the proof of our theorem

**Lemma 1.** *Let $\ell$ be a concave function, and consider a class of loss functions defined as follows*

$$L(y_i, \hat{y}_i) = \ell(\hat{y}_i) + \dot{\ell}(\hat{y}_i)(y_i - \hat{y}_i) - \ell(y) \tag{B1}$$

*For loss functions in this class,*

$$\omega = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\mathcal{T}}\left\{\hat{\zeta}_i(y_i - \overline{y}_i)\right\}$$

*Where $\hat{\boldsymbol{\zeta}}$ is the vector with ith component*

$$\hat{\zeta}_i \equiv -\dot{\ell}(\hat{y}_i)$$

*$\omega$ is the expected optimism (definition 20), $N$ is the number of items in the training set $\mathcal{T}$, $y_i$ is the ith output in the training set, $\overline{y}_i$ is the average of all outputs $Y_i$ for input $\boldsymbol{X}_i$ and $\hat{y}_i = \hat{f}(\boldsymbol{X}_i)$, our model's prediction of what $y_i$ should be.*

*Proof.* [11] For loss functions satisfying B1

$$\begin{aligned}&L(y_i^{NEW}, \hat{y}_i) - L(y_i, \hat{y}_i) \\ &\quad= -\dot{\ell}(\hat{y})\left(y_i - y_i^{NEW}\right) + \ell(y_i) - \ell(y_i^{NEW}) \\ &\quad= \hat{\zeta}_i\left(y_i - y_i^{NEW}\right) + \ell(y_i) - \ell(y_i^{NEW})\end{aligned}$$

Therefore

$$\begin{aligned}&\text{Err}_{\text{in}} - \overline{\text{err}} \\ &\quad= \frac{1}{N}\sum_{i=1}^{N}\hat{\zeta}_i(y_i - \overline{y}_i) + \ell(y_i) - \ell(y_i^{NEW})\end{aligned} \tag{B2}$$

Finally, remember that $\omega = \mathbb{E}_{\mathcal{T}}\{\text{Err}_{\text{in}} - \overline{\text{err}}\}$. When taking this expectation with respect to all training sets, the last two terms in equation B2 vanish, because $\mathbb{E}(y_i^{NEW}) = y_i$, and so $\mathbb{E}\left\{\ell(y_i^{NEW})\right\} = \ell(y_i)$ . As such

$$\omega = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\mathcal{T}}\left\{\hat{\zeta}_i(y_i - \overline{y}_i)\right\}$$

Where, in the last step, we used linearity of expectations to swap the expectation and the sum. □

**Theorem 8.** *For squared error loss (definition 1)*

$$\omega = \frac{2}{N}\sum_{i=1}^{N}\mathbb{C}\text{ov}\left(\hat{Y}_i, Y_i\right)$$

*where $\omega$ is the expected optimism (definition 20), $N$ is the number of items in the training set $\mathcal{T}$, $Y_i$ is the ith output in the training set and $\hat{Y}_i = \hat{f}(\boldsymbol{X}_i)$, our model's prediction of what $Y_i$ should be.*

*Proof.* [12] Putting

$$\ell(x) = x(1 - x)$$

in equation B1 confirms that squared error loss does indeed belong to that class.

By Lemma 1, we therefore have that

$$\omega = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\mathcal{T}}\left\{\hat{\zeta}_i(y_i - \overline{y}_i)\right\}$$

For squared error loss, we also have that

$$\hat{\zeta}_i \equiv 2\hat{y}_i - 1$$

and so

$$\begin{aligned}\omega &= \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\mathcal{T}}\{(2\hat{y}_i - 1)(y_i - \overline{y}_i)\} \\ &= \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\mathcal{T}}\left\{\left(2\hat{y}_i - 2\overline{\hat{y}}_i + 2\overline{\hat{y}}_i - 1\right)(y_i - \overline{y}_i)\right\} \\ &= \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\mathcal{T}}\left\{\left(2\hat{y}_i - 2\overline{\hat{y}}_i\right)(y_i - \overline{y}_i)\right\} \\ &\qquad + \frac{1}{N}\sum_{i=1}^{N}\left(2\overline{\hat{y}}_i - 1\right)\mathbb{E}_{\mathcal{T}}\{y_i - \overline{y}_i\}\end{aligned}$$

[11] The proof of this Lemma is based on a similar theorem for binary outputs in [Efron 1986].

[12] The proof of this Theorem is based on a similar theorem for binary outputs in [Efron 1986].

where $\overline{y}_i = \mathbb{E}(\hat{f}(\boldsymbol{X}_i))$ is a constant (and was therefore taken out of the expectation in the last step). Finally, note that $\mathbb{E}_{\mathcal{T}}(y_i) = \overline{y}_i$, and so the last term vanishes. Thus

$$\begin{aligned}\omega &= \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\mathcal{T}}\left\{\left(2\hat{y}_i - 2\overline{y}_i\right)\left(y_i - \overline{y}_i\right)\right\} \\ &= \frac{2}{N}\sum_{i=1}^{N}\mathbb{C}\text{ov}\left(\hat{Y}_i, Y_i\right)\end{aligned}$$

As required. □

Finally, we derive the form of the optimism in the particular case of the linear model.

**Theorem 9.** *For a prediction method satisfying $\hat{\boldsymbol{Y}} = \mathbf{H}\boldsymbol{Y}$, the expected optimism is given by*

$$\omega = \frac{2}{N}\mathbb{T}\text{r}(\mathbf{H})\sigma_\epsilon^2$$

*Proof.* Recall that in the linear model, $\hat{\boldsymbol{Y}} = \mathbf{H}\boldsymbol{Y}$, where $\mathbf{H}$ is the hat matrix (definition 6). Let $\boldsymbol{H}_i$ represent the $i$th row of the matrix $H$. Then

$$\begin{aligned}\hat{Y}_i &= \boldsymbol{H}_i\boldsymbol{Y} \\ &= \sum_{k=1}^{N} H_{ik}Y_k\end{aligned}$$

Therefore

$$\begin{aligned}\omega &= \frac{2}{N}\sum_{i=1}^{N}\mathbb{C}\text{ov}\left(\hat{Y}_i, Y_i\right) \\ &= \frac{2}{N}\sum_{i=1}^{N}\mathbb{C}\text{ov}\left(\sum_{k=1}^{N} H_{ik}Y_k, Y_i\right)\end{aligned}$$

The $Y_i$ are all independent, so only one covariance remains

$$\begin{aligned}&= \frac{2}{N}\sum_{i=1}^{N}\mathbb{C}\text{ov}\left(H_{ii}Y_i, Y_i\right) \\ &= \frac{2}{N}\sum_{i=1}^{N} H_{ii}\mathbb{V}\text{ar}(Y_i) \\ &= \frac{2}{N}\mathbb{T}\text{r}(\mathbf{H})\sigma_\epsilon^2 \qquad \text{(B3)}\end{aligned}$$

As required. □

**Theorem 10.** *If a linear model is fit using ordinary least squares (with $\hat{f}(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{\beta}^{OLS}$)*

$$\omega = \frac{2p}{N}\sigma_\epsilon^2$$

*where $p$ is the number of covariates in $\boldsymbol{X}$ and $\sigma_\epsilon^2 = \mathbb{V}\text{ar}(Y_i)$, the irreducible error in the underlying model.*

*Proof.* We first show that in this particular case, $\mathbb{T}\text{r}(\mathbf{H}) = p$. We simply note that $\mathbf{H} = \boldsymbol{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$ and that $\mathbb{T}\text{r}(\mathbf{ABC}) = \mathbb{T}\text{r}(\mathbf{CAB})$. As such

$$\begin{aligned}\mathbb{T}\text{r}(\mathbf{H}) &= \mathbb{T}\text{r}\left\{\boldsymbol{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\right\} \\ &= \mathbb{T}\text{r}\left\{\boldsymbol{X}^T\boldsymbol{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\right\}\end{aligned}$$

$\boldsymbol{X}^T\boldsymbol{X}$ is a square matrix, with as many columns as there are columns in $\boldsymbol{X}$. Therefore

$$\begin{aligned}&= \mathbb{T}\text{r}(I_p) \\ &= p \qquad \text{(B4)}\end{aligned}$$

Combining equations B3 and B4, we obtain

$$\omega = \frac{2p}{N}\sigma_\epsilon^2$$

As required.

□

## Appendix C: Expanding the expectation of the loss in model selection

*Proof.* In this appendix, we prove theorem 1

$$\begin{aligned}\mathbb{E}_{(x,y)}\left\{L(y,\hat{f}(\boldsymbol{x}))\right\} &= \mathbb{E}\left\{\left(y - \hat{f}(\boldsymbol{x})\right)^2\right\} \\ &= \mathbb{E}\left\{\left(f(\boldsymbol{x}) + \epsilon - \hat{f}(\boldsymbol{x})\right)^2\right\} \\ &= \mathbb{E}\left\{\epsilon^2 + \left(f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})\right)^2 \right. \\ &\qquad \left. + 2\epsilon\left(f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})\right)\right\}\end{aligned}$$

Note, however, that $\mathbb{E}(\epsilon) = 0$ and that $\epsilon$ is independent of $\boldsymbol{X}$, so that the last term in the expectation vanishes. Further note that $\mathbb{E}(\epsilon^2) = \mathbb{V}\text{ar}(\epsilon)$. Therefore:

$$\begin{aligned}\mathbb{E}\left\{L(y,\hat{f}(\boldsymbol{x}))\right\} &= \mathbb{V}\text{ar}(\epsilon) + \mathbb{E}\left\{\left(f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})\right)^2\right\} \\ &= \mathbb{V}\text{ar}(\epsilon) + \mathbb{V}\text{ar}\left(\hat{f}(\boldsymbol{X})\right) \\ &\qquad + \left[\mathbb{E}\left(\hat{f}(\boldsymbol{X})\right) - f(\boldsymbol{X})\right]^2\end{aligned}$$

As required. □

## Appendix D: Orthonormal Design

**Theorem 11.** *In the orthonormal design case, in which $\mathbf{X}^T\mathbf{X} = \mathbf{I}$, equation 3 can be written as*

$$Q(\boldsymbol{\beta}) = \frac{1}{N}\left\|\boldsymbol{Y} - \boldsymbol{Y}^{\hat{O}LS}\right\|^2 + \left\|\hat{\boldsymbol{\beta}}^{OLS} - \boldsymbol{\beta}\right\|^2$$

*where $\hat{\boldsymbol{\beta}}^{OLS}$ is the ordinary least squares estimate of $\boldsymbol{\beta}$ and $\hat{\boldsymbol{Y}}^{OLS}$ is the ordinary least squares estimate of $\boldsymbol{Y}$.*

*Proof.*

$$\begin{aligned}
\|\boldsymbol{Y}-\mathbf{X}\boldsymbol{\beta}\|^2 &= \boldsymbol{Y}^2 - 2\boldsymbol{\beta}^T\mathbf{X}^T\boldsymbol{Y} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} \\
&= \boldsymbol{Y}^2 - 2\boldsymbol{Y}^T\mathbf{X}\mathbf{X}^T\boldsymbol{Y} \\
&\quad + \boldsymbol{Y}^T\mathbf{X}\mathbf{X}^T\boldsymbol{Y} + \boldsymbol{Y}^T\mathbf{X}\mathbf{X}^T\boldsymbol{Y} \\
&\quad - 2\boldsymbol{\beta}^T\mathbf{X}^T\boldsymbol{Y} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} \\
&= \boldsymbol{Y}^2 - 2\boldsymbol{Y}^T\mathbf{X}\mathbf{X}^T\boldsymbol{Y} \\
&\quad + \boldsymbol{Y}^T\mathbf{X}\mathbf{X}^T\mathbf{X}\mathbf{X}^T\boldsymbol{Y} + \boldsymbol{Y}^T\mathbf{X}\mathbf{X}^T\boldsymbol{Y} \\
&\quad - 2\boldsymbol{\beta}^T\mathbf{X}^T\boldsymbol{Y} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} \\
&= \boldsymbol{Y}^T\left(\mathbf{I}-\mathbf{X}\mathbf{X}^T\right)^T\left(\mathbf{I}-\mathbf{X}\mathbf{X}^T\right)\boldsymbol{Y} \\
&\quad + \left(\mathbf{X}^T\boldsymbol{Y}-\boldsymbol{\beta}\right)^T\left(\mathbf{X}^T\boldsymbol{Y}-\boldsymbol{\beta}\right) \\
&= \left\|\boldsymbol{Y}-\hat{\boldsymbol{Y}}^{\text{OLS}}\right\| + \left\|\hat{\boldsymbol{\beta}}^{\text{OLS}}-\boldsymbol{\beta}\right\|
\end{aligned}$$

As required. □

## Appendix E: Subgradients

In this appendix, we find the value of $\beta$ at which the function

$$q(\beta) = \frac{1}{N}\left(\hat{\beta}^{\text{OLS}}-\beta\right)^2 + p'_\lambda(|\beta|)$$

attains its minimum, when the function $p_\lambda$ is nondifferentiable at the origin.

We will need the concept of a *sub-gradient*

**Definition 21** (Subgradient)**.** Let $f : \mathbb{R}^p \to (-\infty,\infty)$ be a convex function. We say that $\boldsymbol{x}^* \in \mathbb{R}^p$ is a *subgradient* of $f$ at $\boldsymbol{x}$ if

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{x}^*, \boldsymbol{y}-\boldsymbol{x}\rangle \quad \forall \boldsymbol{y} \in \mathbb{R}^p$$

Intuitively, $\boldsymbol{x}$ is the gradient of a plane that touches $f$ at $\boldsymbol{y}$, and lies below it everywhere else.

At points where $f$ is continuous, there is a single subgradient equal to the derivative. At other points, there is a set of subgradients, denoted

$$\partial f(\boldsymbol{x})$$

Note also that

$$\mathbf{0} \in \partial f(\boldsymbol{x}) \Leftrightarrow f(\boldsymbol{y}) \geq f(\boldsymbol{x}) \qquad \forall \boldsymbol{y} \in \mathbb{R}^p$$

In other words, $\boldsymbol{x}$ is a minimum of $f$ if and only if $\mathbf{0}$ is in the set of subgradients at that point.

We now use this concept to minimize $q(\beta)$. We will also find it useful to remember that

$$q'(\beta_j) = \text{sgn}(\beta_j)\left\{\Delta|\beta_j| + p'_\lambda\left(|\beta_j|\right)\right\} - \Delta\hat{\beta}_j^{\text{OLS}}$$

Where, as usual, $\Delta = 2/N$. We also let $\tilde{\lambda} = \lambda/\Delta$.

### 1. The LASSO

For the LASSO

$$q(\beta) = \frac{1}{N}\left(\hat{\beta}^{\text{OLS}}-\beta\right)^2 + \lambda|\beta|$$

Note that

$$\partial q(\beta) = \begin{cases} \Delta\beta - \Delta\hat{\beta}^{\text{OLS}} + \lambda & \beta > 0 \\ \{-\Delta\hat{\beta}^{\text{OLS}} + \theta\lambda : \theta \in [-1,1]\} & \beta = 0 \\ \Delta\beta - \Delta\hat{\beta}^{\text{OLS}} - \lambda & \beta < 0 \end{cases}$$

This function is minimised for those values of $\beta$ for which $0 \in \partial q(\beta)$. This immediately yields

$$\hat{\beta}^{\text{LASSO}} = \text{sgn}(\hat{\beta}^{\text{OLS}})\left(|\hat{\beta}^{\text{OLS}}| - \tilde{\lambda}\right)_+$$

Where $x_+ = \max(x, 0)$.

### 2. SCAD

The SCAD function is not convex, but it is piecewise convex near the origin. Since the origin is the only place at which there is some ambiguity as to the gradient of the function, we can use subgradients again.

The SCAD penalty is

$$p'_\lambda(|\beta|) = \lambda\left\{\mathbb{1}_{|\beta|<\tilde{\lambda}} + \frac{(a\tilde{\lambda}-|\beta|)_+}{(a-1)\tilde{\lambda}}\mathbb{1}_{|\beta|>\tilde{\lambda}}\right\}$$

As such

$$\partial q(\beta) = \begin{cases} \Delta\beta - \Delta\hat{\beta}^{\text{OLS}} & \beta > a\tilde{\lambda} \\ \Delta\beta - \Delta\hat{\beta}^{\text{OLS}} + \frac{a\lambda-\beta}{(a-1)\lambda} & \tilde{\lambda} < \beta < a\tilde{\lambda} \\ \Delta\beta - \Delta\hat{\beta}^{\text{OLS}} + \lambda & 0 < \beta < \tilde{\lambda} \\ \left\{-\Delta\hat{\beta}^{\text{OLS}} + \theta\lambda : \theta \in [0,1]\right\} & \beta = 0 \\ \Delta\beta - \Delta\hat{\beta}^{\text{OLS}} - \lambda & 0 < \beta < \tilde{\lambda} \\ \Delta\beta - \Delta\hat{\beta}^{\text{OLS}} - \frac{a\lambda+\beta}{(a-1)\lambda} & \tilde{\lambda} < \beta < a\tilde{\lambda} \\ \Delta\beta - \Delta\hat{\beta}^{\text{OLS}} & \beta > a\tilde{\lambda} \end{cases}$$

This function is minimised for those values of $\beta$ for which $0 \in \partial q(\beta)$. After some manipulation, this yields

$$\hat{\beta}^{\text{SCAD}} = \begin{cases} \text{sgn}(\hat{\beta}^{\text{OLS}})\left(|\hat{\beta}^{\text{OLS}}| - \tilde{\lambda}\right)_+ & |\hat{\beta}^{\text{OLS}}| \leq 2\tilde{\lambda} \\ \frac{(a-1)\hat{\beta}^{\text{OLS}} - \text{sgn}(\hat{\beta}^{\text{OLS}})a\tilde{\lambda}}{(a-2)} & 2\tilde{\lambda} < |\hat{\beta}^{\text{OLS}}| \leq a\tilde{\lambda} \\ \hat{\beta}^{\text{OLS}} & |\hat{\beta}^{\text{OLS}}| > a\tilde{\lambda} \end{cases}$$

### 3. Sparsity

[Tibshirani 1996] graphically shows that sparsity only occurs for penalty functions such as the LASSO and SCAD, which have ‘corners’ at the origin (ie: are non-differentiable).

We suggest an alternative motivation for this statement. The discussion above has made it clear that for sparsity to occur, 0 must be a member of the set of subgradients for more than one value of $\hat{\beta}$. This, however, can only occur if the set of subgradients contains more than one member when $\beta = 0$ This, in turn, only occurs when the penalty function is non-differentiable at $\beta = 0$ – because then, the subgradients take a range of values at $\beta = 0$ as the 'jump' from the 'pre-corner' gradient to the 'post-corner' gradient.

## Appendix F: The Geometry of Ellipses

The general form of an ellipse is given by

$$AX^2 + 2BXY + CY^2 + 2DX + 2FY + M = 0$$

This can be written in matrix form (which generalises to higher dimensions) as

$$\boldsymbol{X}^T \mathbf{A} \boldsymbol{X} + 2\boldsymbol{J}^T \boldsymbol{X} + M = 0$$

Where

$$\mathbf{A} = \begin{pmatrix} A & B \\ B & C \end{pmatrix} \qquad \boldsymbol{J} = \begin{pmatrix} D \\ F \end{pmatrix}$$

Our first step will be to diagonalise our matrix $\mathbf{A}$, and write it as

$$\mathbf{A} = \mathbf{P}\mathbf{D}^2\mathbf{P}^T$$

where $\mathbf{D}$ is a diagonal matrix and $\mathbf{P}^T\mathbf{P} = \mathbf{P}\mathbf{P}^T = \mathbf{I}$.

We then define a new coordinate system $\tilde{\boldsymbol{X}}$, such that $\tilde{\boldsymbol{X}} = \mathbf{P}^T\boldsymbol{X}$ (this is equivalent to rotating our coordinate axes to align them with the coordinate of the ellipse – ie: with the eigenvectors of $\mathbf{A}$). We can then re-write our ellipse as

$$\tilde{\boldsymbol{X}}^T \mathbf{D}^2 \tilde{\boldsymbol{X}} + 2\boldsymbol{J}^T\mathbf{P}\tilde{\boldsymbol{X}} + M = 0$$

Completing the square, this becomes

$$\left(\mathbf{D}\tilde{\boldsymbol{X}} + \mathbf{D}^{-1}\boldsymbol{J}^T\mathbf{P}\right)^2 - \mathbf{P}^T\boldsymbol{J}\mathbf{D}^{-2}\boldsymbol{J}^T\mathbf{P} + M = 0$$

Finally, we define a new coordinate system $\hat{\boldsymbol{X}}$ such that $\tilde{\boldsymbol{X}} = \hat{\boldsymbol{X}} - \mathbf{D}^{-2}\boldsymbol{J}^T\mathbf{P}$. Our ellipse then becomes

$$\left(\mathbf{D}\hat{\boldsymbol{X}}\right)^2 = \text{Constant}$$

Which makes it clear that the semi-axes of the ellipse are, indeed, equal to the reciprocal of the values on the diagonal of $\mathbf{D}$ – which are the eigenvalues of $\mathbf{A}$.

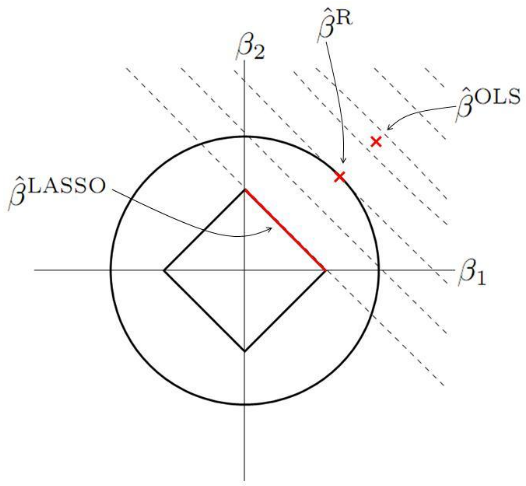

FIG. 14: Figure 12 for a situation in which the matrix $\mathbf{X}$ satisfies the normalisation conditions in defintiion 5. Clearly, there are a number of possible LASSO solutions, most of which involve *both* variables.

## Appendix G: Figure 12

This appendix discusses some subtelties associated with the diagram in figure 12. The caption of the figure cautions the reader that the design matrix $\mathbf{X}$ used to produce the diagram did *not* satisfy the normalisation conditions in definition 5.

To understand why this was necessary, recall that the normalisation conditions imply that, in the two variable case,

$$\mathbf{X}^T\mathbf{X} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

Recall also that the *eigenvectors* of that matrix lie along the axes of the ellipse (as discussion in section F).

The eigenvector of this particular matrix, however, are $(1\ 1)^T$ and $(1\ -1)^T$, and this implies that the ellipse in question lies at a $45^o$ angle to the axes. This means that the ellipse would have been *exactly* parallel to the LASSO penalty function. See figure 14 for the diagram that would have resulted.

Figure 14 makes it clear that in that case, there are a number of LASSO solutions, most involving *both* $\beta_1$ and $\beta_2$. This special case in two dimensions is therefore the exception to the rule that the lasso picks *one* of a group of correlated variables. To illustrate our point, we therefore preferred to use a non-normalised $\mathbf{X}$.

[Akaike 1974] H. Akaike, *A new look at the statistical model identification*, IEEE Transactions on Automatic Control, Vol.19, No.4 (Dec. 1974), pp. 716- 723

[Allen 1971] D.M. Allen, *The Relationship between Variable Selection and Data Agumentation and a Method for Prediction*, Technometrics, Vol. 16, No. 1 (Feb., 1974), pp. 125-127

[Boyd and Vandenberghe 2004] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, 2004)

[Breiman 1996] L. Breiman, *Heuristics of Instability and Stabilization in Model Selection*, The Annals of Statistics, Vol. 24, No. 6 (Dec., 1996), pp. 109-135.

[Breiman and Spector 1992] L. Breiman and P. Spector, *Submodel selection and evaluation in regression: the X-random case*, International Statistical Review, Vol. 60, No. 3 (Dec., 1992), pp. 291-319.

[Candes and Tao 2007] E. Candes and T. Tao, *The Dantzig selector: Statistical estimation when p is much larger than n*, The Annals of Statistics, Vol. 35, No. 6 (2007), pp. 2313-2351

[Cavanaugh 2009] J.E. Cavanaugh, Lecture notes on the Bayesian Information Criterion, Department of Statistics and Actuarial Science, The University of Iowa (September 2009).
`http://myweb.uiowa.edu/cavaaugh/ms_lec_6_ho.pdf`
Retrieved on April 10th 2009.

[Donoho and Johnstone 1994] D.L. Donoho and I.M. Johnstone, *Ideal Spatial Adaptation by Wavelet Shrinkage*, Biometrika, Vol. 81, pp. 425-455

[Donoho 2000] D.L. Donoho, *The curses and blessings of dimensionality*. Aide-Memoire of a Lecture at AMSConference on Math Challenges of the 21st Century.
`http://www-stat.stanford.edu/~donoho`
`/Lectures/AMS2000/Curses.pdf`
Retrieved on April 11th 2010.

[Efron 1979] B. Efron, *Bootstrap Methods: Another Look at the Jackknife*, The Annals of Statistics, Vol. 7, No. 1 (Jan., 1979), pp. 1-26

[Efron 1983] B. Efron, *Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation*, Journal of the American Statistical Association, Vol. 78, No. 382 (Jun., 1983).

[Efron 1986] B. Efron, *How biased is the apparent error rate of a prediction rule?*, Journal of the American Statistical Association, Vol. 81, No. 394 (Jun., 1986), pp. 461-470

[Efron and Tibshirani 1997] B. Efron and R. Tibshirani, *The .632+ Bootstrap Method*, Journal of the American Statistical Association, Vol. 92, No. 438 (Jun., 1997), pp. 548-560

[Efron Hastie Johnstone and Tibshirani 2004] B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, *Least Angle Regression*, The Annals of Statistics, Vol. 32, No. 2 (Apr., 2004), pp. 407-451

[Fan and Li 2001] J. Fan and R Li, *Variable Selection via Nonconcave Penalized Lieklihhod and its Oracle Properties*, Journal of the American Statistical Association, Vol. 96, No. 456 (Dec., 2001), pp. 1348-1360

[Fan and Li 2006] J. Fan and R. Li, *Statistical Challenges with High Dimensionality: Feature Selection in Knowledge Discovery*. arXiv:math/0602133v1 [math.ST]

[Fan and Lv 2008] J. Fan and J. Lv, *Sure Independence Screening for Ultra-High Dimensional Feature Space*, Journal of the Royal Statistical Society Series B (2008), 70, pp. 849-911

[Fan Samworth and Wu 2009] J. Fan, R. Samworth and Y. Wu, *Ultrahigh dimensional variable selection: beyond the linear model*, Journal of Machine Learning Research, Vol. 10 (2009), pp. 2013-2038

[Fan and Lv 2010] J. Fan and J. Lv, *A Selective Overview of Variable Selection in High Dimensional Feature Space* (Invited Review Article), Statistica Sinica 20 (2010), pp. 101-148

[Frank and Friedman 1993] I.E. Frank and J.H. Friedman, *A Statistical View of Some Chemometrics Regression Tools*, Technometrics, Vol. 35, No. 2 (May, 1993), pp. 109-135

[Fu 1998] W.J. Fu, *Penalized Regressions: The Bridge versus the Lasso*, Journal of Computational and Graphical Statistics, Vol. 7, No. 3 (Sep., 1998), pp. 397-416

[Furnival 1971] G.M. Furnival, *All Possible Regressions with Less Computation*, Technometrics, Vol. 13, No. 2 (May, 1971), pp. 403-408

[Furnival and Wilson 1974] M. Furnival and R.W. Wilson, Jr., *Regressions by Leaps and Bounds*, Technometrics, Vol. 16, No. 4 (Nov., 1974), pp. 499-511

[Golub Heath and Wahba 1979] G.H. Golub, M. Heath and G. Wahba, *Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter*, Technometrics, Vol. 21, No. 2 (May, 1979), pp. 215-223

[Hastie and Tibshirani 1990] T. Hastie and R. Tibshirani, *Generalized additive models*, Chapman & Hall, 1990.

[Hastie Tibshirani and Friedman 2009] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning, 2nd Ed.* (Springer, 2009)

[Hocking 1976] R.R. Hocking, Biometrics Invited Paper. *The Analysis and Selection of Variables in Linear Regression*, Biometrics, Vol. 32, No. 1 (Mar., 1976), pp. 1-49

[Hoerl and Kennard 1970] A.E. Hoerl and R.W. Kennard, *Ridge Regression: Biased Estimation for Nonorthogonal Problems*, Technometrics, Vol. 12 (1970), pp. 55-67

[Kass and Raftery 1995] R.E. Kass and A.E. Raftery, *Bayes Factors*, Journal of the American Statistical Association, Vol. 90, No. 430 (Jun., 1995), pp. 773-795

[Kohavi 1995] R. Kohavi, *A Study of Cross-Validation and Bootstrap for Accuracy Prediction and Model Estimation*, International Joint Conference on Artificial Intelligence (IJCAI), Morgan Kaufmann 1995, pp. 1137-1143

[LaMotte and Hocking 1970] L.R. LaMotte and R.R. Hocking, *Computational Efficiency in the Selection of Regression Variables*, Technometrics, Vol. 12, No. 1 (Feb., 1970), pp. 83-93

[Mallows 1973] C.L. Mallows, *Some Comments on CP*, Technometrics, Vol. 15, No. 4 (Nov., 1973), pp. 661-675

[Meinshausen and Büehlmann 2010] N. Meinshausen and P. Büehlmann, *Stability Selection*, to appear in Journal of the Royal Statistical Society, Series B (2010)

[Mosteller and Tukey 1977] F. Mosteller and J.W. Tukey, *Data Analysis and Regression*, Reading, Mass.: Addison-Wesley

[Pötscher and Leeb 2009] B.M. Pötscher and H. Leeb, *On the distribution of penalized maximum likelihood estimators: The LASSO, SCAD, and thresholding*, Journal of Multivariate Analysis, Vol. 100, No. 9 (Oct., 2009), pp. 2065-2082

[Pötscher and Schneider 2009] B.M. Pötscher and U. Schneider, *On the distribution of the adaptive LASSO estimator*, Journal of Statistical Planning and Inference, Vol. 139, No. 8, (Aug. 2009), pp. 2775-2790.

[Pötscher and Schneider 2010] B.M. Pötscher and U. Schneider, *Confidence Sets Based on Penalized Maximum Likeli-*

*hood Estimators in Gaussian Regression*, Electronic Journal of Statistics, Vol. 4 (2010, pp. 334-360

[Schwarz 1978] G. Schwarz, *Estimating the dimension of a model*, Ann. Statist., Vol. 6, No. 2 (1978), pp 461-464

[Stone 1974] M. Stone, *Cross-Validatory Choice and Assessment of Statistical Predictions*, Journal of the Royal Statistical Society. Series B (Methodological), Vol. 36, No. 2 (1974), pp. 111-147

[Stone 1977] M. Stone, *An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike's Criterion*, Journal of the Royal Statistical Society. Series B (Methodological), Vol. 39, No. 1 (1977), pp. 44-47

[Tibshirani 1996] R. Tibshirani, *Regression Shrinkage and Selection via the Lasso*, Journal of the Royal Statistical Society. Series B (Methodological), Vol. 58, No. 1 (1996), pp. 267-288

[Zhang 2007] C.H. Zhang, *Penalized linear unbiased selection*, Manuscript, 2007.

[Zhang 2008] T. Zhang, *Adaptive Forward-Backward Greedy Algorithm for Sparse Learning with Linear Models*, Neural Information Processing System, 2008

[Zhao and Yu 2006] P. Zhao and B. Yu, *On model selection consistency of the LASSO*, The Journal of Machine Learning Research, Vol. 7 (December 2006), pp. 2541-2563

[Zou and Hastie 2005] H. Zou and T. Hastie, *Regularization and variable selection via the elastic net*, Journal of the Royal Statistical Society. Series B (Methodological), Vol. 67, No. 2 (2005), pp.301-320

[Zou Hastie and Tibshirani 2007] H. Zou, T. Hastie and R. Tibshirani. *On the "degrees of freedom" of the LASSO*, The Annals of Statistics, Vol. 35, No. 5 (2007), pp. 2173-2192

[Zou 2006] H. Zou, *The Adaptive LASSO and its oracle properties*, Journal of the American Statistical Association, Vol. 101, No. 476 (December, 2006) pp. 1418-1429

[Zou and Li 2008] H. Zou and R. Li, *One-step Sparse Estimates in Nonconcave Penalized Likelihood Models*, Annals of Statistics, Vol. 36, No. 5 (August 2008), pp. 1509-1533.