5-2010

Finding a Sparse Solution of a Linear System with Applications to Coding Theory and Statistics

Andrew Gordon Wilcox
College of William and Mary

Finding a Sparse Solution of a Linear System with Applications to Coding Theory and Statistics

A thesis submitted in partial fulfillment of the requirement
for the degree of Bachelor of Science in **Mathematics** from
The College of William and Mary

by

Andrew Gordon Wilcox

Accepted for _____

_____

Tanujit Dey, co-director

_____

Chi-Kwong Li, co-director

_____

Larry Leemis

_____

Donald Campbell

Williamsburg, VA

April 28, 2010

Contents

Abstract

In various applications researchers are presented with the problem of recovering an unknown vector x from an underdetermined linear system. Examples include error correction in coding theory and linear regression in statistics. Underdetermined systems produce an infinite number of solutions, so the focus of applications is to find the "simplest" solution for x which corresponds to the solution that has the fewest non-zero elements. In this thesis we analyze different approaches to solving the problem of recovering a sparse vector from an underdetermined linear system. In particular a new condition on the data matrix A for guaranteeing exact recovery is presented. This result comes from a linear algebraic approach and proves that a general condition for ensuring exact recovery involves including a specific vector within the row space of the matrix A. In addition we compare two well-known model selection techniques, the Dantzig selector and Lasso method, used to solve underdetermined systems in a statistical context. Through numerical experiments we isolate specific situations in which one method outperforms the other.

Chapter 1

Introduction

In various applications researchers are presented with underdetermined linear systems, that is systems of equations that contain more unknown variables (p) then observations (n). The challenge in working with underdetermined linear systems is that the system provides fewer equations than unknowns and as a result produces multiple solutions. Thus it is common in applications to search for the "simplest" soulution where "simplest" corresponds to the solution that gives the least number of non-zero components. This solution in which many of the parameter values are zero is known as a *sparse solution*.

The goal of finding the sparsest solution x in the linear system $y = Ax$ has many unique applications including those relevant to the fields of coding theory and statistics. In this problem y is a known $n \times 1$ vector, A is a known $n \times p$ matrix with $p \gg n$, and x is an unknown $p \times 1$ vector. Following the notation of previous literature, let $||x||_{\ell_0}$ denote the number of non-zero elements of x. Thus the problem of recovering the sparsest solution from an underdetermined linear system can be expressed as

$$\min ||x||_{\ell_0} \quad \text{subject to} \quad y = Ax. \tag{1.1}$$

1.1 Error Correction

In following with the example presented by Candes and Tao [4], the field of coding theory presents one application to solving problem (1.1). One focus in the field

of coding theory is the reliable transmission of data. In this field error correction techniques are necessary when sending data over unreliable or noisy channels. One general idea for reliable transmission is to use repetition in sending the data so that the receiver can check for consistency in the received data. However this approach can not guarantee that the receiver will be able to exactly recover the data. So how can we ensure that the receiver will always be able to exactly recover the transmitted data?

Consider the problem in which we are required to recover an input vector $f \in \mathbb{R}^p$ once it has been corrupted such that $y = Df + e$. Note that f is not restricted to be binary, that is the elements of f can be any real number ($f_i \in \mathbb{R}$ for $i = 1, 2, \ldots, p$). This is due to the fact that some data transmission processes, such as image processing, would not exclusively use $(0, 1)$ elements for f. Real numbers are used as the elements of f and thus f is created over the finite field $\mathbb{R}^p$. In this problem let D be a p by n matrix and let e be a vector of error or corruption. When D is chosen to be linear code, D will create repetition of the data such that the important information f is repeated multiple times in the matrix Df.

Imagine a scenario in which Alice needs to relay important information to her friend Bob across an unreliable channel. Let the elements of D be linear code as to repeat the important information contained in f making it less susceptible to the error contained within the channel. As long as D has full rank then Bob can recover the important information contained in f by using the left inverse of D denoted D^{-1}. If the channel is not corrupted then D^{-1} is the only tool necessary for exact recovery, however what happens if the channel that Alice has to use is corrupted? In this scenario the encoded information Df is corrupted by an arbitrary vector $e \in \mathbb{R}^n$ such that $y = Df + e$. Even if Bob knows the coding matrix D, it is unclear as to whether or not he can exactly recover the original vector f.

It should be noted that if the channel is so corrupted such that e contains non-zero values in most of its entries then there is no hope for the recovery of f. Thus it is commonly assumed in coding theory literature that the number of corrupted elements in e is small, that is $||e||_{\ell_0} \leq s$ where $p \gg n \gg s$.

Now the receiver Bob must have some way to decode f from the information he receives, $y = Df + e$. Since D is known and left invertible, the problem of recovering

f can actually be reduced to determining e. Once e is determined it can be subtracted from y and D^{-1} be applied to recover f. Thus to isolate e, consider a matrix A with dimensions n by p whose columns are exactly orthogonal to the columns of D. Note that orthogonal vectors always produce a product of 0 and as such A annihilates D such that $AD = 0$. In this problem the matrix A can always be found because Alice and Bob have control over D. Therefore D can be chosen such that at least one A matrix exists and A can be determined before the transmission such that A is known to both Alice and Bob. After receiving the vector y Bob can apply A such that $Ay = A(Df + e) = ADf + Ae = Ae$. This error correction problem in coding theory has now been reduced to finding the sparse vector e from the linear system Ae.

$$\hat{y} = Ay = Ae \quad \text{subject to} \quad ||e||_{\ell_0} \leq s. \tag{1.2}$$

1.2 Model Estimation

Today, many research areas collect data in which the number of variables or parameters p is much larger than the number of observations n [5]. Examples include rare disease research, radiology and biomedical imaging, and gene expression data. Modeling and estimation in this scenario where $p \gg n$ is considered to be challenging due to this lack of observations on which to base a model. In addition when $p \gg n$ standard modeling techniques, such as Ordinary Least Squares (OLS), fail to produce a solution.

For an example of a modeling problem where $p \gg n$ consider the field of rare disease research in which there is not a single definitive test for the rare disease. Even though a single definitive test for a rare disease may not exist, a doctor may still want to predict the degree to which a new patient has this rare disease. To do this a doctor would use the tests available at the hospital and base a prediction on the combination of test results. For this model the response vector y corresponds to the degree to which previous patients have had the rare disease and the data matrix A contains previous patients' test results. Using a linear regression model, the parameters $x_1, x_2, \ldots, x_p$ correspond to the weights given to the results of each test.

The test results of previous patients who had the disease can be used to estimate

the values of the true weights $x_1, x_2, \ldots, x_p$. These estimates $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_p$ can then be used to predict the degree to which a new patient has the rare disease. This linear model can be written in matrix notation as $y = Ax + e$ where $e \in \mathbb{R}^n$ corresponds to an error vector that contains any information not explained by the tests in the hospital.

In this scenario the accuracy of the prediction is important so that the patient may be correctly diagnosed. If the estimate for the weights are contained in the vector $\hat{x}$ then the prediction $\hat{y}$ has the error

$$\hat{e} = ||y - \hat{y}||_{\ell_2} = ||y - A\hat{x}||_{\ell_2}, \tag{1.3}$$

where the ℓ_2 norm is defined as

$$||x||_{\ell_2} = \sqrt{\sum_{i=1}^{p} x_i^2}. \tag{1.4}$$

However another important criterion for this model should be *parsimony*. By parsimony, we mean that y should be explained by as few elements of A as possible; x should be sparse. In our scenario a sparse x would provide a much more interpretable model in which only a few tests results would contribute to the final diagnosis. This would allow the doctor and hospital to run fewer tests than would be necessary if all variables were estimated. This desire for parsimony where there are less observations then variables has lead us back to the original problem of finding a sparse solution to an underdetmined linear system with the addition of error:

$$y = Ax + e \quad \text{subject to} \quad ||x||_{\ell_0} \leq s. \tag{1.5}$$

1.3 Model Selection

In any situation in which model estimation is necessary, different models can be created to describe the data. So an important question is how do we choose which model is "best"?

In applications, both the predictive power of a model and the simplicity of a model are important in determining which model is the "best." Thus it is important to find some mathematical representation for each of these ideas. Traditionally the predictive power of a model is represented by the model's prediction error which is calculated by the equation presented in (1.3). The simplicity of a model is represented by the model size ($\hat{p}$) which is calculated as

$$\hat{p} = ||x||_{\ell_0} \leq s. \tag{1.6}$$

Due to the importance and high frequency in which model selection problems arise, there are many proposed model selection techniques in statistical literature. These techniques often vary in the method in which they arrive at the "best" model due to the different applications that each selector was created for. This is due to the fact that applications will put varying emphasis on prediction error and model size. These model selectors could be classified based on the theoretical base of the model, however many of them fit the general form of

$$Q(x) = ||y - Ax||_{\ell_2} + \lambda P_j(a_j). \tag{1.7}$$

Here λ is a constant greater than zero, the ℓ_2 norm is defined in (1.4), and P is a general penalization function. Note that the penalization does not have to be the same for each variable, but for simplicity we will assume that the same penalty function is applied to each variable and present the penalty function as P from here on. From this general form, model selection techniques can be classified by the norm used in their penalization.

1.3.1 ℓ_0 Penalization

The first set of selectors penalize a model based on the number of variables that are included in the model but not their magnitude. As such these model selection techniques can be classified as ℓ_0 penalizations. The goal in these model selection techniques is to minimize $Q(x)$ from (1.7) which incorporates both the accuracy of the estimate and the number of non-zero variables in the estimated model. Note that

penalizations of this type are often inefficient because there is no way to minimize $Q(x)$ without searching through all possible sub-models using the original p variables.

The first of these ℓ_0 penalization is the OLS estimate which is computed by

$$\hat{x} = (A^T A)^{-1} A^T y. \tag{1.8}$$

However, to determine the OLS estimate the matrix A must have full rank p. Thus in the case where $p > n$ the OLS estimate can not be determined. For situations where $p < n$, it can still be considered an ℓ_0 penalization since it selects the coefficients which produce the lowest sum of squared error and it uses $P = 0$ as its penalty. Due to the fact that there is no penalty for including another variable in the model, the OLS estimate always provides a non-zero coefficient for each variable. Thus in addition to not being used in the scenario where $p > n$, the OLS estimate should not be used in situations where variable selection is important.

In an attempt to reduce the number of variables in a model, Mallows introduced Mallow's C_p as a way to determine the best model [16]. This model selection technique uses the set of coefficients, x, that minimizes $C_p(x)$ which is defined as

$$C_p(x) = \frac{||y - Ax||_{\ell_2}}{\hat{\sigma}^2} - (n + 2)||x||_{\ell_0} \tag{1.9}$$

where $\hat{\sigma}^2$ is an estimate for the variance of the matrix A.

A more general criterion was later introduced by Akaike [1]. The Akaike Information Criterion (AIC) evaluates the estimated coefficients, x, at the likelihood function ($L(x)$) of the variables in the model instead of taking the sum of squared error. Again the model chosen is the one in which the AIC is minimized for a given set of coefficients, x

$$-2\log(L(a)) + 2||x||_{\ell_0}. \tag{1.10}$$

Research has shown that the AIC procedure is not consistent and for small samples the AIC can lead to over fitting. As a result Sugiura [19], and Hurvich and Tsai [13] created a new criterion called the AICc which made a second order bias adjustment

to the original AIC.

$$AICc = -2\log(L(a)) + 2||x||_{\ell_0}\left(\frac{n}{n - ||x||_{\ell_0} - 1}\right). \tag{1.11}$$

Finally the Schwarz's Bayesian Information Criterion (BIC) [18] is a penalization scheme that is very similar to the AIC penalization but is justified by Bayesian arguments. Again the chosen model is the one with coefficients x that minimize the BIC.

$$-2\log(L(a)) + 2||x||_{\ell_0}\log(n). \tag{1.12}$$

1.3.2 ℓ_1 Penalization

Instead of using a penalization that only focuses on the number of non-zero estimates many selectors use the sum of the absolute value of the coefficients, or the ℓ_1 norm of x. Here the ℓ_1 norm is defined as

$$||x||_{\ell_1} = \sum_{i=1}^{p} |x_i|. \tag{1.13}$$

This penalization scheme is useful because it has the ability to force many of the coefficients to zero and thus perform variable selection while not requiring a full search of every sub-model using the original p variables.

Tibshirani introduced a new type of selector called the least absolute shrinkage and selection operator (LASSO) in 1996 [20]. This selector was unique to the selectors of its time because it was able to simultaneously select variables and estimate their parameters. Yet despite its uniqueness, the Lasso selector can actually be seen as a penalized likelihood model with an ℓ_1 penalty. It can be written as

$$\min_x(||y - A^T x||_{\ell_2} + \lambda||x||_{\ell_1}), \tag{1.14}$$

where the ℓ_1 norm is defined in (1.13) and the ℓ_2 norm is defined in (1.4). Equation (1.14) defines a convex optimization problem where the unique solution for x depends on the value of λ. This unique solution for x often has many elements of x equal to

zero so that the solution is sparse.

It should be noted that Least Angle Regression (LARS) [9] can be used for computer implementation of the Lasso variable selection technique. Since the unique solution of x depends on the value of λ, LARS can be used to calculate all possible Lasso estimates using various values of λ while using an order of magnitude less computer time then other methods present at the time of the publication. LARS is also able to produce a forward stagewise linear regression estimate but is not needed for the purposes of this thesis.

Researchers later showed the the Lasso selection technique did not have oracle properties and as such Zou introduced the Adaptive Lasso (ALASSO) technique [21]. This technique cleverly weights the coefficients before applying the ℓ_1 norm to achieve the oracle properties that Lasso lacked. Thus, the Adaptive Lasso model selection technique can be written as

$$\min_x(||y - A^T x||_{\ell_2} + \lambda||wx||_{\ell_1}) \tag{1.15}$$

where w is a vector of weights.

The Relaxed Lasso technique (Relaxo) was developed to help account for a slow rate of convergence of the Lasso estimator in sparse high-dimensional data [17]. This selection technique includes a relaxation parameter φ to help control the shrinkage of the coefficients

$$\min_x(||y - A^T x||_{\ell_2} + \varphi||x||_{\ell_1}). \tag{1.16}$$

It should be noted that when $\varphi = 1$ the Relaxo selection technique is equivalent to the Lasso method.

1.3.3 ℓ_2 Penalization

As noted previously, the ℓ_2 norm of a vector uses the square root of the sum of squares of the elements in a vector as defined in (1.4). Using this norm for the penalty function in addition to the estimate for x allows for easy implementation that does not require an exhaustive search of the submodels like the ℓ_0 penalizations do. However the problem with the ℓ_2 penalty is that it does not force any of the elements

of x to zero. Shrinkage is performed on the elements of x, but no variable selection is done.

The model selector that implements an ℓ_2 penalty is known as the Ridge Estimator [10] [11]. It chooses the coefficients $\hat{x}$ that minimizes

$$R(x) = ||y - Ax||_{\ell_2} + \lambda||x||_{\ell_2}. \tag{1.17}$$

Here $\hat{x}$ is defined as $\hat{x} = (A^T A + \lambda I)^{-1} A^T y$, where I is the identity matrix.

1.3.4 Other Penalizations

In addition to the norm penalizations already mentioned, other researchers have developed their own penalization schemes using either a combination of norms or an entirely new norm.

Real-world data often contains many variables that are related and thus should be grouped together when performing regression analysis. It is under this premise that Zou and Hastie presented their elastic net variable selection technique [22]. This technique uses a combination of both the ℓ_1 and ℓ_2 norms to achieve their grouping effect. Thus the coefficients x are chosen to minimize

$$L(\lambda_1, \lambda_2, x) = ||y - Ax||_{\ell_2} + \lambda_1||x||_{\ell_1} + \lambda_2||x||_{\ell_2}. \tag{1.18}$$

Finally another selector was proposed by Candes and Tao to deal with data when the number of variables (p) is much greater then the number of observations (n) [5]. This selector can not necessarily be expressed in the same general penalization formula as presented in (1.7), but it is included in this section because it is an important model selector when dealing with dealing with underdetermined systems. Candes and Tao's Dantzig selector is defined as

$$\min ||x||_{\ell_1} \quad \text{such that} \quad ||A(y - Ax)||_{\ell_\infty} \leq \lambda\sqrt{2\log p} \cdot \sigma \tag{1.19}$$

where the ℓ_∞ norm is defined as

$$||x||_{\ell_\infty} = \max_i(x_i) \quad \text{for} \quad i = 1, 2, \ldots, p. \tag{1.20}$$

While this method is slightly different then the normal model selection techniques, the Dantzig selector still attempts to balance predictive power and parsimony. In this selector the ℓ_1 norm ensures sparsity while using the ℓ_∞ norm to constrain the prediction error to being as small as possible.

1.4 Our Study

The problem of solving an underdetermined linear system for a sparse vector x is one that has applications in various fields. This variety has helped us to see that finding a solution to this problem can be approached in many different ways. Using a combinatorial approach will always lead to exact recovery, but is only computationally feasible for small values of p. Optimization leads to a redefinition of this problem that is more computationally friendly. An iterative and fixed point approach leads to the conditions necessary for equivalence between the true and computationally efficient solution. The use of a linear algebraic approach provides a more general condition and the linear regression approach provides estimates for the true values of x when exact recovery is impossible. Finally, a geometric approach to the problem allows for the visual representation of an abstract problem. The details to all of these approaches will be presented in Chapter 2.

In Chapter 3 we explore the question of what model selection technique provides the "best" model in the setting of an underdetermined system with a sparse vector x. A numerical study of the two well known model selection methods, the Dantzig selector [5] and the Lasso method [20], provides insight into the question of when each selector should be used. In the results of this study it was observed that as the number of observations in A and the level of sparsity in x decreases, the selectors did not perform as well. In addition there were differences in the results of the solutions of the selectors when scenarios involving rank and multi-distributional data, however more research should be done before a definitive conclusion is reached. Finally in

Chapter 4 we conclude with a discussion on possible future work related to this project and open questions in this field of research.

Chapter 2

Different Approaches to the Problem

As stated previously, there are a variety of applications that are interested in recovering a sparse vector from an underdetermined linear system. This variety has created interest from researchers with different math backgrounds. As a result different approaches have been taken in an attempt to solve problem (1.1). In this chapter we look to summarize the different approaches used by ourselves and others.

2.1 Combinatorial Approach

In mathematics, combinatorics can be regarded as the study of the ways to map a set of objects into a finite abstract set with a given structure [3]. This includes the study of the existence, construction, and counting of patterns. An example of this type of approach can be used to calculate the number of objects in permutations and combinations. So how can this type of approach be used in recovering our sparse vector x?

In general an underdetermined system will produce an infinite number of solutions. It is only when x is restricted to being the sparsest solution that the problem is transformed such that a unique solution exists. However, despite this transformation it has been noted that the recovery of a sparse solution from an underdetermined

linear system is NP-hard [7]. This is due to the fact that while we know that many of the elements of the unknown vector x are zero, we do not know at which positions these non-zero elements will be.

One possible approach to finding the positions of the non-zero elements is to analyse every possible position of the non-zero elements. This counting of every possible position for the non-zero elements is a combinatorial approach to our problem

$$\min ||x||_{\ell_0} \quad \text{subject to} \quad y = Ax. \tag{2.1}$$

In general, if the system has a solution then the s-sparse vector x can be recovered by an exhaustive search of the subsets containing s columns of A. This approach is feasible because x is s-sparse. Due to the sparsity, every subset Z containing s columns of A can be used in an attempt to solve the matrix equation $y = Ax$ for x. This procedure will produce all solutions to $y = Ax$, and the sparsest solution can be found. However, this combinatorial approach requires $\binom{p}{s}$ subsets and thus the problem has exponential complexity that is computationally infeasible for large p.

It should be noted however that this approach is only infeasible for large p. For situations where p and s are small and large computing power is accessible, this combinatorial approach will exactly recover the sparse solution x. However, we recognize that in most application settings p is traditionally large and other approaches for recovering x are needed.

2.2 Optimization Approach

The inability to find a computationally feasible solution to (2.1) has lead researchers to look for an alternative program. A good alternative program is one that will both provide the same unique solution as (2.1) while finding the solution in computationally efficient time. This approach is a type of optimization in which researchers look for the best solution to (2.1) from a set of alternative programs.

One of the most frequently used programs in this type of approach is called the *Basis Pursuit* [6]. This alternative program minimizes the ℓ_1 norm instead of the ℓ_0

norm used in problem (2.1)

$$\min \|x\|_{\ell_1} \quad \text{such that} \quad y = Ax. \tag{2.2}$$

The use of the ℓ_1 norm looks to minimize the sum of the absolute value of all of the elements of x instead of counting the number of non-zero elements in x and this difference is what allows for a computationally feasible solution.

In [8] it was shown that the use of the ℓ_1 norm would produce exact equivalence between problems (2.1) and (2.2) in $(p/2) \times p$ matrices obtained by the concatenation of two orthonormal bases for a fixed value of s. While this result was useful in some applications such as signal processing, this result was not applicable to other applications due to the strong restrictions on the data. As such, researchers continued to search for a more general condition on A which guarantees unique equivalent solutions for (2.1) and (2.2).

2.3 Iterative and Fixed Point Approach

Donoho and Hu laid the framework for using the ℓ_1 norm as an alternative program to solving (2.1) [8]. However their result placed strong restrictions on the matrix A. In an attempt to relax these restrictions Candes and Tao showed that (2.1) and (2.2) could yield the same unique equivalent solution without a constant fraction of output [4]. The idea of their paper was to restrict the matrix A in such a way that they could guarantee convergence to a fixed point solution.

In deriving the necessary restrictions for A Candes and Tao introduce the idea of a "restricted almost orthonormal systems." First it should be noted that a set of vectors is said to be orthonormal if each pair of vectors within the set are orthogonal[1], and each vector is normalized to have a Euclidean length of 1. According to the authors a "restricted almost orthonormal system" is "a collection of vectors which behaves like an almost orthonormal system but only for sparse linear combinations." [4]

To determine how close the vectors v_j were behaving like an orthonormal system Candes and Tao defined the quantities δ_m and $\theta_{\alpha m,\beta m}$. By definition δ_m corresponds

[1]Two vectors, x and y are orthogonal if their inner product $\langle x, y\rangle = 0$.

to the smallest eigenvalue in all $m \times m$ submatrices of the matrix $A^T A$. In the same way $\theta_{\alpha m,\beta m}$ corresponds to the largest singular value in all αm by βm off-diagonal submatrices of the original matrix $A^T A$. Thus in the case of sparse linear combinations involving no more then S vectors the values of δ_S, $\theta_{S,S}$, and $\theta_{S,2S}$ can be used to measure how much the vectors v_j behave like an orthonormal system.

Candes and Tao were able to show that (2.1) and (2.2) would yield the same unique equivalent solution if the matrix A was constructed out of these "almost orthonormal" vectors v_j. These restrictions on A allowed the authors to guarantee convergence to a fixed point solution and thus be able to exactly recover the vector x. The main result of [4] is known as the "Uniform Uncertainty Principle" (UUP) and it explicitly states the conditions on $A^T A$ needed to achieve "restircted orthonormality."

Theorem 2.3.1 *Suppose that $S \geq 1$ is such that*

$$\delta_{2S} + \theta_{S,S} + \theta_{S,2S} \leq 1$$

and let c be a real vector supported on a set $T \subset J$ satisfying $|T| \leq S$. Put $y := Ac$. Then c is the unique minimizer to the program

$$\min \|x\|_{\ell_1} \quad Ax = y$$

This theorem ensures that A is an "almost orthonormal system" by using the measures of orthonormality δ_{2S} and $\theta_{S,2S}$ defined above[2].

The proof of Theorem 2.3.1 uses an iterative method to guarantee convergence to a fixed point solution. Each iteration shows the existence of a row vector w such that wA will produce S entries of ± 1 and $p - S$ entries of γ_j up to permutation where all $|\gamma_j| < 1$. The existence of this row vector in each iteration allows them to show convergence. Thus, in each iteration they are simply computing the least squares solution for x and using the restrictions on A to prove convergence to the fixed point solution that is equivalent to the fixed point solution of (2.1).

In the original theorem published in manuscripts, the inequality for their main theorem was $\delta_S + \theta_{S,S} + \theta_{S,2S} \leq 1$. However in trying to understand the proof of

[2]This "Uniform Uncertainty Principle" (UUP) is also known and referred to as the "Restricted Isometry Property"(RIP). Both of these names are used interchangeably in the literature.

this theorem it was found that the bounds for δ_S were incorrect. Through email correspondence we were able to receive confirmation from Terrance Tao that the bound written in the article was incorrect and the theorem should include δ_{2S} as presented in Theorem 2.3.1.

While this theorem has done much to advance the theory of exact recovery in underdetermined linear systems, it leaves the open question of how to check the Uniform Uncertainty Principle in a given matrix. It has been noted by the authors themselves that this condition involving eigenvalues and singular values of $A^T A$ is very difficult to test in a random matrix due to the large volume of submatricies that need to be tested. They note that certain types of matrices seem to obey their Uniform Uncertainty Principle in simulations, but these matrices can only theoretically be shown to obey the Uniform Uncertainty Principle with a high probability for vectors with a large amount of sparsity. Matrices that fit this description include random normalised Gaussian matrices and random normalized Bernoulli matrices. Since there is no computationally-efficient algorithm to test whether or not a matrix obeys the Uniform Uncertainty Principle, the application of this result is limited.

2.4 Linear Algebraic Approach

Candes and Tao used an iterative and fixed-point approach to derive conditions for the matrix A that guarantee that the solution of problem (2.1) is unique and equivalent to the solution given by (2.2). The proof of the Uniform Uncertainty Principle defined in Theorem 2.3.1 was thorough and is probably the best that can be achieved using an iterative and fixed-point approach. However their use of the row vector w and our own insight inspired us to approach the problem from a linear algebraic point of view.

Clearly any solution to either (2.1) or (2.2) needed to be unique. As such we started our linear algebraic approach by looking for linear algebraic conditions on A that would require uniqueness in both (2.1) and (2.2). In linear algebra terms, having a unique solution is equivalent to saying that A has $2s$ linearly independent columns where s is the sparsity of the vector x. In addition, the ability to find the vector w

is equivalent to being able to solve the problem which minimizes the ℓ_1 norm[3]. Thus to guarantee this vector w, a condition on the row space of A is needed. This insight lead to the following proposition and corollary [15].

Proposition 2.4.1 *Let A be an $n \times p$ real matrix with rank n and satisfy the following.*

(†) For each choice of $1 \leq i_1 < \cdots < i_s \leq p$ and $\mu_1, \ldots, \mu_s \in \{1, -1\}$, there is a vector v in the row space of A such that the i_rth entry equals μ_r for $r = 1, \ldots, s$, and all other entries have moduli less than 1.

If there is a $c \in \mathbf{R}^p$ such that $Ac = y$ with $||c||_{\ell_1}$, then c is the unique vector with minimum ℓ_1-norm such that $Ac = y$.

Proof. Suppose $Fc = y$ and $Ad = y$ such that c has at most s nonzero entries and d has the minimum ℓ_1-norm. Then we can choose $w \in \mathbf{R}^{1 \times n}$ such that

$$\ell_1(c) = wAc = |wAc| = |wAd| \leq \ell_1(d) \leq \ell_1(c).$$

Thus, $\ell_1(d) = \ell_1(c)$ and d can only have nonzero entries at those position of c. Since any $2s$ columns of F is linearly independent, we see that $d = c$.

If $\tilde{c}$ is another solution of $Ax = y$ with at most s entries, then $\tilde{c} = d = c$. Thus, the problems (P1) and (P2) have the unique solution c. □

Corollary 2.4.2 *Suppose A satisfies (†). Attempting to solve $Ad = y$ for the solution with the minimum ℓ_1-norm, results in one of the following outcomes:*

(a) *The system has no solution.*

(b) *The vector d with minimum ℓ_1 norm has more than s nonzero entries, then $Ax = y$ has no solution with at most s nonzero entries.*

(c) *The vector d has no more than s entries. Then d is the unique solution for (P1) and (P2).*

From Proposition 2.4.1 we see that the restriction on A needed for equivalence conditions the row space of A. With basic linear algebra techniques we can produce an alternative to this proposition which conditions the null space of A [15].

[3]The vector w is the vector described when describing the proof of Theorem 2.3.1 in section 2.3. It has the property that wA will produce S entries of ± 1 and $p - S$ entries of γ_j up to permutation where all $|\gamma_j| < 1$.

Proposition 2.4.3 *Suppose A is an $n \times p$ real matrix with rank n. Let G be an $(p-n) \times p$ matrix such that $G[G^t|F^t] = [I_{n-m}|0_{n-p,p}]$. Let $\mathcal{C}$ be the set of column vector of G. Then (†) holds if and only if the following holds.*

(‡) For any partition of $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$ such that $\mathcal{C}_1 = \{v_1, \ldots, v_s\}$ and $\mathcal{C}_2 = \{v_{s+1}, \ldots, v_p\}$, the set

$$\left\{ \sum_{j=1}^{s} \mu_j v_j : \mu_1, \ldots, \mu_s \in \{1, -1\} \right\}$$

is a subset of the interior of

$$\text{conv} \left\{ \sum_{j=s+1}^{p} \mu_j v_j : \mu_1, \ldots, \mu_s \in \{1, -1\} \right\}.$$

Proof. Note that v lies in the row space of A if and only if v^t lies in the column space of A. One easily checks that the (†) and (‡) are equivalent. □

Denote by GP_r the set of generalized permutation matrices in M_r, and denote by $\mathbf{1}_r$ the vector of all 1 in $\mathbb{R}^r$.

Corollary 2.4.4 *Suppose A and G are defined as in the above propostion. Then (‡) fails if and only if there is a $1 \times (n-p)$ vector v and $Q \in GP_p$ such that*

$$vGQ(I_s \oplus P)\mathbf{1}_1 \geq 0 \qquad \text{for all diagonal matrices } P \in GP_{p-s}.$$

Proof. Use the fact that a convex polytope S_1 does not lie in the interior of another convex polytope S_2 if and only if there is a linear functional f and a vertex x of S_1 such that $f(x) \geq f(y)$ for all vectices $y \in S_2$. □

By using a linear algebraic approach we were able to find conditions that restrict the row and null space of the matrix A. It is believed that Proposition 2.4.1 is a necessary condition for the matrices that obey the Uniform Uncertainty Principle presented in (Theorem 2.3.1). However this is presently unproven and is a goal that is discussed more in Chapter 4.

Unfortunately, just like the Uniform Uncertainty Principle, the condition proposed in Proposition 2.4.1 is not able to checked for a given matrix A. As of now there is no computationally feasible way to check whether or not a specific vector lies within the row space of a matrix. This compounds our problem of trying to prove the necessity of Proposition 2.4.1 since it is impossible to even numerically test which condition is more general. Both the computational feasibility of Theorem 2.3.1 and Proposition 2.4.1 and the question as to which condition is more general are open research topics that arose from this thesis and are discussed in Chapter 4.

2.5 Linear Regression Approach

In a noiseless scenario the sparse vector x can be exactly recovered by using the alternative program (2.2) as long as the matrix A obeys the Uniform Uncertainty Principle or adheres to Proposition 2.4.1. However in many applications noise or error is present. So how do we recover a sparse vector x when the matrix A has more parameters then observations and is corrupted by noise?

In statistics the standard approach for estimating x in a linear systems is a linear regression model. This model can be mathematically defined as

$$y = Ax + e \tag{2.3}$$

where y is a $n \times 1$ vector of responses, A is a $n \times p$ data matrix, x is a $p \times 1$ unknown vector of coefficients, and e is an $n \times 1$ vector of random error. When we consider the situation where $p \gg n$ and x is sparse, we see that solving for a linear regression model from (2.3) is very similar to solving for x in our initial problem (2.1). Note that in this setting exact recovery of x is impossible due to the noise e, however estimating $\hat{x}$ close to the true value of x using linear regression models is feasible.

The linear regression approach should be able to find the "best" model for estimating x. As discussed in Section 1.3 model selection can be a daunting task due to the large number of model selection criteria available for linear regression. All of these selection criteria look to balance the importance of prediction error and parsimony. The simplest and most used model selection criteria used for linear regression

is Ordinary Least Squares (OLS). However as previously shown, OLS estimates do not exist when $p > n$. In addition since we are trying to find a sparse vector x, variable selection should be an important component of whatever variable selection technique is used. Thus variable selection techniques such as ridge regression which do not provide any variable selection would provide a poor estimate for x because it does not set any of the elements of $\hat{x}$ equal to zero.

When the number of variables p is larger then the number of observations n we assume that most the data sets will have large values of p. Thus we should exclude model selection criteria that rely on searching through all subsets of p since this is not computationally feasible for large values of p. Thus many of the ℓ_0 penalizations such as AIC, AICc, and BIC are not feasible in underdetermined systems where p is large.

Thus in our linear regression approach to problem (2.3) we looked for model selection techniques that both minimized prediction error and estimated sparsity in the vector x. Two methods that fit this approach were the Dantzig selector and the Lasso method. Both of these methods use linear programs that are able to simultaneously select which variables should be non-zero and estimate their coefficients. However the way in which the programs select the variables and estimate the coefficients are different.

The Dantzig selector was created by Emanual Candes and Terrance Tao as an extension to their previous results of exactly recovering a sparse vector from an underdetermined system [4]. The Dantzig Selector is used when exact recovery is not possible because error is present [5]. This selector looks to restrain the number of parameters within the model by using the ℓ_∞ norm. The ℓ_∞ norm is the norm that takes the element with the largest absolute value from the vector. The selector computes the estimate $\hat{x}$ using the following program

$$\min ||x||_{\ell_1} \quad \text{such that} \quad ||A * (y - Ax)||_{\ell_\infty} \leq \lambda\sqrt{2\log p} \cdot \sigma \tag{2.4}$$

where σ is the standard deviation of the data in the matrix A and λ is a constant which is greater than 0. Note that in this program the ℓ_1 norm is used to ensure sparsity while using the ℓ_∞ norm is used to constrain the prediction error. In fact it

can be shown that there is an upper bound on the prediction error that is equivalent to

$$||\widehat{x} - x||_{\ell 2}^2 \leq C^2 \cdot 2\log p \cdot \left(\sigma^2 + \sum_i \min(x_i^2, \sigma^2)\right) \tag{2.5}$$

provided that A obeys the Uniform Uncertainty Principle and the true x is sufficiently sparse [5].

An alternative approach to the Dantzig Selector for estimation a sparse vector in an underdetermined system is Tibrishani's Lasso method [20]. The Lasso method seeks to minimizes the residual sum of squares while making sure that the ℓ_1 norm of the coefficients stays less than a constant for variable selection:

$$\min \ ||(y - Ax)||_{\ell 2} \quad \text{such that} \quad ||x||_{\ell 1} \leq s. \tag{2.6}$$

Again the Lasso technique simultaneously selects variables and estimates their parameters. The implementation of this model selection technique is very efficient when computed by the LARS algorithm [9].

Note that both the Dantzig and Lasso selectors fit the criteria for being the "best" model in the case of a sparse vector in an underdetermined linear system. They both provide a model selection tool that can be used in the case of large p, perform variable selection, and constrain prediction error.

2.6 Geometrical Approach

Finally the problem of recovering a sparse vector x from an underdetermined linear system can be approached from a geometric perspective. In the different approaches presented thus far, vector norms have frequently been used to help solve the problem of finding a sparse vector in an underdetermined linear system. The ℓ_0 norm is used in the original definition of the problem, the ℓ_1 norm is used in the basis pursuit and Lasso method, the ℓ_2 norm is used in finding prediction error, and the ℓ_∞ is used in the Dantzig selector.

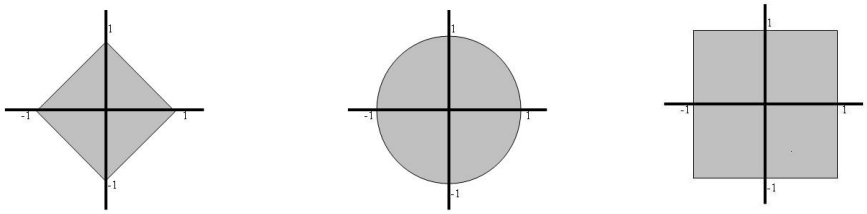
In the most basic sense a norm is simply a function that assigns sizes to vectors within a given vector space. By definition a vector norm must satisfy the following

properties [12].

Definition 2.6.1 *Let V be a vector space over a field $\boldsymbol{F}$. A function $||\bullet|| : V \to R$ is a vector norm if for all $x, y \in V$,*

1. $||x|| \geq 0$,
2. $||x|| = 0$ *if and only if* $x = 0$,
3. $||cx|| = |c|||x||$ *for all scalars* $c \in \boldsymbol{F}$,
4. $||x + y|| \leq ||x|| + ||y||$.

Since a norm is simply an operator on a vector space, it is possible to visualize the vector space in low dimensions and visually observe the processes of using norms. In fact in $\mathbb{R}^2$ it is possible to represent the set of all vectors normalized to 1 for each of the different vector norms.

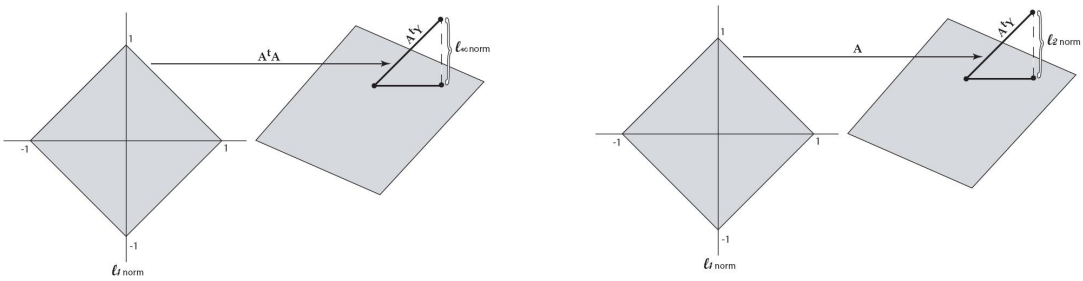


(a) ℓ_1 Norm (b) ℓ_2 Norm (c) ℓ_∞ Norm

Figure 2.1: Geometric interpretation of different vector norms

This geometric interpretation of vectors and vector norms helped us understand the fixed point and iterative approach used by Candes and Tao in their proof of the Uniform Uncertainty Principle [4]. In addition this geometric approach can also be used to understand the programs used by the Dantzig selector and Lasso method. Essentially both the Dantzig selector and Lasso method transform the ℓ_1 norm ball by a function of the data matrix A. The selectors then use a vector norm to minimize the distance between their estimate and the true solution. This can be seen visually in Figure (2.2).

In addition by studying the geometry of the model selection techniques we can observe the relationship between the two selectors in a two dimensional space. The



(a) Dantzig Selector (b) Lasso Method

Figure 2.2: The geometry of the Dantzig Selector (a) and the Lasso Method (b)

solutions to the Dantzig and Lasso methods are equivalent to the solutions of the following equations where $\hat{x}_{ls}$ is equivalent to the least squares estimate [14].

$$\min ||x|| \quad \text{subject to} \quad ||A^T A(x - \hat{x}_{ls})||_\infty \leq \lambda_D \tag{2.7}$$

$$\min ||x|| \quad \text{subject to} \quad ||A(x - \hat{x}_{ls})||_\infty^2 \leq \lambda_L \tag{2.8}$$

From (2.7) and (2.8) we can see that the Dantzig solution must lie within the ℓ_∞ norm ball centred at $\hat{x}_{ls}$ and the Lasso solution must lie within the ℓ_2 norm ball centred at $\hat{x}_{ls}$. Using this geometrical thinking it can be shown that these solutions will always be the same in a two dimensional setting [14]. Figure 2.3 is taken from [14] and shows the equivalence of the two solutions for differing values of ρ where ρ corresponds to the correlation between the two columns of A. Note that the equivalence of the solutions does not depend on ρ which changes the shape of the feasible region.

Through this type of geometrical thinking many more relationships between the solution produce by the Dantzig and Lasso methods can be produced. One notable relationship found in this way is that the Dantzig selector will always produces a sparser solution then the Lasso method when $\lambda_D = \lambda_S$ [14]. In addition, a paper recently presented at the Conference of Information Sciences and Systems presented a set of conditions for which the solution of the Dantzig and Lasso methods are the same for dimensions larger than 2 [2]. By studying the geometry of norms and the different model selection techniques, connections can be made between the solutions of the Dantzig and Lasso techniques.

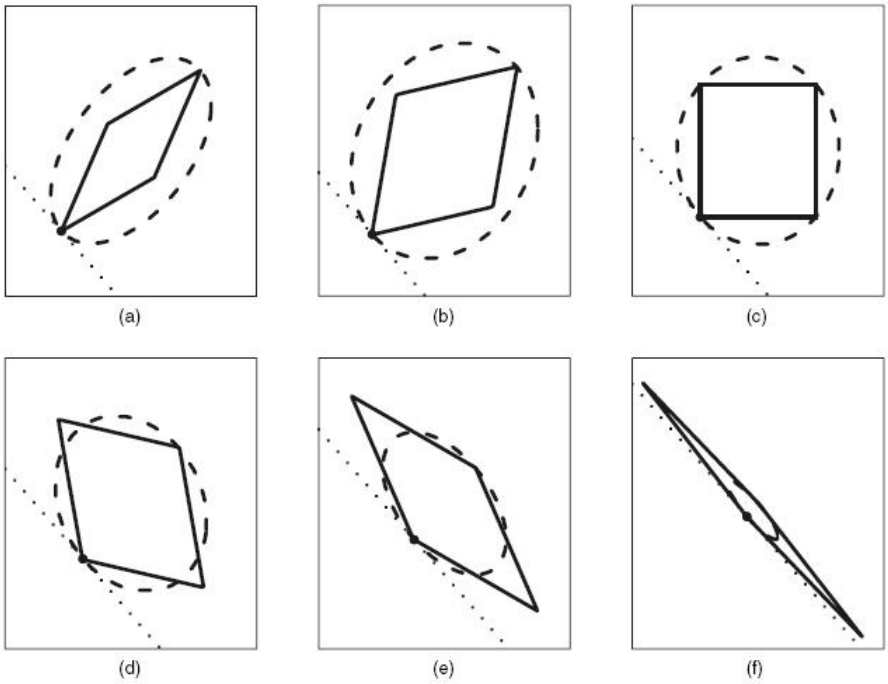


Figure 2.3: Lasso (—) and Dantzig Selector ($\diamond$) solutions in a $p = 2$ dimensional space. Here the ℓ_1 norm $(\cdots)$ is being minimized.
(a) $\rho = -.5$; (b) $\rho = -.2$; (c) $\rho = 0$; (d) $\rho = .2$; (e) $\rho = .5$; (f) $\rho = .9$

Chapter 3

Numerical Study

As noted in Chapter 2, if we approach the problem of recovering a sparse vector from an underdetermined linear system through linear regression analysis both the Dantzig selector and Lasso perform well. Both methods are able to estimate coefficients while simultaneously performing variable selection to ensure a sparse estimate. In this section we present a series of simulation studies to provide performance comparisons between the estimated models of the Dantzig selector and Lasso method. Each simulation study was performed in an attempted to isolate a single characteristic of the data and observe changes in model performance based on perturbations of that characteristic. In each set of simulations we generated 100 data sets to test the models produced by the Dantzig selector and Lasso method.

It is apparent from (2.4) and (2.6) that the estimation of both the Dantzig and Lasso model relies heavily on the value λ in the Dantzig model and the value of s in the Lasso model. These values can be thought of as tuning parameters for the model and the way in which these tuning parameters are found has also been an important research topic. Recently a simulation study was performed to analyse the different approaches of selecting λ and s [14]. The results of this study showed that one effective way of choosing the tuning parameters was to test different values for λ or s and choose the value that produced the lowest mean squared error over 10-fold cross validation. In 10-fold cross validation the data is first randomly divided into 10 sub-samples of approximately equal size. Then the model is trained using 9 of the subsets and tested on the 1 subset not used for training. This process is repeated

10 times and the mean squared error is then averaged to produce one error estimate. The parameter value λ or s that is chosen, is the parameter that produced the lowest mean squared error from the 10-fold cross-validation.

Another effective approach is to run the selectors twice in a procedure known as Double Dantzig or Double Lasso respectively [14]. This procedure first fits the data using the theoretical optimum tuning parameters $\lambda = \sigma\sqrt{2\log p}$ and $s = \sigma\sqrt{2\log p}$ before discarding all zero parameters and running the selectors again. On this second training of the model 10-fold cross-validation was used on this subset of variables to find the new optimal value for λ or s. Again, λ and s were chosen to be the parameter that produced the lowest mean squared error in the 10-fold cross-validation on the subset of parameters. These doubling procedures tended to produce a much sparser solution when compared with straight cross-validation but did require more computing time [14]. Since both ways were effective in calculating the tuning parameter and straight cross-validation used less computing time, in our study we simply performed ten-fold cross-validation over a net of values for for λ and s to obtain the parameter used in the estimation of our models.

To analyse the performance of the Dantzig and Lasso model selection techniques we used three different measures. The first measure is prediction error as defined in (1.3). It is common practice in statistical simulations to obtain the prediction error for a model by using a percentage of the data to train the model and the remaining percentage to test the model. Thus in our simulations the estimate $\hat{x}$ was obtained from training the selectors on 80% of our original data. This 80% was chosen randomly without replacement so that no observation held more weight than another. The prediction error was then calculated by computing $A\hat{x}$ for the remaining 20% of the data and comparing the computed values to the known values of y for that subsection of the data.

The second measure of the selectors is what is commonly referred to as the misclassification rate of the parameter. This rate is calculated by defining each estimated parameter as a zero or non-zero value. The number of misclassifications for a single simulation is taken to be the number of times a true non-zero parameter is estimated as zero plus the number of times a true zero parameter is estimated as non-zero [1]. The

[1] In statistical literature this is sometimes referred to as the number of false positives and the number of false negatives respectively.

total number of misclassifications is then calculated by summing all of the misclassifications for all data sets generated. Finally the misclassification rate is calculated by dividing the total number of misclassifications by the number of data sets generated, which was 100 in all of our simulation studies.

Finally the number of non-zero parameters estimated or the model size as defined in (1.6) is also calculated. This measure can be used to compare which selector is providing a sparser solution in each of our settings. In addition by looking at the misclassification rate along with the model size allows us to see how many of the true non-zero parameters the model is estimating as non-zero parameters on average.

3.1 Size of the Matrix A

In our first simulation study we looked to see if the number of observations in our data matrix A had any effect in estimating a model. For this study the matrix A was generated using independently identically distributed random variables from a standard normal distribution. For the vector x we independently generated five non-zero values from a normal distribution with mean 0 and standard deviation 2. Therefore the ideal model size in this setting is 5. These non-zero values were then inserted into random positions of the vector x as not to restrict our study to one setting of x. Finally the true values for y were calculated by standard matrix multiplication such that $y = Ax$.

In all of our settings we wanted to test the selectors both with and without the presence of error. This will hopefully lead to a better understanding of the selectors as they strive for exact recovery in the case of no error and strive for estimation of the best model in the more realistic setting where error is present. In the case where error is present, we calculate the vector e by independently generating its entries from a normal distribution with a mean 0 and standard deviation 0.3. The true values for y were then calculated as $y = Ax + e$. While many simulation studies use error values generated from the standard normal distribution with a mean of 0 and a standard deviation of 1, we did not want the error term to overpower the data. By choosing a smaller standard deviation we were able to constrain the entries in the error vector to be closer to 0. This ensures that the error entries will be smaller than the true

values of y with a higher probability.

The size of the matrix is an important characteristic to study for these selectors because it shows how the number of observations effect the performance of these selectors. The Dantzig selector was designed to handle cases where $p \gg n$ and the Lasso selector has been shown to produce similar results for cases with a low number of observations [14]. However it is important to see if there is a threshold for the number of observations that must be present to perform variable selection. We clearly expect that as the number of observations decreases the predictive power of both selectors will also decrease. However it is unclear as to whether or not they will decrease at a similar rate or whether the threshold is the same for both selectors.

The data matrix A was originally generated as a 50×200 matrix such that there were 200 possible parameters and 50 observations. To perturb the size of the data matrix A we changed the testing and training percentages as discussed in the beginning of this chapter. We studied three different sizes for A when training the selectors: 45×200, 35×200, and 25×200. These values correspond to training and testing percentages of 90%/10%, 70%/30%, and 50%/50%. It should also be noted that any parameter that was estimated by the model to be $< .0001$ was taken to be zero for the calculation of the misclassification rate and the model size. The results of this simulation study is presented in Table 3.1. In the table the prediction error (PE), misclassification rate (Miss), and model size (Msize) is presented with their Monte Carlo standard deviations below the result in parenthesis.

Table 3.1 confirms our intuition that decreasing the number of observations decreases the predictive power of the model. In the case with no error this is due to the fact that with fewer observations the selectors generated models with a larger model size. Note that with 45 observations and no error the selectors were able to correctly identify the true non-zero parameters but as the number of observations decreased, the selectors predicted a larger model size which had to include parameters that were zero for the true x. This larger model size lead to a larger prediction error and higher misclassification rate.

In the setting with error the Dantzig selector follows the same pattern of predicting a larger model size as the number of observations decreases. However, the Lasso selector predicts the largest model size when there are 45 observations and its

Table 3.1: Simulation results for perturbations in the size of the matrix A

No Error			
Obs.	Measure	Lasso	Dantzig
	PE	.0001	0.05
		(.0012)	(0.58)
45	Miss	0.02	0.01
		(0.20)	(1.00)
	Msize	5.02	5.10
		(0.20)	(1.00)
	PE	4.98	0.72
		(18.85)	(3.92)
35	Miss	2.21	0.45
		(5.44)	(2.17)
	Msize	6.99	5.39
		(5.14)	(1.96)
	PE	52.07	33.30
		(115.55)	(61.29)
25	Miss	8.32	5.24
		(7.77)	(6.87)
	Msize	11.70	8.64
		(7.28)	(6.14)

With Error			
Obs.	Measure	Lasso	Dantzig
	PE	1.05	2.90
		(0.75)	(3.27)
45	Miss	26.12	6.29
		(9.11)	(9.99)
	Msize	30.68	10.27
		(9.15)	(10.31)
	PE	8.09	13.84
		(16.35)	(13.42)
35	Miss	21.33	8.77
		(7.06)	(9.09)
	Msize	25.67	12.53
		(7.26)	(9.76)
	PE	124.30	89.59
		(250.48)	(166.30)
25	Miss	14.12	10.67
		(6.29)	(7.06)
	Msize	16.56	13.01
		(6.63)	(7.82)

smallest model size when there were 25 observations. This is a curious result but could potentially be explained by Lasso's use of least squares. With error present a situation could be created in which the least squares estimate was able to use a combination of positive and negative coefficients to create estimates with smaller prediction error then any of the more sparse models. Thus a larger model size using these combinations could be present even when the number of observations is the largest.

In this simulation it is impossible to conclude that one selector clearly outperforms the other for certain sizes of A. The Dantzig estimator was always able to provide a sparser model then the Lasso method but the prediction error for both selectors were very similar, normally within a standard deviation of one another. It is clear that having more observations is clearly beneficial to both selectors in terms of predictive power, but this was clear to us through intuition.

3.2 Rank of Matrix A

Our second simulation experiment looked to isolate the effect of the rank of the data matrix A. In the standard linear regression technique of OLS, the matrix A must be of full rank so that there is no multicollinearity between the variables such that x can be identified. While our selectors do not carry this same assumption, we are interested to see if the rank of A has any effect on our two methods.

The data sets used in this second simulation had a very similar structure to the data sets used in the first simulation. Once again the vector x had five randomly positioned independently generated non-zero values from a normal distribution with mean 0 and standard deviation 2. The error vector e was again calculated from a normal distribution with a mean of zero and a standard deviation of 0.3 and the true values for y were calculated by standard matrix multiplication such that $y = Ax$ or $y = Ax + e$ in the presence of error.

The data matrix A had entries that were once again independently and identically distributed from a standard normal distribution with mean 0 and standard deviation 1. To control the rank of the matrix, however, a larger matrix C was created by multiplying $B^T B$ where B was a full-rank matrix with dimensions of $r \times 200$. In this case r stands for the desired rank of the data matrix A. The 50×200 matrix A was then created by taking the first 50 rows from the larger square matrix C. This algorithm ensured that the rank of A was exactly equal to r. The results of our simulation study on rank are presented in Table 3.2 which uses the same notation as Table 3.1[2].

First, note that in the setting where error is not present the Dantzig selector is able to produce exact recovery for full-rank matrices. This results possibly shows that full-rank normal matrices obey the conditions necessary for exact recovery as presented in Chapter 2. In contrast the Lasso selector struggles greatly for full-rank matrices producing large and variable prediction error with a model size three and a half times larger then the true model. Note that in the case of low-rank matrices both the Dantzig and Lasso method were able to produce low prediction errors despite not being able to correctly identify the true non-zero parameters. This is possibly due to

[2]Note that in this study the prediction error is calculated using the 80%/20% training/test percentage as presented in the beginning of Chapter 3.

Table 3.2: Simulation results for a change in the rank of A

No Error			
Rank	Measure	Lasso	Dantzig
5	PE	1.06e-27	0.13
		(2.82e-27)	(0.72)
	Miss	10.35	9.24
		(1.36)	(1.19)
	Msize	5.89	5.00
		(0.90)	(0.00)
25	PE	34.34	2.92e-16
		(242.23)	(2.24e-15)
	Miss	21.43	9.13
		(8.89)	(11.26)
	Msize	23.03	12.99
		(7.39)	(9.83)
50	PE	248.14	1.47e-19
		(576.40)	(2.46e-19)
	Miss	13.12	0.00
		(13.17)	(0.00)
	Msize	17.68	5.00
		(13.40)	(0.00)

With Error			
Rank	Measure	Lasso	Dantzig
5	PE	1.09	1.62
		(0.58)	(1.31)
	Miss	10.34	8.91
		(1.49)	(1.22)
	Msize	6.16	4.81
		(1.02)	(0.52)
25	PE	2.70	9.14
		(2.01)	(22.13)
	Miss	21.17	16.83
		(6.02)	(6.38)
	Msize	23.71	20.49
		(4.50)	(5.63)
50	PE	21.73	38.02
		(170.67)	(90.36)
	Miss	28.63	14.22
		(4.78)	(10.05)
	Msize	33.53	19.04
		(4.74)	(10.13)

the fact that the variables contain multicollinearity and other variables can be used to produce results similar to the true values.

When error is added to the system we seem to get very curious results. In these situations both selectors performed better in the low-rank setting and were comparable to one another in all settings. After seeing the results in the setting with no error we would have expected the Dantzig selector to perform very well in the case of full-rank matrices, but note that this is not the case. This possibly suggests that the addition of error to the system changes the matrix structure so drastically that our results with error have no real connection to our results without error.

3.3 Sparsity of the Vector x

The vector x is an unknown and in application settings it may be difficult to tell what the true sparsity of x will be. However, in some cases there is an intuition or a justifiable reason for why only a certain number of variables are important.

In these settings it is important to know how the level of sparsity in x affects the models produced by the Dantzig and Lasso selectors. In addition from a theoretical perspective this is important to understanding why the selectors perform better in certain situations.

The set-up for this simulation is similar to the set-up described earlier. A is once again a 50×200 matrix whose entries were independently and identically drawn from a standard normal distribution with mean 0 and standard deviation of 1. In addition, the vector x had s randomly positioned independently generated non-zero values from a normal distribution with mean 0 and standard deviation 2, where s is the level of sparsity that we are testing. The error vector e was again calculated from a normal distribution with mean zero and standard deviation 0.3 and the true values for y were calculated by standard matrix multiplication such that $y = Ax$ or $y = Ax + e$ in the presence of error.

Our intuition is that a true solution that is more sparse will be easier to recover then a true solution that is less sparse (has more non-zero values). A sparser solution requires estimates for fewer columns of the matrix A and thus it should be more difficult to find an incorrect combination of columns. The results of this study are presented in Table 3.3 and the notation in this table is consistent with our previously used notation.

At the highest level of sparsity (fewest number of non-zero values) and no error we see that the Dantzig selector is able to exactly recover the unknown vector x. As the number of non-zero parameters increases the predictive power of both models decreases. This is consistent with our intuition that the methods perform better at high levels of sparsity. Note that 5 non-zero parameters corresponds to $5/200 = 0.025$ of the entries in A. As the number of non-zero values increases to 5% and 10% the prediction error and model size drastically increases for both selectors. In fact the magnitude of the errors and model size possibly show that 5% non-zero values may be the largest level that can be accurately handled by these selectors.

When error is added to the system, the Dantzig selector again is not able to exactly recover and in fact the Lasso prediction error is on average better then the Dantzig selector prediction error. However the large discrepancy in model size makes it very difficult to conclude that the Lasso method is performing better in the presence of

[h]

Table 3.3: Simulation results for a change in the level of sparsity of x

No Error				With Error			
# Non-zero	Measure	Lasso	Dantzig	# Non-zero	Measure	Lasso	Dantzig
	PE	1.03	1.10e-21		PE	2.24	6.06
		(10.27)	(4.74e-21)			(1.25)	(7.11)
5	Miss	0.54	0.00	5	Miss	24.22	7.51
		(2.86)	(0.00)			(8.00)	(9.99)
	Msize	5.54	5.00		Msize	28.82	11.61
		(2.87)	(0.00)			(8.02)	(10.25)
	PE	70.84	41.54		PE	43.17	36.84
		(146.61)	(104.94)			(95.67)	(44.13)
10	Miss	16.86	13.27	10	Miss	23.49	22.22
		(9.99)	(12.48)			(8.30)	(9.89)
	Msize	22.62	20.41		Msize	29.49	27.74
		(9.88)	(10.81)			(9.60)	(11.61)
	PE	358.11	330.59		PE	395.10	378.23
		(255.69)	(244.26)			(316.81)	(324.90)
20	Miss	28.72	33.16	20	Miss	29.81	32.99
		(7.57)	(5.66)			(7.23)	(6.08)
	Msize	26.34	33.7		Msize	27.79	33.57
		(10.55)	(6.27)			(10.37)	(6.23)

error. In this setting the trend of increasing prediction error and model size as sparsity decreases just like the case of no-error. This is different from our results on the rank of the matrix in Table 3.2 where the trend present with no error was not present once error was added. Both methods produce models that perform similarly in all cases and thus we can not claim that one method outperforms the other.

3.4 Multi-Distributional Data

Our simulation studies thus far have looked exclusively at situations in which the data matrix A was drawn from a single distribution. This setting is very standard and accepted to be true for many data sets in applications, but situations arise when data is actually drawn from two or more distributions. In this setting data that is drawn from one distribution can be seen as the data driving the resulting vector y, while all other distributions are just noise within the data matrix. Note that this noise we are referring to within the matrix is different from the error vector e that

is used to capture what the data can not explain. This data noise can be thought of as being irrelevant variables or variables that have no explanatory power. Since the variables from extra distributions have no predictive power with respect to y, they should always be estimated to have a zero coefficient.

In this setting the selectors might have a difficult time recognizing the different distributions that are present within the data. Thus, we did not want to make the simulation any more challenging then what was necessary. To accomplish this we reduced the size of our A matrix to 20×20 so that there were both fewer variables to decide between and the number of observations was equivalent to the number variables. This increase in the proportion of observations to variables should help the selectors produce a lower prediction error and find a model size closer to the ideal model size. In this simulation the coefficient vector x once again had non-zero values from a normal distribution with mean 0 and standard deviation 2. However instead of randomly distributing these non-zero parameters we grouped them together and paired them with the single "true" distribution chosen for A. To ensure that the position of the non-zero coefficients did not make a difference we performed simulation studies where the position of the non-zero coefficients and true distribution was in the first, middle, and last columns of the data matrix. When an error vector was introduced to the simulation, the error was drawn from a normal distribution with mean 0 and standard deviation 0.3. As always the response vector was calculated by $y = Ax$ or $y = Ax + e$ depending on whether or not error was present.

To replicate the situation in which the data matrix includes many distributions, we had to decide on different distributions for A. For this simulation study, the standard normal distribution was chosen to be the true distribution in which the response vector y was drawn from. Since the sparsity was still equal to 5 in this setting, 5 columns of A were drawn from this standard normal distribution. To add data noise within the data matrix we drew the other entries in A from a chi-squared distribution with one degree of freedom and a linear combination of the 5 columns already chosen from the standard normal distribution. Ten of the columns of A were drawn from the chi-squared distribution and the remaining 5 columns were the linear combination of the 5 columns drawn from the standard normal distribution. By using a linear combination of the 5 true columns we were able to observe whether or not the

selectors could distinguish between two sets of vectors that shared the same vector space. In addition we are able to observe whether or not the selectors could ignore the noise which is drawn from a different distribution.

Our intuition is that the selectors should be able to create a model based on a combination of columns from both the standard normal distribution and the linear combination. If these selectors can be used in any potential multi-distributional settings then they need to be able to ignore the chi-squared distribution[3] and only focus on vectors that use the same vector space. The results of this simulation are shown in Table 3.4 using all of our previous notation. The form of A corresponds to what distribution occupied the first five columns of A.

Table 3.4: Simulation results for a multi-distributional A

No Error			
Form A	Measure	Lasso	Dantzig
True First	PE	0.03 (0.20)	1.77 (6.25)
	Miss	3.40 (1.85)	3.49 (1.90)
	Msize	5.18 (0.75)	4.99 (0.50)
Comb. First	PE	0.18 (1.72)	0.82 (4.30)
	Miss	3.49 (1.79)	3.42 (1.77)
	Msize	5.17 (0.79)	5.00 (0.40)
Noise First	PE	7.52e-28 (1.79e-27)	0.66 (3.37)
	Miss	3.21 (1.78)	3.40 (1.76)
	Msize	5.01 (0.10)	5.08 (0.39)

With Error			
Form A	Measure	Lasso	Dantzig
True First	PE	54.39 (211.98)	7.03 (21.53)
	Miss	12.19 (2.52)	7.00 (4.26)
	Msize	14.01 (2.14)	8.22 (4.54)
Comb. First	PE	3.04e5 (3.02e6)	9.49 (23.50)
	Miss	12.45 (2.58)	6.74 (3.99)
	Msize	14.15 (1.64)	7.74 (4.33)
Noise First	PE	1.20e3 (1.08e4)	3.81 (5.73)
	Miss	12.49 (2.25)	6.29 (3.60)
	Msize	14.03 (1.83)	7.03 (4.02)

In the case of no error we see that both the Dantzig selector and Lasso method confirmed our intuition. Notice that the form of A made no difference to either selector as they always were able to produce a model with size nearly identical to the ideal model size with low prediction error. In addition the misclassification rate of

[3]In 3.4 we refer to the chi-squared distribution as noise.

approximately three and a half coupled with the low prediction error shows us that they were able to set all parameters from the chi-squared distribution equal to zero. It is curious to note the near exact recovery performed by the Lasso method when the chi-squared distribution was in the first 10 columns of A. Note that it is not exact recovery due to a non-zero misclassification rate, but the prediction error was brought to essentially zero. This is a curious result because we are unsure as to why this would happen.

While the Lasso and Dantzig selectors perform similarly in the case of no error, there is a large discrepancy once error is added. For the Dantzig selector we see an expected small increase in the prediction error and model size with the addition of error. Note that the increase in model size does not force the Dantzig to choose parameters from the chi-squared distribution but instead has the Dantzig selector just choosing more of the vectors from the true distribution or the linear combination. So just like the case with no noise, all of the parameters chosen are from the same vector space but there are just more chosen when error is present. On the other hand, the Lasso method performs poorly when error is added to the system. To the Lasso method, the addition of error must make the two different distributions look very similar as the Lasso solution has a model sizes greater then 10. Since there are only 5 columns from the true distribution and 5 more columns that share a vector space, when the Lasso selector chooses a model size of 14 it has to be choosing vectors from the noisy chi-squared distribution. It seems that the error has stripped the Lasso method of all of its power to discriminate between distributions. As a result the Lasso solution contains much more error and is much less sparse then the corresponding Dantzig solution.

This simulation study provides us with a situation in which one method clearly outperforms the other. For this case of multi-distributional data, the Dantzig selector outperforms the Lasso method when error is present. This result should not be generalized to include any multi-distributional data scenario since we only tested the situation where the normal and chi-squared distributions are used. However this hopefully will provide the starting point for a more in-depth study on whether or not this result is consistent no matter what the distributions found in the data matrix A.

Chapter 4

Discussion and Future Work

Model selection continues to be a very popular topic in statistics literature due to its real-world applications. In addition, linear regression in the case where $p \gg n$ continues to be a challenge due to the lack of observations. However the emergence of the Lasso selector and its adjustments along with the Dantzig selector has provided an opportunity for accurate estimates for underdetermined systems when x is sparse. These selectors have provided an excellent framework for recovery x when error is not present and accurate estimation of x when error is present in the data. I believe that this topic will continue to gain attention and new adjustments or perhaps even a new selector will soon emerge providing an even better estimates for sparse x.

It should be noted that our study was not attempting to definitively say whether the Lasso or Dantzig selector is a better selector in general. Instead our goal was to isolate specific situations in which one selector outperforms the other so that when that specific situation is present in a data set, researchers will be aware of our results and suggestions. In addition we wished to help show the relationship between the two selectors through our simulation studies and geometric interpretation so that further research can springboard from our results.

Our simulation studies were able to provide some interesting insights into both the Dantzig and Lasso selectors. First it seems that full rank matrices from standard normal distributions must adhere to the Uniform Uncertainty Principle and our condition on the row space of A, as exact recovery is possible in this scenario. Reducing the rank moved the Dantzig selector away from exact recovery potentially implying

that full rank is necessary for exact recovery when working with matrices with standard normal entries. Secondly it should be noted that sparser true solutions lead to more accurate predictions and model sizes closer to the ideal model size in both the Dantzig and the Lasso. Finally we have also seen that the Dantzig selector outperforms the Lasso selector in the case of multi-distributional data when error is present. More research is required to see if this result is consistent over many combinations of distributions, but until then, if a data set can visually be identified or is known to come from two separate distributions then we would suggest using the Dantzig selector over the Lasso method.

Hopefully these simulation studies can help springboard more research in this area of comparing model selection techniques. The initial leg work and coding has now all been written for this project to be extended beyond what is presented in this thesis. This new research could include a larger search for isolated situations in which one selector outperforms the other or it could include more in-depth simulation studies on the characteristics presented in this thesis. In addition we would like to see the model selectors tested on real world data in addition to simulation studies. If a situation can be isolated, we would like to see if real data exists that mirrors our situation and use it to test whether or not the results are the same as in the simulation study.

Another idea for future research that has come out of this thesis is to create a model selector that combines components of both the Dantzig and Lasso methods. This may include creating a new program that uses both the ℓ_1 and ℓ_∞ norms, or it may be a new algorithm that runs both selectors in a certain ordering. The goal is that this new selector combines what each method does well so that it would perform well in a variety of situations.

Finally we would like to see our work involving exact recovery be expanded as well. Currently both our condition involving the row space of A and the Uniform Uncertainty Principle are both computationally inefficient. Therefore for a given matrix A we have no idea whether or not exact recovery is possible. Any improvement to these conditions to make them computationally efficient or a new condition to guarantee exact recovery would be greatly beneficial to this field of research. In our work we observe a specific example in full rank matrices that produces exact recovery and hopefully this result will lead to further investigation of this issue.

Bibliography

[1] H. Akaike. A Bayesian analysis of the minimum AIC procedure. *Ann. Inst. Statist. Math.*, 30:9–14, 1978.

[2] M. Asif and J. Romberg. On the lasso and dantzig selector equivalence. Conference Paper. Conference on Information Sciences and Systems (CISS).

[3] C. Berge. *Principles of Combinatorics*. Academic Press Inc., New York, 1971.

[4] E. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51:4203–4215, 2005.

[5] E. Candes and T. Tao. The dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, 35:2313–2351, 2007.

[6] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal of Scientific Computing*, 20:33–61, 1999.

[7] D. Donoho. For most large underdetermined systems of linear equations the minimal ℓ^1-norm solution is also the sparsest solution. Manuscript, 2004.

[8] D. Donoho and X. Huo. Uncertainty principles and ideal atomic decompostion. *IEEE Transactions on Information Theory*, 47:2845–2862, 2001.

[9] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32:407–499, 2004.

[10] A. E. Horel and R. W. Kennard. Ridge regression: Applications to nonorthogonal problems,. *Technometrics*, 12:69–82, 1970.

[11] A. E. Horel and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[12] R. Horn and C. Johnson. *Matrix Analysis*. Cambridge University Press, New York, 1985.

[13] C. M. Hurvich and C. L. Tsai. Regression and time series model selection in small samples. *Biometrika*, 76:297–307, 1989.

[14] G. James, P. Radchenko, and J. Lv. Dasso: connections between the dantzig selector and lasso. *Journal of Royal Statistical Society, B*, 71:127–142, 2009.

[15] C.K. Li, R. Sze and Y.T. Poon. Solutions of linear equations with special constraints. Private Conversation.

[16] C. L. Mallows. Some comments on C_p. *Technometrics*, 15:661–675, 1973.

[17] N. Meinshausen. Relaxed lasso. *Computational Statistics and Data Analysis*, 52:374–393, 2007.

[18] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

[19] M. Sugiura. Further analysis of the data by Akaike's information criterion and the finite corrections. *Communications in Statistics: Theory and Methods*, 7:13–26, 1978.

[20] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society, B*, 58:267–288, 1996.

[21] H. Zou. The adaptive lasso and its oracle properties. *Journal of American Statistical Association*, 101:1418–1429, 2006.

[22] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of Royal Statistical Society, B*, 67:301–320, 2005.