Electronics and Computer Science
Faculty of Physical and Applied Sciences
University of Southampton

Alejandro Saucedo

29th April, 2014

SPARSE AND GROUP REGRESSION MODELS IN PORTFOLIO OPTIMIZATION

(ANNEX)

Contents

1 Description of Sections

1.1 Background Research

Given the complex nature of this research topic, there was a lot background research required to obtain an intuitive understanding on the models and concepts related. This section aims to provide the reader with the core knowledge required on the financial, mathematical, statistical and algorithmic concepts, definitions and formulations related to this research topic.

Understanding on several financial concepts will be important to provide knowledge on the application of this model. This section provides the mathematical notations and definitions that will be consistent throughout the paper, as well as financial definitions and formulations, such Volatility, Financial Index, Modern Portfolio Theory, Portfolio Optimization, Index Tracking, Value-at-Risk, Conditional-Value-at-Risk, between others.

It will also be important to cover the machine learning and algorithmic models that are present in this research project. These are broken down and analyzed in order to provide an intuitive and clear knowledge. The models used in this paper consist of feature-level regression models (i.e. sum of absolute values, sum of squares, ridge regression (L0+L2-norm) CVaR and Lasso), and sparse-inducing group-level regression models (Group Lasso, Sparse Group Lasso and Multiple Sparse Group Lasso). The algorithmic approaches taken to impose the cardinality constraints in the simple non-sparse models are full-search and greedy algorithms.

1.2 Implementation

The aim of this section is to collect results and observe the behaviour these feature and group level regression models. The main objective in this paper is to discover the effects of taking into account classification information in financial data, as opposed to considering the features as individual entities. We strive to obtain results that can show whether using this extra information will allow for better results.

Each individual regression model is applied to Market Index Data for a specific window of time. Experiments are based completely on the Index Tracking portfolio optimization problem. Our results are quantified on the output Tracking Error, which is equivalent to the test error of our model. In few words, the Index Tracking minimization problem consists on finding a subset of stocks from a specific portfolio that behaves as similar as possible to the original portfolio (i.e. has the lowest tracking error).

Each model contains several constants that need to be adjusted to provide reliable and relevant results – some of these constants define level of sparsity to be achieved, as well as other characteristics. Several experiments will be carried out to find efficient values for these variable to provide reliable results.

Although stochastic data will be used initially, the main financial data set used in this paper has been downloaded using Yahoo finance API and consists of the stocks comprising the FTSE100 (Pronounced 'footsie' 100) – which is a share index of the 100 companies listed on the London Stock Exchange with the highest market capitalization.

The implementation will consist of two main sections – the implementation of feature-level regression models, and the implementation of group-level regression models.

1.3 Analysis

In this section, this paper aims to provide a thorough and in-depth analysis of the results obtained in the implementation section.

The objective of this section is to provide an insight based on the observations drawn from the implementation that would allow for an objective perspective on how using grouping information affects results obtained. In order to provide such insight, this section contains thorough analysis on the behaviour of both sets of models applied in the implementation – feature and group level regression models. Intuition should be provided to the reader through comparing the benefits and drawbacks observed on each of these implementations.

To conclude this section, a real-life application of the implementations in this paper will be given. The objective of this is to provide a more clear understanding on the usefulness of these models in the real financial world. This sub-section should also provide an idea on whether the use of classification information in finance is good choice.

1.4 Expansion

Based in the results and analysis a new idea arose from the rich amount of clustering information available for financial information. It was observed that the Sparse Group Lasso model was limited to only one category. In this section a new concept is proposed to encourage further research in this area of machine learning. The concept is basically a Multiple Sparse Group Model, and as the name implies this concept aims to consider multiple categories during the regression process, allowing for consideration for categorizing characteristics of financial instruments such as type of instrument, sector, industry, volatility, historical return correlation, etc.

1.5 Conclusion

We finalize this paper by providing a brief overview of the results obtained, and the final observations made. This section is very important as it would be ideal that this paper can provide a base, or an addition to current research. Conclusions made in this paper are

2 Expanded Background of Literature Search

2.1 Mathematical Definitions (Redefined)

2.1.1 General

We will use σ to refer to the volatility of a financial instrument (i.e. the standard deviation of the daily returns of a specific financial instrument).

Norms will be referred as Lφ–norms, where $\varphi \in \{1,2, \dots\}$ as $\|\boldsymbol{x}\|_{\varphi} = [\sum_{i=1}^{n} |x_i|^{\varphi}]^{\frac{1}{\varphi}}$.

The statistical and mathematical notations used in this paper are all considered within a specific time window t = 1, ...,T and a number of assets n.

Input data is of the form $\mathbf{R}_t \in \mathbb{R}^n = (\mathbf{R}_{t,1}, \mathbf{R}_{t,2}, \dots, \mathbf{R}_{t,n})^T$ where $\mathbf{R}$ is an $\mathbb{R}^{T\times n}$ matrix where each column is a vector of returns of all the assets of the portfolio at time t.

Our target data is of the form $\boldsymbol{I} = \boldsymbol{R}_t^T \boldsymbol{\pi}'$, where $\boldsymbol{I}$ is a Market Index (i.e. a share index of the n companies listed on a specific Stock Exchange with the highest market capitalization) and $\boldsymbol{\pi} \in \mathbb{R}^n = (\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_n)^T$ is the proportions of each stock $\boldsymbol{\pi}_i$ – in this case $\boldsymbol{\pi}' = 1/n$.

In all our regression problems, $\boldsymbol{\pi}$ will be the parameter to be learned. It is important to note that throughout all the models of this paper, the constraint $\boldsymbol{\pi} > 0$ will always be present, as it enseures that no shortings are done. Finally, as this variable contains percentage values, another constraint that will be held at all times is $\sum_{=0}^{n} \boldsymbol{\pi} = 1$.

2.1.2 Group Notation

When dealing with groups, we will have m groups of size $p_i \in \boldsymbol{p}$, where $\sum_{i=1}^{n} p_i = n$.

Our stocks will be grouped in these m groups of size p_i, which means our data $\mathbf{R}$ is in groups $\overline{\boldsymbol{R}} = (\overline{\boldsymbol{R}}_1, \overline{\boldsymbol{R}}_2, \dots, \overline{\boldsymbol{R}}_m)$, where $\overline{\boldsymbol{R}}_i \subseteq \mathbf{R}$ and $\overline{\boldsymbol{R}}_{i,t} \in \mathbb{R}^{p_i} = (\overline{\boldsymbol{R}}_{i,t,1}, \dots, \overline{\boldsymbol{R}}_{i,t,p_i})^T$.

Although our Index Portfolio would be referenced with the same variable $\mathbf{I}$, a new group definition is now introduced as $\boldsymbol{I} = \overline{\boldsymbol{R}}_1 \overline{\boldsymbol{\pi}}_1 + \overline{\boldsymbol{R}}_2 \overline{\boldsymbol{\pi}}_2 + \cdots + \overline{\boldsymbol{R}}_m \overline{\boldsymbol{\pi}}_m$, where $\overline{\boldsymbol{\pi}}_i \in \mathbb{R}^{p_i}$ are the parameters to learn that belong to the stocks in group $i \in \{1,2, \dots, m\}$.

For the new group formulation proposed we will need to introduce the concept of superscripting (same as in arrays in code) in order to obtain specific elements in matrices and vectors through indexing. In this paper we will use the pseudo-code notation $\boldsymbol{A}[\,\boldsymbol{x},\boldsymbol{y}\,]$, where $\mathbf{X}$ is a matrix of any size, $\mathbf{x}$ and $\mathbf{y}$ are vectors of the same size containing the positions in the Matrix to be superscripted, and they are of the form $\boldsymbol{x_i},\boldsymbol{y_i} \in \mathbb{R} \quad \wedge (\forall x,y.\ |x| < |\boldsymbol{A}[\,x,:\,]| \quad \wedge\ y < |\boldsymbol{A}[\,x,:\,]|) \wedge (\forall x,y.\ \max(x) < |\boldsymbol{A}[\,x,:\,]| \wedge \max(y) < |\boldsymbol{A}[\,x,:\,]|))$. A column can be used to denote that all columns or rows are selected, so $\boldsymbol{A}[\,:,\boldsymbol{y}\,]$ would superscript all the row elements for columns contained in the indexes $\mathbf{y}$.

For indexing we will use a variable $\boldsymbol{\Psi_g} \in \boldsymbol{\psi}$ that will denote the indexes of the stocks of each group. For example, if group $\bar{\boldsymbol{\pi}}_{\boldsymbol{g}}$ contains stocks 3, 5, 6, etc, then $\boldsymbol{\Psi_g} = \{3, 5, 6, \ldots\}$. This allows to introduce the logical biconditional $\left(\bar{\boldsymbol{\pi}}_{\boldsymbol{g}} \subseteq \boldsymbol{\pi}\right) \Leftrightarrow \left(\boldsymbol{\pi}\left[\,1,\boldsymbol{\Psi_g}\,\right] = \bar{\boldsymbol{\pi}}_{\boldsymbol{g}}\right)$.

The final notation required for the new proposed formula is the sum of all the columns or rows as $sum(\boldsymbol{A},1) = \sum_{i=1}^{r} \boldsymbol{A}[\,:,i\,] \wedge sum(\boldsymbol{A},1) = \sum_{i=1}^{c} \boldsymbol{A}[\,i,:\,]$ where r is the number of rows in **A** and c is the number of columns in **A**.

2.2 Financial Definitions

2.2.1 Basic financial terms

- **Financial risk** - potential loss or uncertainty from an investment, and can be measured through mathematical and statistical models.
- **Volatility** - A factor that is often used when measuring financial risk previously defined as σ. The volatility of a financial instrument is the standard deviation of the historical returns.
- **Portfolio –** A group of financial assets such as bonds, stocks where an amount of money is invested
- **Market Index** – An aggregate of stocks from a financial market that aim to represent an entire stock market.

2.2.2 Portfolio Diversification

In finance it is a very important concern to obtain diversification when dealing with investments. What is meant by diversification is to distribute risk by investing in several (preferably uncorrelated) financial instruments. This way, if the price of a group of financial instruments goes down, losses would not be as bad, as a diversified portfolio would have investments in numerous disjoint groups.

2.2.3 Stochastic Simulation Models

In this paper it was a very important concern to find an effective way to simulate financial data. The models that were considered were the GARCH, ARIMA, EULER and a Monte Carlo approach. The approach taken was the Monte Carlo approach

2.2.4 GARCH Prediction Model

This model is derived from the family of AutoRegressive Conditional Hteroskedacity (ARCH) models, and it is used to model time series – hence why it is often applied in finance. In the case of the GARCH model, it is just an ARCH variation which is assumed for the error variance.

2.2.5 Autoregressive Integrated Moving Average

2.2.6 Monte Carlo

The Monte Carlo model is based in a simulation model that is provided by the Matlab core library, and it is based in generating a normally distributed random value that represents the distribution of each stock at each point in time. This model proved to be an accurate and simple approach and this was the reason why this model was chosen.

In matlab, the algorithm is defined as:

Algorithm 1 - Monte Carlo Implementation

```
for i=1:numberStocks
      time = 0;
      volatility = volbase + rand()*volvar;
     prices = [];

      while time < timeToExpiry
         time = time + sampleRate;
         drift = (riskFreeRate - dividend - volatility*volatility/2)*sampleRate;
         perturbation = volatility*sqrt( sampleRate )*randn();
         price = price*exp(drift + perturbation);
         times = [times time];
         prices = [prices, price];
      end

      all_prices = [all_prices; prices];
   end
```

2.2.7 Modern Portfolio Theory

In finance, portfolio optimization is an extremely popular and revisited concept since its debut in the original paper by [1] Markowitz(1952), 'Porfolio Theory'. More widely known as Modern Portfolio Theory, it consists of a minimization on the risk for a given level of expected return **r** through choosing the proportions (weights) of the assets **π** that comprise the portfolio. The formulation introduced in this paper is defined as follows:

Equation 1 – MPT Minimization

$$\min_{\pi} \quad \boldsymbol{\pi}^T \mathbf{R} \boldsymbol{\pi}$$
$$\text{s.t.} \quad \boldsymbol{r}^{\mathrm{T}} \boldsymbol{\pi} = \gamma \quad \sum_{i=1}^{n} \pi_i = 1$$

Using input data over a range of time **T** for **n** assets, this Markowitz portfolio optimization formula minimizes on the portfolio variance given by $\boldsymbol{\pi}^T \mathbf{R} \boldsymbol{\pi}$, where **π** is the proportion invested on each asset, and $\mathbf{R} \in \mathbb{R}^{\mathrm{nxn}}$ is a covariance matrix of asset is the parameter to be learned. The amount of expected return is given by γ, and a vector of expected returns for each asset is given by $\boldsymbol{r} \in \mathbb{R}^{\mathrm{n}}$. This model revolutionized portfolio theory since its debut, and has been thoroughly revisited since.

2.2.8 Tracking Error

In this paper, we will base all our calculations in a very commonly used loss function – the Tracking Error. Tracking error is very intuitive, especially when seen graphically – it basically measures how similar a portfolio A behaves to a portfolio B. We will refer to tracking error as $\xi_{\varphi,t}(\boldsymbol{\pi})$ defined as:

Equation 2 - Tracking Error

$$\xi_{\varphi,t}(\boldsymbol{\pi}) = \frac{1}{T} \sqrt[1/\varphi]{\left[\sum_{t=1}^{T} \left| \boldsymbol{I}_t - \boldsymbol{R}_t^{\mathrm{T}} \boldsymbol{\pi} \right|^{\varphi} \right]}$$

This tracking error function will also be used to measure the results of the accuracy of our model compared to other models proposed.

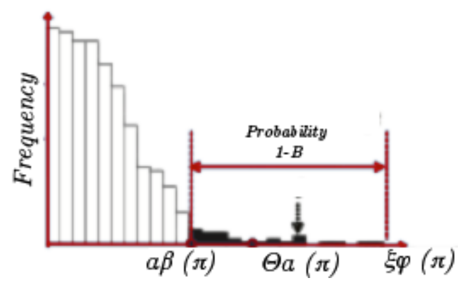
2.3 Background in Regression Applications for Finance

In order to obtain an understanding on how regression models can be effectively implemented in financial datasets, we require to introduce some financial concepts before this, including Value-at-Risk and it's variations.

Figure 1

2.3.1 Value-at-Risk

A very important risk management model in finance is the Value-at-Risk model, or VaR [2] Philippe (1996). The most common application of Value-at-Risk is the β%-VaR. Given a percentage β% and a set of historical prices, we can build a frequency table which we assume has normal distribution, and then we can estimate the loss that we are β% sure that will not be exceeded.



Mathematically, given a confidence index α based on β% (i.e. 95% would be 2 standard deviations), a mean μ and a volatility σ, we can compute the model presented in [6] Mina et al. β%-VaR from a portfolio as follows:

Equation 3 - B%-VaR

$$\beta\%VaR = \alpha * (\sigma - \mu)$$

Value-at-Risk is excellent when dealing with raw calculations from a given volatility, however it lacks of convexity characteristics, making it hard to optimize, and hence unattractive for minimization problems.

2.3.2 Expected Shortfall

β-Conditional-Value-at-Risk, also known as Expected Shortfall, is the conditional expectation of losses above β%-VaR – in other words, the probability that a specific loss will exceed the β%-VaR. Unlike VaR, CVaR has many desired characteristics such as convexity, that will allow for the definitions required for using this as a minimization problem.

To define this, we base ourselves in the mathematical notation in [Akiko 2010] and in the logic of [4] Rockafellar R. et al. (2000). We initially base ourselves in a regression problem to obtain a linear function approximator y = (w,x) + b from m = 1, ..., M samples (x_i, y_i). Our variables $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$ are our input and output values respectively and $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are the variables to be learned.

2.3.3 β-VaR

Once defined our loss function, in this case, Tracking Error, we assume we an get probability of a specific result of $\xi\varphi(\boldsymbol{\pi})$ by computing $\rho(\xi\varphi(\boldsymbol{\pi}))$. This allows us to define a function $\Theta_\alpha(\boldsymbol{\pi})$ that gives us the cumulative distribution for our probability loss function $\rho(\xi\varphi(\boldsymbol{\pi}))$ given a given confidence α. This is, in other words, the probability of $\xi_\varphi(\boldsymbol{\pi})$ not exceeding a threshold given by α. The function $\theta_\alpha(\boldsymbol{\pi})$ is defined as follows:

Equation 4

$$\Theta_\alpha(\boldsymbol{\pi}) = \int_{\rho\left(\xi_{\varphi,t}(\boldsymbol{\pi})\right) \leq \alpha} \rho\left(\xi_{\varphi,t}(\boldsymbol{\pi})\right) \, d\,\xi_{\varphi,t}(\boldsymbol{\pi})$$

Finally, now that we can calculate the cumulative distribution of our loss function $\xi\varphi(\boldsymbol{\pi})$ we can obtain the definition of our β-VaR formula $\alpha_\beta(\boldsymbol{\pi})$ defined as:

Equation 5

$$\alpha_\beta(\boldsymbol{\pi}) = \min_\alpha \; \Theta_\alpha(\boldsymbol{\pi}) \geq \beta$$

Now that we have obtained β-VaR, or the lowest percentage amount that we will be β% sure that will not be exceeded, we can define the CVaR as the average loss **exceeding** VaR. Now that we have reached this definition, we can bring up the proof in [4] Rockafellar R. (2000), which shows the convex nature of CVaR, and allows for a minimization definition as follows:

Equation 6

$$\min_{\boldsymbol{\pi},b,\alpha} \quad \alpha + \frac{1}{(1-\beta)T} \sum_{t=1}^{T} |\xi_{\varphi,t}(\boldsymbol{\pi}) - \alpha|^+$$

Where $|\mathrm{t}|^+ = t$ when t > 0 and $|\mathrm{t}|^+ = 0$ when $\mathrm{t} \leq 0$.

2.4 Feature Regression Models

Now that the necessary formulations were introduced to provide the reader with a core understanding on some of the minimization functions in financial data, as well as an idea on their potential applications we can proceed to discuss the regression models that will be used for single-feature analysis in this paper.

2.4.1 CVaR Minimization

It is proven in Akiko [REFERENCE 1] that this CVaR minimization is equivalent to the Support Vector Regression algorithm, implying optimality, and also includes a proof that allows us to introduce the variable $\mathbf{z_t}$, where $\mathbf{z_t}$ allows us to use the definition of $|x|^+$ by a simple rearrangement of constraints, hence simplify the implementation of the problem in our algorithm massively. This allows us to get our final definition of our CVaR minimization formula as follows:

Equation 7

$$\min_{\boldsymbol{\pi},\alpha,\boldsymbol{z}} \quad \alpha + \frac{1}{(1-\beta)T}\sum_{t=1}^{T} \boldsymbol{z_t}$$

$$\text{s.t.} \qquad \boldsymbol{z_t} - \xi_{\varphi,t}(\boldsymbol{\pi}) + \alpha \geq 0$$

$$\boldsymbol{z_t} \geq 0, \quad i \in \mathrm{T},$$

It is worth bringing up the fact that although the formulation provided in AKIKO [REFERENCE 1], called the Norm Constrained CVaR (NCCVaR) implements a sparsity inducing variable C2 which induces sparsity in the set of weights. Although the NCCVaR formulation won't be used, a lot of the proofs and concepts present in this paper were of great help in order to be able to use the CVaR minimization formula efficiently. This method will be referred to as *CVaR.*

2.4.2 Lasso Regression (Expanded)

Before proceeding to the introduction of the group regression methods, it is necessary to introduce some core concepts in regards to sparsity model – one crucial one is the Lasso Regression model.

The Lasso regression model is comprised of a sum of squares, plus a scaled sum of the absolute value of the parameters – using our variables for Index, Return portfolio and weights, our model would be defined as follows.

Equation 8

$$\min_{\boldsymbol{\pi}} \varrho(\boldsymbol{\pi}) = \sum_{i=1}^{T} \left|\boldsymbol{I}_t - \boldsymbol{R}_{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{\pi}\right| + \lambda\,|\boldsymbol{\pi}|$$

s.t.

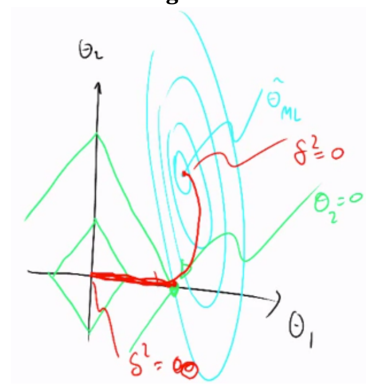
$$\sum_{i=1}^{T} \boldsymbol{\pi_i} = 1, \quad \boldsymbol{\pi_i} > \boldsymbol{0}$$

At first sight, it may seem very similar to the ridge (L0+L2-norm) model introduced previously, however, what makes this model stand out is the second coefficient –

namely the scaled sum of the absolute value of the parameters, which, different from the past model, this coefficient controls the sparsity of our final $\boldsymbol{\pi}$.

To visualize this we'll take n=2 as an example, which is pictured in the figure 3 on the right, where the x axis holds the value for our $\boldsymbol{\pi}_1$ and the y axis holds the value for $\boldsymbol{\pi}_2$.

Figure 2

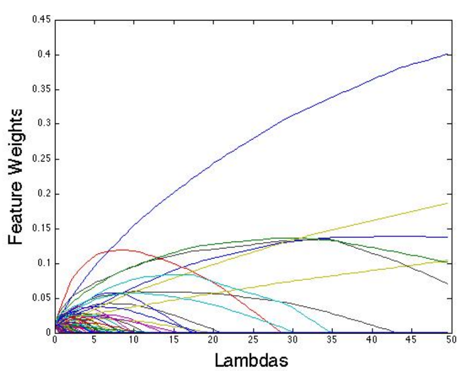


The center of the green diamond represents the point of $\varrho(\boldsymbol{\pi})$ when $\lambda = \infty$, and the center of the blue circle represents the point when $\tau = 0$.

The red path denotes the value $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ take as τ changes value. It can be observed that when $\tau = 0$, θ_2 will be equal to the maximum likelihood of $\boldsymbol{\pi}_2$, denoted in blue by $\hat{\boldsymbol{\pi}}_{ML}$ which is the result of the first term of our equation. When $\tau = \infty$, the values of $\boldsymbol{\pi}$ will need to be as small as possible in order to minimize our cost function, which will result in our point in the origin.

By differentiating our cost function and equating it to zero, we find that many of our variables become zero – this is shown in Figure 2

Figure 3



Going back to figure 3 we can observe this behaviour when our blue circle intersects the corner of the green diamond. It is proven that intersections between these two are very likely to happen in the corners – which implies that one of our parameters has taken the value of zero – in this example, our parameter θ_2 takes the value of zero when the red line intersects the x axis.

Intuitively, we will build an algorithm that initially takes a guess for all our parameters, then it will proceed to calculate the optimal value of each of the parameters using the partial derivative of our cost function, and repeat this until it converges.

Equation 9

$$\frac{\partial J(\theta)}{\partial \theta_j} = 2\sum_{i=1}^{m} x^2\theta_j - 2\sum_{i=0}^{m}(y_i - x_{i-j}\theta_j)x_{ij} + \frac{\partial \delta}{\partial \theta_j}|\theta_j|$$

The reason why the last term has not been derived yet is because it consists of an absolute value – for this, subdifferentials will be used. In order to simplify the formulation shown, we will make the following assignments.

Equation 10

$$a_j = 2\sum_{i=1}^{m} x^2$$

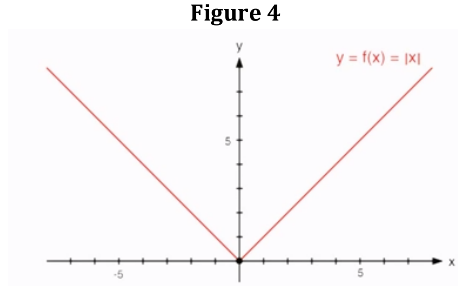
$$c_j = 2\sum_{i=0}^{m} (y_i - x_{i-j}\theta_j)x_{ij}$$

This would give us a much simpler formula to expand on its subdifferential, which consist of the following:

Equation 11

$$\frac{\partial\, J(\theta)}{\partial\, \theta_j} = a_j\theta_j + c_j + \frac{\partial\delta}{\partial\theta_j}|\theta_j|$$

Figure 4



Basically, with a subdifferential, we have a point which cannot be differentiated normally, as we would require a single gradient to explain a subset of gradients – in Figure 3, our point would be the origin, and we would have to define the differential of our formula as:

Equation 12

$$f(x) = |x|$$

$$\partial f(x) = \begin{cases} \{-1\} & \text{if } x < 0 \\ [-1, 1] & \text{if } x = 0 \\ \{+1\} & \text{if } x > 0 \end{cases}$$

So, our formulas would consist of exactly the same format, but instead we would use our variables a, b and θ_j as follows:

Equation 13

$$= \begin{cases} \{a_j\theta_j - c_j - \delta^2\} & \text{if } \theta_j < 0 \\ [-c_j - \delta^2, -c_j + \delta^2] & \text{if } \theta_j = 0 \\ \{a_j\theta_j - c_j + \delta^2\} & \text{if } \theta_j > 0 \end{cases}$$

Now, by setting our values to zero, our estimate $\hat{\theta}$ can be calculated by setting these differentials to zero for each of the θ_j, which gives us the following estimates:

Equation 14

$$\widehat{\theta}_j = \begin{cases} (c_j + \delta^2)/a_j & \text{if } c_j < -\delta^2 \\ 0 & \text{if } c_j \in [-\delta^2, \delta^2] \\ (c_j - \delta^2)/a_j & \text{if } c_j > \delta^2 \end{cases}$$

Knowing this estimate now allows us to implement the simple Lasso Algorithm in a very simple manner:

Algorithm 2 - Lasso PseudoCode

1. **Initialize** θ (Can be either random, or with something like ridge, etc)
2. **Repeat** until converged
   a. **For** j = 1,2, … , d DO
      i. $a_j = 2\sum_{i=1}^{m} x^2$
      ii. $c_j = 2\sum_{i=0}^{m} (y_i - x_{i-j}\theta_j)x_{ij}$
      iii. **if** $(c_j < -\delta^2)$
         1. $\theta_j = (c_j + \delta^2)/a_j$
      iv. **elseif** $(c_j > -\delta^2)$
         1. $\theta_j = (c_j - \delta^2)/a_j$
      v. **else**
         1. $\theta_j = 0$

2.5 Model Selection Approaches (Expanded)

2.5.1.1 Forward Search Application (Expanded)

Due to the exponential complexity of the problem in [0] Mahesan et al. (2013) (i.e. to find an optimal subset of stocks from a given Market Index), it was very limited when it comes to even relatively small portfolios. Given that the average Market Index is composed of at least 100 distinct assets, this makes the implementation of a full search is almost impossible.

For this reason, a greedy approach was taken - namely, this was done through a forward model selection algorithm, where one optimal stock is chosen and added to the subset on each step until the subset reaches the desired cardinality.

Formally, the model is formulated as a subset selection, where each step, a locally optimal asset is chosen and added to the portfolio. The algorithm proposed in [0] Mahesan et al. (2013) is converted into a subset selection problem defined as follows:

Algorithm 1 Greedy Algorithm

Initialize: $S_0 = \emptyset$ and $k = 1$. Set $f^\tau(\widetilde{\boldsymbol{\pi}}^\tau_{S_0})$ to a value large enough.
while $k \leq C_0$ **do**
  For all $s \in N \setminus S_{k-1}$, compute $\widetilde{\boldsymbol{\pi}}^\tau_{S_{k-1}\cup\{s\}}$
  Select s^* such that $\min_{s\in N\setminus S_{k-1}} f^\tau(\widetilde{\boldsymbol{\pi}}^\tau_{S_{k-1}\cup\{s\}})$ and set $\widetilde{\boldsymbol{\pi}}^\tau_{S_k} \leftarrow \widetilde{\boldsymbol{\pi}}^\tau_{S_{k-1}\cup\{s^*\}}$.
  Set $S_k \leftarrow S_{k-1} \cup \{s^*\}$ and $k \leftarrow k+1$.
end while
$\boldsymbol{\pi}^\tau_G \leftarrow \widetilde{\boldsymbol{\pi}}^\tau_{S_{C_0}}$ and $S^\tau_G \leftarrow S_{C_0}$.
return $\boldsymbol{\pi}^\tau_G$ and S^τ_G.

To apply this algorithm, we begin with an empty set, then for all the available stocks, we add one and run the L0+L1-norm model on each. We then calculate the tracking error for each, and select the asset with the lowest tracking error as our local optimal. We do this until the cardinality of our set reaches the L-0 constraint C_0. This greedy forward search algorithm will be referred to as GFS throughout this paper.

2.6 Group Regression Approaches

2.6.1 Model Selection Approach (Expanded)

Before considering complex group regression models, this paper considers an approach based in the approach in [0] Mahesan et al. (2013) that initially inspired this research project – mainly it is being referred to the forward-search approach that was taken to select a subset of stocks from a Market Index with the lowest tracking error.

As it was revisited in the last section, the approach taken in [0] was a greedy, forward-search approach, and this was the choice over a full-search approach due to the NP characteristics of the subset selection problem. Given that stocks are considered individually, and the Market Indexes vary from 100 to 500 stocks, the time complexity of this problem would be exponential on the number of stocks to consider, which would not allow for a full-search approach.

This section proposes to reduce this individual stock subset selection problem reduces to a group subset selection problem, and assuming that our group number does not consists of a large amount of groups (e.g. more than 20), a full-search approach is actually possible, as we could have a full subset selection on our groups.

Similar to the previous algorithm, we can select any regression model and implement a full-search group subset selection as follows:

Algorithm 1 Greedy Algorithm

Initialize: $S_0 = \emptyset$ and $k = 1$. Set $f^\tau(\widetilde{\boldsymbol{\pi}}^\tau_{S_0})$ to a value large enough.
while $k \leq C_0$ **do**
  For all $s \in N \setminus S_{k-1}$, compute $\widetilde{\boldsymbol{\pi}}^\tau_{S_{k-1}\cup\{s\}}$
  Select s^* such that $\min_{s\in N\setminus S_{k-1}} f^\tau(\widetilde{\boldsymbol{\pi}}^\tau_{S_{k-1}\cup\{s\}})$ and set $\widetilde{\boldsymbol{\pi}}^\tau_{S_k} \leftarrow \widetilde{\boldsymbol{\pi}}^\tau_{S_{k-1}\cup\{s^*\}}$.
  Set $S_k \leftarrow S_{k-1} \cup \{s^*\}$ and $k \leftarrow k+1$.
end while
$\boldsymbol{\pi}^\tau_G \leftarrow \widetilde{\boldsymbol{\pi}}^\tau_{S_{C_0}}$ and $S^\tau_G \leftarrow S_{C_0}$.
return $\boldsymbol{\pi}^\tau_G$ and S^τ_G.

Similar to results presented in [0], with this full-search algorithm, we would be able to select a subset of financial instruments from a market index based on prior knowledge on the instruments that allowed us to create the groupings.

2.7 Zero-Constrained Models

2.7.1 Zero-Constrained Group Lasso Implementation

Here we present the implementation of the Zero-Constrained Group Lasso implementation which was defined as follows:

$$\min_{\overline{\boldsymbol{\pi}}\in\,\mathbb{R}^{\boldsymbol{n}}} \;\; \left\|\boldsymbol{I} - \boldsymbol{R}^{\mathrm{T}} sum(\widehat{\overline{\boldsymbol{\pi}}},2)\right\| + \lambda * sum\left(\sqrt{p^{\mathrm{T}}} * \sqrt{sum(\widehat{\overline{\boldsymbol{\pi}}}.\,\hat{}\,2,2)}\right)$$

$$\text{s.t.} \qquad \boldsymbol{\pi_i} > \boldsymbol{0} \quad \wedge \quad \widehat{\overline{\boldsymbol{\pi}}}\left[\,\sim\!\boldsymbol{\Psi_g}\,, g\,\right] = 0$$

The implementation was carried out with the help of the CVX library. It receives as input a matrix 'not_group_idx' where each row represents group g of rows containing 1's in the indexes of the stocks that are **not** contained in group $\overline{\boldsymbol{\pi}}_g$, hence not_group_idx is our $\sim\!\boldsymbol{\Psi_g}$ variable defined previously. It also receives as input the size of the groups as a vector, which represents our variable $\boldsymbol{p}$. Finally, the last argument taken is the lambda coefficient, which determines how much the model will be inducing group sparsity.

With this, our output will be pimat_hat, which represents the variable introduced earlier $\widehat{\overline{\boldsymbol{\pi}}}$ where $\widehat{\overline{\boldsymbol{\pi}}}\left[\,\sim\!\boldsymbol{\Psi_g}\,, g\,\right] = 0$ and $sum\left(\widehat{\overline{\boldsymbol{\pi}}},2\right) = \boldsymbol{\pi}$.

With this, the GroupLasso function has been implemented in Matlab as follows:

```
Function [pimat_hat] = GroupLasso(not_group_idx, group_sizes, lambda)
group_sizes = sqrt(group_sizes);
cvx_begin
    variable pimat_hat (totalassets,n)

    minimize(sum_square_pos(norm(I_train - R_train'* sum(pimat_hat,2)))
                                        + lambda*norms(pimat_hat,2,1)*sqr_group_sizes)

    subject to
        pimat_hat >= 0
        pimat_hat(not_group_idx) == 0
cvx_end
end
```

2.7.2 Zero-Constrained Sparse Group Lasso Implementation

This definition can be easily expanded into the sparse group formulation by taking the Zero-Constrained Group Lasso definition and expanding it into the Zero-Constrained Group Lasso Implementation, which we define as follows:

$$\min_{\widehat{\boldsymbol{\pi}} \in \mathbb{R}^{\boldsymbol{n}}} \quad \left\| \boldsymbol{I} - \boldsymbol{R}^{\mathrm{T}} sum(\widehat{\widehat{\boldsymbol{\pi}}}, 2) \right\| + \lambda_1 \; sum\left(\sqrt{p^{\mathrm{T}}} * \sqrt{sum(\widehat{\widehat{\boldsymbol{\pi}}}.\,\hat{}\,2{,}2)} \right) + \lambda_2 \left\| sum(\widehat{\widehat{\boldsymbol{\pi}}}, 2) \right\|_{\mathbf{1}}$$

$$\text{s.t.} \qquad \boldsymbol{\pi_i} > \mathbf{0} \quad \wedge \quad \widehat{\widehat{\boldsymbol{\pi}}}\left[\sim\boldsymbol{\Psi_g} \, , g \, \right] = 0$$

The only difference in this version is that in this case, the 'lambda' parameter is replaced with two new parameters – namely 'l1_groups', which controls between-group sparsity (Group Lasso characteristic), and 'l2_groups' which controls within group lasso sparsity (Simple Lasso characteristic). The SparseGroupLasso function was implemented in Matlab code as follows:

```
Function [pimat_hat] = SparseGroupLasso(not_group_idx, group_sizes, l1_groups, l2_features)
cvx_begin
    variable pimat_hat (totalassets,n)

    minimize(sum_square_pos(norm(I_train - sum(R_train'* pimat_hat,2)))
                            + l1_groups*norms(pimat_hat,2,1)*sqr_group_sizes
                            + l2_features*sum(abs(pimat_hat (:))) )

    subject to
        pimat_hat >= 0
        pimat_hat(not_group_idx) == 0
cvx_end
end
```

3 Alternative Approaches

In this brief subsection, alternatives to the CVX libraries are briefly discussed – Namely the SPAMS and [REFERENCE] libraries.

These two libraries were not implemented in this paper – nevertheless they provided several core concepts that were used, and the research that accompanies these libraries is present in several sections in this paper – including [REFERENCE] and [REFERENCE].

Figure [Figure] on the right shows the results obtained when implementing the Group Lasso Model previously. As it can be observed, the results are notably less accurate than the ones obtained with the CVX library - the reason for this is due to the constraints needed for our implementation – these libraries proved to loose accuracy when implemented with a positivity constraint on the weights.

Lambda	Zeros	Error
0	0	0.0267
0.05	47	2.6464
0.1	47	5.2671
0.15	47	7.8959
0.2	47	10.5329
0.25	72	10.9667
0.3	47	15.8043
0.35	47	18.4401
0.4	47	21.0767
0.45	72	20.7983
0.5	72	23.6749
0.55	72	27.4671
0.6	72	28.8844
0.65	72	30.8535
0.7	47	36.8924
0.75	72	34.8185
0.8	72	38.3813
0.85	47	44.8011
0.9	72	45.7797
0.95	72	44.3252
1	72	48.2338

4 Tracking Error (All Data)

4.1 Feature Selection

	Market				Stochastic				Market				Stochastic				
Zeros	**Abs**	**Sq.**	**Ridge**	**CVaR**	**Abs**	**Sq.**	**Ridge**	**CVaR**	**Abs**	**Squares**	**Ridge**	**CVaR**	**Abs**	**Sq.**	**Ridge**	**CVaR**	
0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**50**	0.055	0.048	0.032	0.046	0.236	0.189	0.169	0.229
1	0.007	0.002	0.006	0.002	0.013	0.036	0.029	0.073	**51**	0.054	0.047	0.033	0.044	0.237	0.187	0.169	0.228
2	0.009	0.007	0.009	0.006	0.087	0.051	0.043	0.117	**52**	0.055	0.047	0.032	0.047	0.239	0.187	0.171	0.228
3	0.012	0.010	0.011	0.012	0.119	0.074	0.056	0.121	**53**	0.054	0.047	0.033	0.046	0.241	0.187	0.177	0.227
4	0.014	0.011	0.012	0.013	0.120	0.085	0.073	0.125	**54**	0.054	0.049	0.032	0.046	0.234	0.186	0.177	0.229
5	0.015	0.013	0.013	0.015	0.130	0.089	0.080	0.128	**55**	0.052	0.050	0.033	0.049	0.234	0.185	0.174	0.231
6	0.018	0.019	0.013	0.017	0.134	0.098	0.090	0.130	**56**	0.058	0.050	0.034	0.052	0.234	0.184	0.176	0.233
7	0.021	0.024	0.013	0.019	0.145	0.101	0.094	0.142	**57**	0.061	0.049	0.033	0.055	0.243	0.184	0.183	0.242
8	0.024	0.025	0.013	0.019	0.153	0.100	0.095	0.146	**58**	0.060	0.048	0.035	0.056	0.237	0.183	0.189	0.237
9	0.024	0.025	0.013	0.021	0.169	0.112	0.099	0.165	**59**	0.060	0.048	0.034	0.052	0.238	0.183	0.198	0.248
10	0.025	0.027	0.014	0.022	0.176	0.121	0.109	0.174	**60**	0.064	0.048	0.036	0.053	0.235	0.183	0.203	0.249
11	0.026	0.028	0.014	0.023	0.181	0.116	0.112	0.176	**61**	0.065	0.049	0.035	0.049	0.234	0.183	0.207	0.248
12	0.028	0.029	0.014	0.024	0.177	0.117	0.116	0.200	**62**	0.064	0.049	0.038	0.051	0.230	0.187	0.213	0.247
13	0.026	0.029	0.015	0.026	0.175	0.119	0.118	0.213	**63**	0.067	0.050	0.038	0.052	0.227	0.191	0.216	0.245
14	0.026	0.028	0.015	0.027	0.175	0.130	0.116	0.202	**64**	0.068	0.050	0.038	0.048	0.227	0.192	0.217	0.243
15	0.029	0.028	0.016	0.025	0.174	0.136	0.120	0.207	**65**	0.067	0.050	0.038	0.047	0.227	0.197	0.221	0.243
16	0.031	0.029	0.017	0.025	0.198	0.135	0.125	0.188	**66**	0.064	0.050	0.040	0.048	0.227	0.197	0.232	0.247
17	0.033	0.030	0.019	0.026	0.187	0.148	0.131	0.198	**67**	0.065	0.050	0.039	0.047	0.227	0.197	0.238	0.248
18	0.030	0.030	0.020	0.026	0.203	0.153	0.135	0.191	**68**	0.068	0.048	0.041	0.045	0.227	0.197	0.246	0.255
19	0.034	0.033	0.019	0.026	0.209	0.155	0.138	0.194	**69**	0.071	0.047	0.042	0.047	0.227	0.197	0.247	0.248
20	0.036	0.034	0.020	0.028	0.208	0.151	0.139	0.191	**70**	0.074	0.047	0.043	0.055	0.227	0.197	0.249	0.248
21	0.034	0.033	0.020	0.027	0.214	0.152	0.142	0.180	**71**	0.074	0.048	0.045	0.055	0.227	0.197	0.255	0.252
22	0.034	0.033	0.020	0.032	0.213	0.164	0.144	0.179	**72**	0.074	0.051	0.046	0.055	0.227	0.198	0.261	0.253
23	0.032	0.032	0.022	0.033	0.220	0.166	0.153	0.174	**73**	0.074	0.050	0.046	0.057	0.227	0.199	0.265	0.252
24	0.036	0.032	0.021	0.029	0.226	0.162	0.151	0.201	**74**	0.078	0.052	0.048	0.060	0.227	0.204	0.278	0.252
25	0.037	0.034	0.023	0.031	0.231	0.158	0.152	0.201	**75**	0.077	0.053	0.048	0.062	0.227	0.208	0.285	0.252
26	0.037	0.035	0.023	0.033	0.219	0.161	0.153	0.191	**76**	0.079	0.053	0.049	0.063	0.227	0.218	0.287	0.252
27	0.035	0.034	0.024	0.032	0.212	0.167	0.164	0.195	**77**	0.077	0.055	0.051	0.068	0.228	0.220	0.291	0.252
28	0.033	0.034	0.024	0.032	0.208	0.175	0.166	0.211	**78**	0.077	0.058	0.052	0.074	0.232	0.224	0.291	0.252
29	0.034	0.037	0.024	0.035	0.210	0.176	0.162	0.210	**79**	0.079	0.060	0.052	0.074	0.244	0.228	0.288	0.252
30	0.033	0.036	0.026	0.039	0.218	0.174	0.159	0.211	**80**	0.078	0.064	0.056	0.077	0.245	0.232	0.287	0.252
31	0.036	0.037	0.027	0.040	0.226	0.174	0.161	0.206	**81**	0.082	0.067	0.059	0.082	0.255	0.235	0.291	0.252
32	0.038	0.038	0.028	0.043	0.209	0.175	0.161	0.213	**82**	0.085	0.070	0.061	0.081	0.260	0.240	0.294	0.259
33	0.040	0.040	0.028	0.039	0.219	0.173	0.163	0.215	**83**	0.091	0.074	0.061	0.079	0.265	0.244	0.291	0.263
34	0.038	0.038	0.029	0.041	0.219	0.172	0.168	0.212	**84**	0.090	0.076	0.071	0.083	0.274	0.249	0.292	0.269
35	0.038	0.037	0.030	0.036	0.220	0.170	0.167	0.209	**85**	0.096	0.081	0.072	0.083	0.277	0.257	0.302	0.279
36	0.043	0.039	0.030	0.034	0.203	0.170	0.165	0.206	**86**	0.099	0.084	0.078	0.087	0.288	0.264	0.306	0.288
37	0.044	0.041	0.031	0.040	0.205	0.175	0.168	0.198	**87**	0.099	0.090	0.084	0.088	0.293	0.278	0.313	0.302
38	0.043	0.041	0.032	0.040	0.193	0.173	0.169	0.208	**88**	0.101	0.096	0.087	0.089	0.299	0.292	0.317	0.312
39	0.046	0.042	0.032	0.040	0.193	0.176	0.168	0.206	**89**	0.106	0.102	0.095	0.096	0.317	0.307	0.325	0.319
40	0.045	0.042	0.032	0.041	0.185	0.190	0.167	0.212	**90**	0.109	0.111	0.101	0.109	0.332	0.321	0.354	0.332
41	0.043	0.043	0.032	0.041	0.185	0.191	0.166	0.198	**91**	0.113	0.121	0.103	0.122	0.346	0.336	0.372	0.347
42	0.044	0.045	0.032	0.044	0.190	0.189	0.164	0.198	**92**	0.124	0.131	0.112	0.137	0.370	0.356	0.400	0.359
43	0.046	0.046	0.032	0.042	0.198	0.187	0.167	0.206	**93**	0.139	0.146	0.120	0.150	0.383	0.372	0.408	0.377
44	0.044	0.047	0.032	0.042	0.211	0.186	0.166	0.222	**94**	0.163	0.164	0.143	0.169	0.400	0.388	0.417	0.397
45	0.045	0.048	0.031	0.043	0.217	0.186	0.168	0.231	**95**	0.189	0.188	0.182	0.188	0.416	0.414	0.444	0.420
46	0.046	0.047	0.032	0.044	0.237	0.186	0.170	0.238	**96**	0.224	0.223	0.227	0.222	0.431	0.432	0.455	0.440
47	0.048	0.046	0.032	0.044	0.232	0.187	0.172	0.244	**97**	0.282	0.282	0.285	0.282	0.453	0.459	0.479	0.467
48	0.050	0.046	0.031	0.047	0.232	0.191	0.173	0.233	**98**	0.415	0.416	0.415	0.415	0.482	0.483	0.499	0.484

4.2 Lasso and Group Lasso

	Market		Stochastic	
Lambda	**Zeros**	**Error**	**Zeros**	**Error**
0	0	0.000	0	0.000
1	31	0.056	22	0.071
1	42	0.075	39	0.091
2	54	0.088	46	0.105
2	59	0.096	51	0.120
3	63	0.103	57	0.135
3	65	0.108	60	0.148
4	68	0.114	64	0.159
4	69	0.119	66	0.167
5	71	0.124	67	0.177
5	74	0.130	69	0.188
6	76	0.133	70	0.199
6	77	0.137	73	0.210
7	78	0.141	73	0.221
7	78	0.144	73	0.233
8	79	0.148	74	0.244
8	80	0.151	74	0.256
9	80	0.155	75	0.269
9	80	0.160	75	0.281
10	81	0.166	75	0.293
10	81	0.172	76	0.306
11	83	0.176	76	0.318
11	83	0.181	77	0.331
12	84	0.186	78	0.344
12	84	0.191	79	0.357
13	84	0.195	80	0.369
13	85	0.200	82	0.381
14	86	0.203	82	0.393
14	86	0.206	83	0.405
15	86	0.210	84	0.417
15	86	0.213	85	0.428
16	86	0.217	88	0.437
16	86	0.220	88	0.445
17	87	0.224	88	0.452
17	89	0.228	88	0.460
18	89	0.232	90	0.467
18	89	0.235	90	0.474
19	89	0.239	90	0.481
19	89	0.243	90	0.488
20	89	0.247	90	0.495
20	90	0.251	90	0.502
21	90	0.254	91	0.509
21	91	0.255	91	0.516
22	91	0.257	91	0.522
22	91	0.258	91	0.529
23	91	0.260	91	0.537
23	91	0.262	91	0.544
24	91	0.264	91	0.551
24	91	0.266	91	0.558
25	91	0.268	91	0.565

	Market (Sectors)		Market (Spectral)		Stochastic (Spectral)	
Lambda-1	**Zeros**	**Error**	**Zeros**	**Error**	**Zeros**	**Error**
0	0	0.000	0	0.000	0	0.000
1	0	0.013	0	0.014	22	0.071
1	0	0.026	2	0.026	39	0.091
2	0	0.039	4	0.036	46	0.105
2	21	0.050	4	0.046	51	0.120
3	21	0.058	4	0.056	57	0.135
3	21	0.069	21	0.062	60	0.148
4	21	0.079	21	0.067	64	0.159
4	41	0.087	33	0.070	66	0.167
5	41	0.092	33	0.072	67	0.177
5	56	0.096	33	0.075	69	0.188
6	56	0.098	33	0.077	70	0.199
6	56	0.101	33	0.080	73	0.210
7	65	0.104	33	0.082	73	0.221
7	65	0.106	33	0.085	73	0.233
8	65	0.108	33	0.088	74	0.244
8	65	0.111	33	0.091	74	0.256
9	65	0.113	33	0.095	75	0.269
9	65	0.116	33	0.098	75	0.281
10	65	0.119	33	0.101	75	0.293
10	65	0.122	33	0.105	76	0.306
11	65	0.126	33	0.108	76	0.318
11	65	0.129	33	0.111	77	0.331
12	65	0.132	33	0.115	78	0.344
12	65	0.135	33	0.119	79	0.357
13	65	0.139	33	0.122	80	0.369
13	65	0.142	33	0.126	82	0.381
14	65	0.146	33	0.130	82	0.393
14	65	0.150	33	0.133	83	0.405
15	65	0.154	33	0.137	84	0.417
15	84	0.157	33	0.141	85	0.428
16	84	0.158	33	0.145	88	0.437
16	84	0.159	51	0.148	88	0.445
17	84	0.160	51	0.151	88	0.452
17	84	0.161	51	0.155	88	0.460
18	84	0.162	51	0.158	90	0.467
18	84	0.163	51	0.162	90	0.474
19	84	0.164	51	0.165	90	0.481
19	84	0.165	51	0.169	90	0.488
20	84	0.166	51	0.172	90	0.495
20	84	0.167	51	0.176	90	0.502
21	84	0.169	51	0.179	91	0.509
21	84	0.170	51	0.183	91	0.516
22	84	0.172	51	0.186	91	0.522
22	84	0.173	51	0.190	91	0.529
23	84	0.175	51	0.194	91	0.537
23	84	0.177	51	0.197	91	0.544
24	84	0.178	51	0.201	91	0.551
24	84	0.180	51	0.205	91	0.558
25	84	0.182	51	0.208	91	0.565

4.3 Sparse Group Lasso by Dataset

5000

		Market (Sectors)		Market (Spectral)		Stochastic (Spectral)	
	Lambda-2	**Lambda-1**	**Error**	**Zeros**	**Error**	**Zeros**	**Error**
0	**0**	0	0.000	0	0.000	0	0.000
0	**1**	9	0.035	8	0.034	7	0.050
0	**1**	18	0.049	18	0.050	18	0.069
0	**2**	25	0.058	23	0.058	26	0.080
0	**2**	28	0.066	28	0.066	31	0.088
0	**3**	33	0.074	33	0.074	38	0.096
0	**3**	35	0.080	36	0.080	43	0.102
0	**4**	39	0.084	37	0.085	45	0.107
0	**4**	43	0.089	42	0.089	46	0.113
0	**5**	44	0.093	46	0.094	46	0.119
0	**5**	47	0.097	50	0.098	48	0.126
0	**6**	47	0.101	50	0.102	50	0.132
0	**6**	50	0.106	50	0.106	51	0.139
0	**7**	50	0.110	51	0.110	52	0.146
0	**7**	53	0.113	54	0.114	55	0.152
0	**8**	56	0.116	56	0.118	58	0.158
0	**8**	61	0.119	59	0.121	58	0.165
0	**9**	61	0.121	60	0.125	58	0.172
0	**9**	62	0.122	61	0.128	60	0.178
0	**10**	62	0.124	62	0.131	61	0.185
0	**10**	62	0.125	63	0.134	61	0.191
0	**11**	64	0.127	64	0.136	63	0.197
0	**11**	64	0.128	64	0.138	63	0.204
0	**12**	64	0.130	68	0.140	63	0.210
0	**12**	65	0.131	69	0.142	63	0.216
0	**13**	65	0.133	69	0.144	63	0.222
0	**13**	68	0.134	69	0.146	63	0.228
0	**14**	68	0.135	69	0.149	63	0.234
0	**14**	70	0.137	69	0.151	64	0.240
0	**15**	71	0.138	74	0.152	64	0.246
0	**15**	72	0.139	74	0.154	64	0.253
0	**16**	72	0.141	74	0.155	64	0.259
0	**16**	72	0.142	74	0.156	64	0.265
0	**17**	72	0.143	74	0.157	64	0.270
0	**17**	72	0.145	74	0.158	65	0.276
0	**18**	74	0.146	75	0.160	66	0.282
0	**18**	74	0.147	76	0.161	68	0.288
0	**19**	74	0.148	77	0.162	69	0.294
0	**19**	74	0.149	77	0.163	69	0.299
0	**20**	78	0.150	77	0.165	70	0.305
0	**20**	79	0.151	77	0.166	70	0.310
0	**21**	79	0.152	77	0.167	70	0.315
0	**21**	80	0.152	77	0.169	70	0.320
0	**22**	80	0.153	77	0.170	70	0.325
0	**22**	80	0.153	77	0.171	70	0.331
0	**23**	80	0.154	78	0.173	70	0.336
0	**23**	81	0.155	80	0.174	70	0.341
0	**24**	81	0.155	82	0.175	70	0.347
0	**24**	81	0.156	82	0.176	70	0.352

10000

		Market (Sectors)		Market (Spectral)		Stochastic (Spectral)	
Lambda-1	**Lambda-2**	**Zeros**	**Error**	**Zeros**	**Error**	**Zeros**	**Error**
0	**0**	0	0.000	0	0.000	0	0.000
0	**1**	13	0.039	12	0.039	9	0.053
0	**1**	26	0.055	27	0.054	22	0.072
0	**2**	31	0.064	31	0.063	30	0.083
0	**2**	36	0.073	37	0.073	34	0.092
0	**3**	42	0.082	44	0.082	45	0.100
0	**3**	47	0.087	47	0.087	47	0.106
0	**4**	49	0.091	52	0.091	48	0.113
0	**4**	54	0.095	54	0.095	50	0.120
0	**5**	55	0.100	56	0.100	51	0.128
0	**5**	60	0.104	60	0.104	53	0.135
0	**6**	60	0.107	60	0.107	56	0.142
0	**6**	60	0.110	60	0.110	59	0.149
0	**7**	61	0.113	63	0.114	59	0.155
0	**7**	62	0.116	63	0.117	62	0.162
0	**8**	63	0.119	67	0.119	63	0.169
0	**8**	67	0.122	67	0.122	63	0.175
0	**9**	67	0.124	67	0.125	63	0.182
0	**9**	68	0.126	67	0.127	64	0.189
0	**10**	69	0.129	69	0.130	64	0.196
0	**10**	69	0.131	69	0.133	64	0.203
0	**11**	69	0.133	70	0.136	66	0.209
0	**11**	72	0.135	71	0.138	67	0.216
0	**12**	72	0.137	72	0.141	67	0.222
0	**12**	73	0.138	73	0.143	68	0.229
0	**13**	73	0.140	74	0.145	68	0.235
0	**13**	74	0.141	74	0.146	70	0.242
0	**14**	75	0.142	74	0.148	70	0.248
0	**14**	76	0.142	74	0.150	71	0.255
0	**15**	76	0.143	74	0.151	71	0.261
0	**15**	77	0.144	75	0.154	71	0.267
0	**16**	77	0.145	75	0.156	72	0.273
0	**16**	78	0.145	76	0.158	72	0.279
0	**17**	78	0.146	77	0.160	72	0.285
0	**17**	78	0.147	77	0.163	72	0.291
0	**18**	78	0.148	77	0.165	72	0.297
0	**18**	78	0.149	77	0.168	73	0.303
0	**19**	79	0.150	77	0.170	74	0.309
0	**19**	79	0.152	80	0.172	75	0.315
0	**20**	79	0.153	80	0.174	75	0.321
0	**20**	79	0.155	80	0.176	75	0.327
0	**21**	79	0.157	80	0.177	75	0.333
0	**21**	79	0.158	80	0.179	75	0.339
0	**22**	79	0.160	81	0.182	75	0.345
0	**22**	79	0.163	81	0.184	75	0.351
0	**23**	82	0.165	81	0.186	76	0.358
0	**23**	82	0.166	81	0.188	76	0.364
0	**24**	82	0.168	81	0.190	76	0.371
0	**24**	82	0.171	81	0.192	76	0.377

4.4 Sparse Group Lasso - by Dataset (Cont...)

Market (Sectors)	500		1000		5000		10000
Zeros	Error	Zeros	Error	Zeros	Error	Zeros	Error
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	0.015	0	0.020	9	0.035	13	0.039
0	0.023	0	0.029	18	0.049	26	0.055
0	0.029	1	0.037	25	0.058	31	0.064
0	0.036	1	0.044	28	0.066	36	0.073
0	0.043	1	0.050	33	0.074	42	0.082
0	0.049	3	0.056	35	0.080	47	0.087
0	0.056	3	0.062	39	0.084	49	0.091
21	0.061	5	0.068	43	0.089	54	0.095
22	0.066	22	0.073	44	0.093	55	0.100
22	0.071	23	0.077	47	0.097	60	0.104
22	0.075	28	0.082	47	0.101	60	0.107
22	0.080	28	0.086	50	0.106	60	0.110
28	0.085	28	0.090	50	0.110	61	0.113
42	0.087	31	0.094	53	0.113	62	0.116
42	0.089	48	0.097	56	0.116	63	0.119
42	0.091	48	0.099	61	0.119	67	0.122
42	0.093	48	0.100	61	0.121	67	0.124
42	0.095	48	0.102	62	0.122	68	0.126
42	0.097	48	0.103	62	0.124	69	0.129
42	0.099	48	0.105	62	0.125	69	0.131
42	0.102	48	0.107	64	0.127	69	0.133
44	0.104	48	0.108	64	0.128	72	0.135
56	0.106	48	0.109	64	0.130	72	0.137
56	0.107	48	0.111	65	0.131	73	0.138
56	0.108	48	0.112	65	0.133	73	0.140
56	0.109	48	0.114	68	0.134	74	0.141
56	0.110	49	0.115	68	0.135	75	0.142
56	0.111	49	0.117	70	0.137	76	0.142
56	0.113	49	0.118	71	0.138	76	0.143
56	0.114	49	0.120	72	0.139	77	0.144
56	0.115	49	0.121	72	0.141	77	0.145
56	0.117	49	0.122	72	0.142	78	0.145
56	0.118	49	0.124	72	0.143	78	0.146
56	0.119	50	0.125	72	0.145	78	0.147
56	0.120	50	0.127	74	0.146	78	0.148
56	0.121	50	0.128	74	0.147	78	0.149
56	0.123	50	0.129	74	0.148	79	0.150
56	0.124	60	0.131	74	0.149	79	0.152
56	0.125	62	0.131	78	0.150	79	0.153
65	0.126	62	0.132	79	0.151	79	0.155
65	0.127	62	0.133	79	0.152	79	0.157
65	0.127	62	0.133	80	0.152	79	0.158
65	0.128	62	0.134	80	0.153	79	0.160
65	0.128	62	0.135	80	0.153	79	0.163
65	0.129	62	0.135	80	0.154	82	0.165
65	0.129	62	0.136	81	0.155	82	0.166
65	0.129	62	0.136	81	0.155	82	0.168

Market (Spectral)	500		1000		5000		10000
Zeros	Error	Zeros	Error	Zeros	Error	Zeros	Error
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.034	0	0.020	8	0.034	12	0.039
18	0.050	0	0.030	18	0.050	27	0.054
23	0.058	3	0.039	23	0.058	31	0.063
28	0.066	3	0.046	28	0.066	37	0.073
33	0.074	3	0.052	33	0.074	44	0.082
36	0.080	8	0.057	36	0.080	47	0.087
37	0.085	8	0.062	37	0.085	52	0.091
42	0.089	10	0.067	42	0.089	54	0.095
46	0.094	10	0.072	46	0.094	56	0.100
50	0.098	11	0.076	50	0.098	60	0.104
50	0.102	12	0.081	50	0.102	60	0.107
50	0.106	12	0.086	50	0.106	60	0.110
51	0.110	12	0.091	51	0.110	63	0.114
54	0.114	13	0.096	54	0.114	63	0.117
56	0.118	13	0.101	56	0.118	67	0.119
59	0.121	19	0.105	59	0.121	67	0.122
60	0.125	30	0.109	60	0.125	67	0.125
61	0.128	38	0.112	61	0.128	67	0.127
62	0.131	38	0.114	62	0.131	69	0.130
63	0.134	38	0.117	63	0.134	69	0.133
64	0.136	38	0.120	64	0.136	70	0.136
64	0.138	39	0.122	64	0.138	71	0.138
68	0.140	39	0.125	68	0.140	72	0.141
69	0.142	40	0.127	69	0.142	73	0.143
69	0.144	40	0.130	69	0.144	74	0.145
69	0.146	40	0.132	69	0.146	74	0.146
69	0.149	40	0.135	69	0.149	74	0.148
69	0.151	40	0.137	69	0.151	74	0.150
74	0.152	40	0.139	74	0.152	74	0.151
74	0.154	41	0.142	74	0.154	75	0.154
74	0.155	41	0.144	74	0.155	75	0.156
74	0.156	41	0.146	74	0.156	76	0.158
74	0.157	41	0.149	74	0.157	77	0.160
74	0.158	42	0.151	74	0.158	77	0.163
75	0.160	43	0.153	75	0.160	77	0.165
76	0.161	43	0.155	76	0.161	77	0.168
77	0.162	43	0.158	77	0.162	77	0.170
77	0.163	43	0.160	77	0.163	80	0.172
77	0.165	43	0.163	77	0.165	80	0.174
77	0.166	43	0.165	77	0.166	80	0.176
77	0.167	44	0.168	77	0.167	80	0.177
77	0.169	57	0.170	77	0.169	80	0.179
77	0.170	57	0.172	77	0.170	81	0.182
77	0.171	57	0.174	77	0.171	81	0.184
78	0.173	58	0.175	78	0.173	81	0.186
80	0.174	58	0.177	80	0.174	81	0.188
82	0.175	58	0.178	82	0.175	81	0.190

4.5 Sparse Group Lasso - by Dataset (Cont...)

Stochastic (Spectral)	500		1000		5000		10000
Zeros	Error	Zeros	Error	Zeros	Error	Zeros	Error
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	0.027	0	0.036	7	0.050	9	0.053
0	0.039	3	0.052	18	0.069	22	0.072
2	0.047	7	0.062	26	0.080	30	0.083
2	0.054	7	0.070	31	0.088	34	0.092
2	0.061	16	0.077	38	0.096	45	0.100
10	0.065	17	0.081	43	0.102	47	0.106
11	0.069	18	0.086	45	0.107	48	0.113
11	0.072	18	0.090	46	0.113	50	0.120
11	0.075	18	0.095	46	0.119	51	0.128
11	0.078	19	0.099	48	0.126	53	0.135
11	0.081	19	0.103	50	0.132	56	0.142
11	0.084	19	0.108	51	0.139	59	0.149
11	0.087	19	0.112	52	0.146	59	0.155
11	0.090	19	0.116	55	0.152	62	0.162
11	0.093	22	0.121	58	0.158	63	0.169
11	0.096	24	0.125	58	0.165	63	0.175
11	0.099	24	0.129	58	0.172	63	0.182
11	0.102	24	0.132	60	0.178	64	0.189
11	0.105	25	0.136	61	0.185	64	0.196
12	0.108	25	0.140	61	0.191	64	0.203
12	0.111	25	0.144	63	0.197	66	0.209
12	0.115	25	0.148	63	0.204	67	0.216
12	0.118	25	0.152	63	0.210	67	0.222
12	0.121	25	0.156	63	0.216	68	0.229
12	0.124	26	0.159	63	0.222	68	0.235
13	0.128	27	0.163	63	0.228	70	0.242
13	0.131	27	0.167	63	0.234	70	0.248
13	0.134	27	0.171	64	0.240	71	0.255
13	0.137	27	0.175	64	0.246	71	0.261
13	0.141	27	0.178	64	0.253	71	0.267
13	0.144	28	0.182	64	0.259	72	0.273
13	0.147	28	0.186	64	0.265	72	0.279
13	0.151	28	0.189	64	0.270	72	0.285
13	0.154	28	0.193	65	0.276	72	0.291
13	0.157	29	0.197	66	0.282	72	0.297
40	0.161	29	0.201	68	0.288	73	0.303
40	0.163	29	0.204	69	0.294	74	0.309
40	0.165	31	0.208	69	0.299	75	0.315
42	0.167	51	0.212	70	0.305	75	0.321
42	0.169	51	0.214	70	0.310	75	0.327
42	0.171	51	0.216	70	0.315	75	0.333
42	0.173	51	0.218	70	0.320	75	0.339
42	0.175	51	0.221	70	0.325	75	0.345
42	0.177	51	0.223	70	0.331	75	0.351
42	0.179	51	0.225	70	0.336	76	0.358
42	0.181	51	0.228	70	0.341	76	0.364
42	0.183	51	0.230	70	0.347	76	0.371
42	0.184	52	0.233	70	0.352	76	0.377
42	0.186	52	0.235	71	0.358	76	0.384