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

Alejandro Saucedo

29th April, 2013

# SPARSE AND GROUP REGRESSION MODELS IN PORTFOLIO OPTIMIZATION

Project Supervisor: Prof. Mahesan Niranjan

Second Examiner: Prof. Vladimiro Sassone

A progress report submitted for the award of

Software Engineering BSc

# Abstract

Current approaches to portfolio optimization only consider financial instruments as individual entities, not taking any advantage of the huge amounts of categorizing information available on the underlying financial instruments (i.e. instrument type, industry, sector, volatility level, etc). In this paper we take a novel approach to portfolio optimization, where our main objective is to show that it is possible to exploit categorizing information on the financial instruments in the portfolio and benefit from group correlations present in the data to obtain richer results.

This paper focuses in a very popular branch of portfolio optimization – namely, market index tracking, where the aim is, chosen a Market Index (i.e. a set of high-performing stocks), we want to find a subset that follows the behaviour of its respective Market Index as close as possible. Our approach to solving this problem consists of feature-level regression models with a cardinality (L0-norm) constraint[1], and sparse-inducing group-level regression models[2]. These two approaches will be introduced, analysed and compared in order to provide an insight on the effect group characteristics have when implemented in financial datasets.

Given that the Sparse Group Model analysed in this paper is limited to a single category (i.e. only one category of groups can be taken into consideration) a new regression model was proposed based on our results. This model suggests to take into consideration multiple categories of groups (i.e. type of financial instrument, sector, volatility, etc.) in order to provide more diverse portfolios, and more accurate results.

Finally, the code that was written for the implementation of these models can be found online in a well documented GitHub repository, which has been registered on an Open Source licence and is available for download.

[1] Simple Regression Models: Sum of Absolute Values, Sum of Squares, Ridge Regression, Conditional-Value-at-Risk and Lasso.
[2] Group Lasso and Sparse Group Lasso

# Contents

# 1 Introduction

Portfolio optimization has been a major research topic since it was introduced in Markowitz's paper, Modern Portfolio Theory [2] – one of the most influential research papers in the field of finance. Since its debut, this theory has been expanded into several branches, including index tracking, enhanced indexation, absolute return, between others.

Until this date, portfolio optimization models have only considered financial instruments as individual entities of data – that is, using only simple regression models (e.g. linear, ridge, weighted regression, etc). In this paper, we take a novel approach into portfolio optimization, where we study the effects of sparse inducing models, and expand this concept by considering financial instruments as groups, as opposed to as only individual instruments. The information on which instruments comprise which groups can be obtained through services like yahoo finance, and can consist of simple characteristics such as financial sector, industry of the underlying company, or any characteristic of the instruments that allows for classification.

Some of our assumptions when analysing our results will be based in the the conclusions stated in "Market and Industry factors in stock price behavior"[10], a paper from the Journal of Business which strives to determine how much of cross-sectional interdependence among a set of series of monthly price relatives could be explained by market and industrial factors, as described in [11]. This paper approaches this problem with a factor analysis approach, and concludes that "the movement of a group of security price changes can be broken down into market and industry components… the petroleums, for example, are highly clustered in their movement over time, as are steels, rails, and, to a lesser extent, utilities."[10] With this, we can observe that a key characteristic in financial group categories is the correlation between the stocks within a group. This finding will be essential throughout the implementation and analysis of this paper, as the sparse regression models act on correlations found in the data.

The working code for the Implementation of this paper has been registered in an Open Source public licence. At the time of writing, the code can be obtained form http://github.com/axsauze/sparse. Due to the word limit in this paper, background on all the financial terms, concepts and definitions used in this paper, as well as thorough explanations on the regression and algorithmic approaches used in this paper have been moved to the "Annex.pdf" document stored in the repository.

This paper is structured in five main sections – namely, Background Research, Implementation, Analysis, Expansion, and Conclusion.

# 2 Background and Report of Literature Search

In this paper we aim to discover the effects of feature and group level sparsity to Market Index tracking. In order to achieve this, it is required to understand several financial terms, as well as the Machine Learning concepts used in this paper. This section aims to provide the reader with the core knowledge required on definitions, formulas and concepts related to the topics that comprise feature and group level regression approaches in our financial datasets. A big number of topics had to be moved to the annex document located repository due to the word limitation – topics included there are very important for a better understanding of this project, and include: Simulation Models, MPT, Value at Risk, Expected Shortfall, β-VaR, CVaR, Risk and Diversification.

This project was initially inspired by [1], where an innovative regression model was proposed – namely the L0+L2-norm model. This model proposed is basically a ridge regression model (i.e. constrained on an L2-norm) with a cardinality constraint (i.e. constrained on an L0-Norm) – hence the name of the model.

## 2.1 Core Definitions

### *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 L$\varphi$–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 $\mathbb{R}^{Txn}$ 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 – this is known as the 'short sale constant' which ensures that there is no

shorting. 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 $\overline{\boldsymbol{R}} = (\overline{\boldsymbol{R}}_{\mathbf{1}}, \overline{\boldsymbol{R}}_{\mathbf{2}}, \dots, \overline{\boldsymbol{R}}_{\boldsymbol{m}})$, where $\overline{\boldsymbol{R}}_{\boldsymbol{i}} \subseteq \mathbf{R}$ and $\overline{\boldsymbol{R}}_{i,t} \in \mathbb{R}^{\boldsymbol{p_i}} = (\overline{\boldsymbol{R}}_{i,t,1}, \dots, \overline{\boldsymbol{R}}_{i,t,p_i})^{\mathrm{T}}$.

Although our Index Portfolio would be referenced with the same variable $\mathbf{I}$, a new group definition is now introduced as $\boldsymbol{I} = \overline{\boldsymbol{R}}_{\mathbf{1}}\overline{\boldsymbol{\pi}}_{\mathbf{1}} + \overline{\boldsymbol{R}}_{\mathbf{2}}\overline{\boldsymbol{\pi}}_{\mathbf{2}} + \cdots + \overline{\boldsymbol{R}}_{\boldsymbol{m}}\overline{\boldsymbol{\pi}}_{\boldsymbol{m}}$, where $\overline{\boldsymbol{\pi}}_{\boldsymbol{i}} \in \mathbb{R}^{\boldsymbol{p_i}}$ 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 $\overline{\boldsymbol{\pi}}_{\boldsymbol{g}}$ contains stocks 3, 5, 6, etc, then $\boldsymbol{\Psi_g}$ = {3, 5, 6, …}. This allows to introduce the biconditional $\left(\overline{\boldsymbol{\pi}}_{\boldsymbol{g}} \subseteq\ \boldsymbol{\pi}\right) \Leftrightarrow\ \left(\boldsymbol{\pi}\left[\, 1, \boldsymbol{\Psi_g}\,\right] =\ \overline{\boldsymbol{\pi}}_{\boldsymbol{g}}\right)$.

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 $\mathbf{A}$ and c is the number of columns in $\mathbf{A}$.

### *2.1.3 Index Tracking*

In this paper, all our results will revolve around the Index Tracking portfolio optimization problem. This problem consists of, given a market index (i.e. set of stocks), we need to find a subset of size $C_0$ that minimizes tracking error. Tracking error is basically a synonym for Test Error, and it is defined as follows:

**Equation 1 - Tracking Error**

$$\frac{1}{T}\sqrt[1/\varphi]{\left[\sum_{t=1}^{T}\left|I_t - R_t^{\mathrm{T}}\pi\right|^{\varphi}\right]}$$

The reason why in finance we focus on finding a smaller subset that tracks a Market Index is because it is infeasible to purchase all the constituent stocks from a Market Index due to expensive transaction costs.

## 2.2 Feature-Level 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.

### *2.2.1 CVaR Minimization*

It is proven in [2] 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 2 - CVaR**

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

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

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

It is worth mentioning that a formulation is proposed in [2], which is a variation of this model called the Norm Constrained CVaR which implements sparsity through a variable C2. 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.

### *2.2.2 Absolute Error Model (Abs)*

The Abs method is probably the simplest, yet very effective minimization methods - this is basically a minimization of the absolute sum of errors (L1-Norm). As it can be observed this would be the exact same as a minimization on Index Tracking Error, as the formulations are exactly the same. We will refer to this mode as *Abs*.

**Equation 3 - Abs**

$$\min_{\boldsymbol{\pi}} \ \|\boldsymbol{I} - \boldsymbol{R}^{\mathrm{T}}\boldsymbol{\pi}\|_1$$

### *2.2.3 Least Squares Minimization (Squares)*

The least squares minimization is a classic when it comes to Machine Learning. This method is also one of the simplest, most efficient and most straightforward method. Throughout this paper, this method will be referred to as *Squares*. This is the same model implemented in Markowitz' MPT paper [REFERENCE], which is defined as follows:

**Equation 4 - Squares**

$$\min_{\boldsymbol{\pi}} \ \|\boldsymbol{I} - \boldsymbol{R}^{\mathrm{T}}\boldsymbol{\pi}\|_2^2$$

### *2.2.4 L0+L2-norm Model (Ridge Regression)*

The L0+L2-norm model as proposed in [1], as its name implies is basically a regression model constrained by an L0-norm and an L2-norm. In simple words, this model is basically a Ridge Regression model with a cardinality constraint. When this mode was proposed, the main objective was to find an effective application in Index Tracking – mainly finding a subset of size smaller than $C_0$ (L0-norm constraint limited by $\|\boldsymbol{\pi}\|_0 \leq C_0$) from an initial portfolio, which behaves as similarly as possible to the initial.

As it can be observed, the norm constrained CVaR provides us with the **L2-norm** component of the L0+L2-norm model, which allows to re-define the L0+L2 model as follows:

**Equation 5 - Ridge**

$$\min_{\boldsymbol{\pi}} f^{\tau}(\boldsymbol{\pi}) = \sum_{t=1}^{T} \left|\boldsymbol{I}_t - \boldsymbol{R}_{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{\pi}\right| + \ \lambda\|\boldsymbol{\pi}\|_2^2$$

s.t.

$$\|\boldsymbol{\pi}\|_0 \leq C_0$$

$$\|\boldsymbol{\pi}\|_2 \leq C_2$$

This allows us to observe the clear division between the L0-norm given by the cardinality constraint $\|\boldsymbol{\pi}\|_0 \leq C_0$ and the L2-norm given by the density constraint $\|\boldsymbol{\pi}\|_2 \leq C_2$. These two norms are controlled by the variable $\lambda$. If $\lambda \to \infty$, the model would behave as a L2-norm constrained model, and $\lambda = 0$ would give use an L-0-norm constrained model only. In [0] Mahesan et al. (2013) the constraint $C_2$ is not taken into account, hence in that case, when $\lambda \to \infty$, the optimal solution exists when all the values of $\boldsymbol{\pi_i} = {}^{1}/_{C_0}$.

#### *2.2.5 Lasso Regression*

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 6 - Lasso**

$$\min_{\boldsymbol{\pi}} \sum_{i=1}^{T} \left| \boldsymbol{I}_t - \boldsymbol{R}_t^{\mathrm{T}} \boldsymbol{\pi} \right| + \lambda \left| \boldsymbol{\pi} \right|$$

What makes this model special is the scaled sum of absolute values, which penalizes the equation with the values of the weights. Due to this, features that are less correlated to the target data are drawn to zero. An in depth explanation on this model, as well as an example can be found in the Annex in the repository.

### 2.3 Model Selection Approaches

#### *2.3.1 Feature Selection*

When implementing the *Abs, Ridge, Squares* and *CVaR* regression models it is required to propose a way to induce sparsity in the weights of the parameter to learn. This will allow us to obtain a subset of stocks to represent our Market Index. The way this will be achieved in this paper is through an L0-norm constraint to all our single-feature models, and is implemented with the Greedy Forward Search algorithm proposed in [1]. This algorithm will be referred to as *GFS* (Greedy Forward Search).

#### *2.3.2 Group Selection*

In this paper it is required to find an algorithm that will allow us to induce sparsity between groups while still being able to apply our single-feature regression models (i.e. Abs, Ridge, Squares and CVaR). For this, we expand the GFS approach

introduced previously into a group full search algorithm which we'll refer to as FS. This will be, as the name proposes, a full search algorithm subset algorithm that will try every possible combination of groups to find an 'optimal ' subset.

## 2.4 Group Regression Approaches

This section contains information model selection does not allow for varied weightings on each of the groups – instead it can only either include or exclude them completely.

### *2.4.1 Group Lasso Regression*

Now that the reader has obtained a core knowledge on the characteristics and functionality of the Lasso model and its components it is possible to expand the definition into groups. For this, we assume that the variables are clustered in groups, and instead of having a penalization on the sum of absolute values, the values of Euclidean norms of the parameters in each group are used. What this does is drive all values in each group to the same exact value, so with this, we would have a lasso model which induces sparsity within groups.

Group lasso was proposed in a very interesting paper by Yuan & Lin (2007) [9]. Recalling the mathematical group notation introduced in the beginning of this paper, this model has been adapted to our financial implementations, which is defined as follows:

**Equation 7 - Group Lasso**

$$\min_{\overline{\boldsymbol{\pi}} \in \mathbb{R}^n} \quad \frac{1}{2}\left\| \boldsymbol{I}_t - \sum_{g=1}^{m} \boldsymbol{R}_{g,t}^{\mathbf{T}} \overline{\boldsymbol{\pi}}_g \right\|_2^2 + \lambda \sum_{g=1}^{m} \sqrt{p_g} \|\overline{\boldsymbol{\pi}}\|_2$$

Intuitively it can be observed that this model follows the same methodology than the individual Lasso model presented earlier – the only difference is that this formulation acts on data as groups of stocks, as opposed to individual stocks. We can observe that a minimal Euclidean norm (L2-norm) can be achieved only if all the elements of the group are of the same size, as a group with larger standard deviation would have a larger Euclidean form. If the group size for all groups is one, this problem would reduce to a simple individual Lasso model as shown in [12]. Similar to the Lasso formulation, if the value of $\lambda$ is too high, group sparsity would be enforced in our variable $\overline{\boldsymbol{\pi}}$ making all the groups approach to zero.

### *2.4.2 Sparse group lasso*

The only problem with the simple group lasso introduced previously is that all the features in each group have equal values, which means that there is no sparsity within the elements inside a group. What this means is that if one group of features is zero, then all the features in the group would also be zero as well.

An innovative model is proposed in [13] which not only provides the sparsity between groups, but also provides sparsity within individual levels. This is known as the Sparse Group Lasso, which is defined as follows:

**Equation 8 - α-Sparse Group Lasso**

$$\min_{\overline{\boldsymbol{\pi}} \in \mathbb{R}^{\boldsymbol{n}}} \quad \frac{1}{2n}\left\|\boldsymbol{I} - \sum_{g=1}^{m} \boldsymbol{R}_g^{\mathrm{T}} \overline{\boldsymbol{\pi}}_g\right\|_2^2 + (1-\alpha)\lambda \sum_{g=1}^{m} \sqrt{p_g}\left\|\overline{\boldsymbol{\pi}}_{\boldsymbol{g}}\right\|_2 + \alpha\lambda\|\boldsymbol{\pi}\|_1$$

$$\text{s.t.} \qquad \alpha \in [0,1]$$

As it can be observed, the only difference in this formulation is the last term included, which is simply the L1-norm regularizer (sum of absolute values) found in the simple single-feature lasso model. This term, together with our variable $\alpha$, which balances the ratio between which this regression model considers group sparsity and individual feature sparsity.

This model is revisited again, and a new approach is given in [12] where another simple implementation of the Sparse Group Lasso is proposed. In this implementation, the $\alpha$ coefficient is dropped, allowing us to introduce $\lambda_1$ and $\lambda_2$:

**Equation 9 - Sparse Group Lasso**

$$\min_{\overline{\boldsymbol{\pi}} \in \mathbb{R}^{\boldsymbol{n}}} \quad \left\|\boldsymbol{I} - \sum_{g=1}^{m} \boldsymbol{R}_g^{\mathrm{T}} \overline{\boldsymbol{\pi}}_g\right\|_2^2 + \lambda_1 \sum_{g=1}^{m} \sqrt{p_g}\left\|\overline{\boldsymbol{\pi}}_{\boldsymbol{g}}\right\|_2 + \lambda_2\|\boldsymbol{\pi}\|_1$$

It should be mentioned that from our practical experience, the best results are obtained when $\lambda_2$ contains a scaled value of $\lambda_1$ (Hence the value of $\alpha$), however, for the sake of simplicity, and for being able to offer the reader with a clear difference for when we refer to either the Group Sparsity inducing coefficient ($\lambda_1$) or when we refer to the Feature Sparsity inducing coefficient ($\lambda_2$). We will instead use the definition $\lambda_2 = \Omega\lambda_1$ where $\Omega$ will define the scale factor.

To provide an intuition to the reader on what this model tries to achieve, we provide the following perspective on each of the cofactors of the formulation:

**Equation 10 - Intuitive Sparse Group Lasso**

$$\min \quad Test\ error^2 + \lambda_1 * \begin{matrix} Variation \\ within\ groups \end{matrix} + \lambda_2 * \begin{matrix} Sum\ of\ absolute \\ weights\ of\ features \end{matrix}$$

The first coefficient induces a simple the minimization on test error, or tracking error, the second coefficient penalizes the formulation if there is a high variation within groups (higher Euclidean distance) inducing balance of weights within groups together with group sparsity, and the final coefficient penalizes the formulation on the sum of absolute values of the individual features, which induces sparsity within groups.

## 2.5 Zero-Constrained Group Lasso Application

The biggest challenge in this paper was to come up with a formula that does not require the inner loops that most implementations require as the machine learning library used for the implementation cannot process the inner loops required for cofactors such as $\sum_{g=1}^{m} \boldsymbol{R}_g^{\mathrm{T}} \overline{\boldsymbol{\pi}}_g$ and $\sum_{g=1}^{m} \sqrt{p_g} \left\| \overline{\boldsymbol{\pi}}_{\boldsymbol{g}} \right\|_2$. In order to come up with a solution, a new constraint was implemented which allowed us to solve this equation with only matrix multiplications – this constraint is explained below.

### *2.5.1 Mathematical Definitions*

For this section we will require the variable $\boldsymbol{\Psi}_{\boldsymbol{g}}$ introduced in our mathematical notation where $\boldsymbol{\Psi}_{\boldsymbol{g}}$ contains the indexes of for the stocks in group g – which are the stocks contained in $\overline{\boldsymbol{\pi}}_{\boldsymbol{g}}$. With this, the formal notation should also be intuitive $\left(\overline{\boldsymbol{\pi}}_{\boldsymbol{g}} \subseteq \boldsymbol{\pi}\right) \Leftrightarrow \left(\boldsymbol{\pi}\left[\, 1, \boldsymbol{\Psi}_{\boldsymbol{g}} \,\right] = \overline{\boldsymbol{\pi}}_{\boldsymbol{g}}\right)$.

For the sake of simplicity we will be using the function sum(x,i) throughout our implementation. This function was introduced in the first section of this paper, and outputs the sum of the rows (i=1) or columns (i=2). The sum of the columns or rows of any matrix can be represented with a matrix multiplication of a column or row vector of ones respectively, however, this function will be used for making the explanation simpler.

### *2.5.2 Constraints*

Initially we have to replace the first summation, namely $\left(\sum_{g=1}^{m} \boldsymbol{R}_g^{\mathrm{T}} \overline{\boldsymbol{\pi}}_g\right)$, for a cofactor that does not require this loop. Our solution proposed is $\left(sum(\boldsymbol{R}^{\mathrm{T}} \widehat{\overline{\boldsymbol{\pi}}}, 2)\right)$, where where $\widehat{\overline{\boldsymbol{\pi}}} \in \mathbb{R}^{\boldsymbol{n} \times \boldsymbol{m}} \wedge \left(\widehat{\overline{\boldsymbol{\pi}}}\left[\, \boldsymbol{\Psi}_{\boldsymbol{g}}, g \,\right] = \overline{\boldsymbol{\pi}}_g\right) \wedge sum\left(\widehat{\overline{\boldsymbol{\pi}}}, 1\right) = \boldsymbol{\pi}$. In simple terms, $\widehat{\overline{\boldsymbol{\pi}}}$ can be seen as an n by m (Number of stocks by number of groups) matrix, where each row contains the weights for a specific group in its respective index position and the rest of the elements are zeros.

To provide a more clear understanding on this, lets assume $\boldsymbol{\pi} = (0.2, 0.3, 0.2, 0.3)$, we have m=2 groups where $\boldsymbol{\Psi_1} = \{1,3\} \wedge \boldsymbol{\Psi_2} = \{2,4\}$ which gives us $\overline{\boldsymbol{\pi}}_{\boldsymbol{1}} = (0.2, 0.2)$ and $\overline{\boldsymbol{\pi}}_{\boldsymbol{1}} = (0.3, 0.3)$. These definitions allow us to build $\widehat{\overline{\boldsymbol{\pi}}}$ as:

$$\widehat{\overline{\boldsymbol{\pi}}} = \begin{bmatrix} 0.2 & 0 & 0.2 & 0 \\ 0 & 0.3 & 0 & 0.3 \end{bmatrix}$$

With this we can relate our new introduced terms to the previous implementations of the group lasso – it should be obvious that $(\widehat{\overline{\boldsymbol{\pi}}}[\,\boldsymbol{\Psi_1}, 1\,] = \overline{\boldsymbol{\pi}}_1) \wedge (\widehat{\overline{\boldsymbol{\pi}}}[\,\boldsymbol{\Psi_2}, 2] = \overline{\boldsymbol{\pi}}_2)$. It should also be noted that in this example we have that $sum(\widehat{\overline{\boldsymbol{\pi}}}, 1) = \boldsymbol{\pi}$ holds, however we will need this constraint to hold for all cases.

Now that we obtained a way to represent this cofactors with only matrix multiplications, we need a constraint that ensures that always $sum(\widehat{\overline{\boldsymbol{\pi}}}, 1) = \boldsymbol{\pi}$**.** In other words, to make sure that all the elements in our variable $\widehat{\overline{\boldsymbol{\pi}}}$ are always zero if they do not belong to their respective group. Following this definition, we specify $\sim\boldsymbol{\Psi_g}$ to be the set containing the indices of the elements that are **not** in group g. Formally, we define $\widehat{\overline{\boldsymbol{\pi}}}[\,\sim\boldsymbol{\Psi_g}, g\,] = \boldsymbol{\pi} \backslash \overline{\boldsymbol{\pi}}_{\boldsymbol{g}}$**.** This allows us to introduce the constraint that makes this implementation possible, which is:

$$\left(\forall g \in \{1, \dots, m\}.\ \widehat{\overline{\boldsymbol{\pi}}}[\,\sim\boldsymbol{\Psi_g}, g\,] = 0\right) \Longrightarrow \left(\boldsymbol{\pi} = sum(\widehat{\overline{\boldsymbol{\pi}}}, 1)\right)$$

The only thing left now is to make the term $\sum_{g=1}^{m} \sqrt{p_g} \left\| \overline{\boldsymbol{\pi}}_{\boldsymbol{g}} \right\|_2$ suitable for our operations. It can be noticed that the only thing we will require now is a way to calculate the L2-norm of each of our individual vectors (rows of $\overline{\boldsymbol{\pi}}_{\boldsymbol{g}}$), as currently our function $\| \, . \, \|_2$ returns a scalar which represents the L2-Norm of the matrix, however we need the L2 norm for each of the individual groups. This is a very simple thing to do, which can be achieved with the formula $\sqrt{sum(A^2, 2)}$. This allows us to transform our second cofactor into:

$$\text{sum}(\sqrt{p^{\mathrm{T}}} * \sqrt{sum(\widehat{\overline{\boldsymbol{\pi}}}^2, 2)})$$

### *2.5.3 Model Definition*

With this new constraint, and this new second term, it is possible to introduce the Group Lasso and Sparse Group lasso models that will be used throughout this paper:

**Equation 11 - Zero-Constrained Group Lasso**

$$\min_{\overline{\boldsymbol{\pi}} \in \mathbb{R}^{\boldsymbol{n}}} \ \frac{1}{2} * \left\| \boldsymbol{I} - sum(\boldsymbol{R}^{\mathrm{T}} \widehat{\overline{\boldsymbol{\pi}}}, 2) \right\| + \ \lambda * \ sum\left( \sqrt{p^{\mathrm{T}}} * \sqrt{sum(\widehat{\overline{\boldsymbol{\pi}}}^2, 2)} \right)$$

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

Which is equivalent to the lasso formulation presented earlier, however, all the computations are all done through matrix multiplications, which makes it suitable for our implementation. An implementation of the algorithm can be found in the annex in the repository.

# 3 Implementation

This section assumes that the reader has an appropriate understanding on the financial terms and regression models used throughout this paper, and contains the implementation of the models and algorithms presented in the previous section.

This section begins with a description of the development environment in which the code snippets were ran and tested, as well as information on the mathematical libraries used to obtain such results. This section will then follow with the implementation of the feature regression models, and finally the group regression models.

As mentioned in the introduction, the working code for this section has been registered on a public Open Source licence, which at the time of writing is being hosted in GitHub, and can be downloaded at http://github.com/axsauze/sparse.

## 3.1 Working Environment

All the experiments in this paper were ran in a MacBook pro with OS X Mountain Lion 10.8.5, Intel Core i7 2.9 GHz processor, with 8 GB 1600MHz DDR3 SDRAM.

All code is written in Matlab R2014a, and two main libraries were used for linear regression computations – namely CVX and SPAMS. Details on compilation and installation of these can be found in the main repository.

### *3.1.1 CVX*

CVX is a Machine Learning library for Matlab that provides modelling tools that allow the programmer to specify constraints and objectives using standard Matlab code. This library allows great flexibility when it comes to solving simple regression problems such as least-squares, weighted regression, ridge regression, etc.

The CVX library would allow us to translate simple regression problems as follows:

$$\begin{aligned} \text{minimize} \quad & \|Ax - b\|_2 \\ \text{subject to} \quad & Cx = d \\ & \|x\|_\infty \leq e \end{aligned}$$

```
cvx_begin
    variable x(n)
    minimize( norm( A * x - b, 2 ) )
    subject to
        C * x == d
        norm( x, Inf ) <= e
cvx_end
```

### *3.1.2 SPAMS*

There are very few implementations out there for accurate group selection models. The Sparse Modelling Software (SPAMS) library released in 2010 is one of the few libraries that provide a very wide range of sparse regression tools. Unfortunately the library is compiled, and has a lot of dependencies, so it's more of a black box, and can't be tweaked to adapt it to our needs. The implementation of these models in financial data is also not as accurate as the CVX modelling toolbox, however, it is very worth mentioning this library, as it provided a better understanding on some sparse concepts applied in this paper. These tools are based on a paper released by the same creators of this library [15], which contains in-depth detail on all of the sparse regression models included in this library, together with several algorithmic approaches and optimization methods.

Although this library takes several approaches to achieve convergence in its constituent regression models, Lasso related models utilize an algorithm devised in [14] which comes from the ISTA (Iterative Shrinkage-Thresholding Algorithms) family of algorithms – namely the FISTA iterative method. This method is used in signal image processing due to its much greater converging speed.

## 3.2 Implementation Design

### *3.2.1 Financial Implementation*

Experiments are carried out on various sets of data in order to provide more robust and reliable results. Sets of data consist of both, real and simulated financial data.

#### *3.2.1.1 Market Financial Data*

In this paper, the Market Index to be analysed will be the FTSE100, which is obtained through the Yahoo Finance API. Our data will consist of the 100 stocks that comprise this index for a time window of T=274 working days (starting from 1st of November, 2011, until the 1st of December, 2012). This allows us to form our matrix of returns $\mathbf{R} \in \mathbb{R}^{T \times n}$ such that $\mathbf{R}_t$ is a vector of the returns of all the stocks at time t, defined as $\mathbf{R}_t = (\mathbf{R}_{t,1}, \mathbf{R}_{t,2}, \ldots, \mathbf{R}_{t,n})^T$. As stated previously, our Market Index $\boldsymbol{I}$, or target data, is computed by multiplying $\boldsymbol{I} = \boldsymbol{R}_t^T \boldsymbol{\pi}'$, where $\boldsymbol{\pi}'_i = 1/n$ for I = {0,…,n}.

#### *3.2.1.2 Simulated Financial Data*

Similarly, simulated data will be defined by the same mathematical notation as the Market Financial data, and will consist of a number of n=200 stocks for a time window T=300. Several time-series data generation models were considered and tested in order to find the most efficient model for this paper. These models include the GARCH, ARMAX, EULER, and finally Monte Carlo. The model chosen for the implementations in this paper was the Monte Carlo model, as the results proved to

simulate our FTSE100 data better in regards to group correlations within the data. These group correlations were analysed with the spectral clustering algorithm we introduced to find groups of correlated stocks. More detailed definitions and applications of these models are found in the annex in the repository.

#### *3.2.1.3 Data Grouping*

The grouping approaches in this paper are based in the assumption introduced in section 1 where correlations between the stocks within groups of financial categories are present (Groups can include stock industry, sector, etc)[9]. In this paper, the grouping categories considered are namely the sector of the underlying stocks, as well as groupings obtained through a spectral clustering algorithm based on the respective stock correlation matrix.

#### *3.2.1.4 Error Measurement*

For all the regression models implemented in this paper, our input data **R** will be divided into **R_train** and **R_test**, accounting for the first 75% and the last 25% of the original data respectively. The variable $\boldsymbol{\pi}$ will be trained on **R_train**, and the tracking error (test error) will be measured with the offset or **R_test** with our market index **I**. Given that **R_train** comprises the first section of data, and **R_test** the last section, this will account for how well the model predicts the behaviour of future stock price data.

### 3.3 Feature-Level Regression Models

In this section we implement the Abs, Squares, Ridge and CVaR regression models with an L0-norm (cardinality) constraint, (applied through the GFS approach proposed in [1]).

It is worth mentioning that the reason why the GFS approach was needed for the L0-norm constraint was due to its exponential nature; in order to find the 'optimal' subset, we would require a full search considering the possible combination of stocks, making this impossible for any portfolio larger than 20 stocks.

#### *3.3.1 L0-norm constrained models*

In this section, the GFS algorithm is implemented in both market and stochastic data for n iterations. The cardinality constraint $\|\boldsymbol{\pi}\|_1 < C_0$ with $C_0 = 0$. On each iteration, $C_0$ increased by one, and one stock is added to the subset – this is, the stock that computed the lowest tracking error with the regression model chosen. Figures [FIGURE] and [FIGURE] show the results for this implementation, where tracking error for the *Abs*, *Squares*, *Ridge* (L0+L2-norm constrained model[1]) and *CVaR* regression models is plotted against a linearly growing $C_0$. It should be noted that

for the sake of consistency in this paper, the x-axis in the graphs is plotted in an increasing number of zeros (i.e. a decreasing $C_0$ constraint).

It is evident that the *Ridge* model is the most accurate when it comes to predicting accurately – this followed by *CVaR*, *Ridge*, and finally *Abs* being the least accurate.

**Figure 1 - FTSE100**

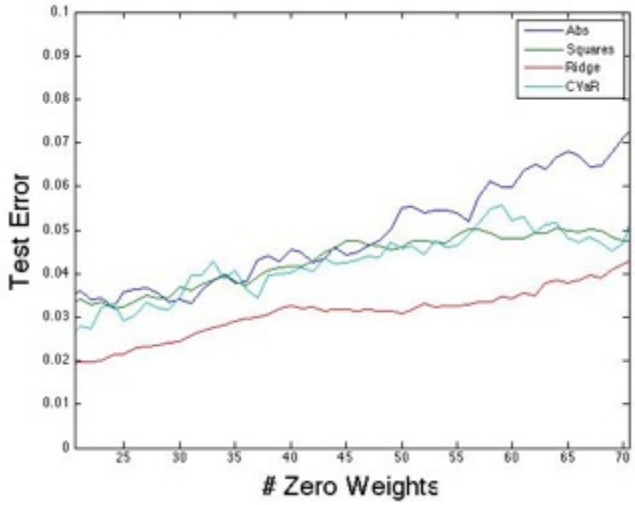
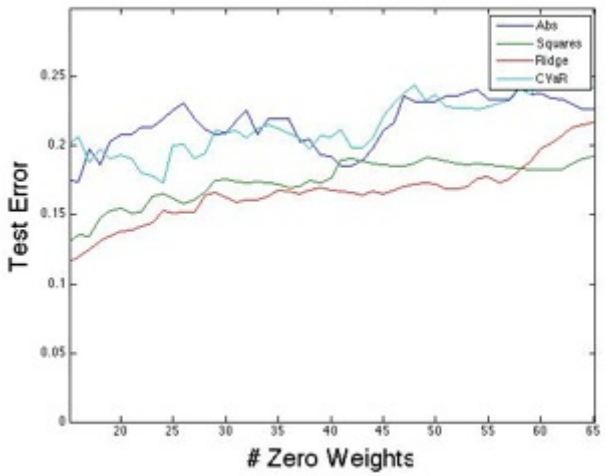



**Figure 2 - Monte Carlo**




An interesting observation was the order in which the stocks were chosen on every iteration by each of these regression models. In order to quantify this, it was required to calculate the size of the union of each of the subsets at each. To make this clear, let's say that on our 3rd iteration, if the stocks chosen are Abs={1,2,3}, Squres={2,3,4}, Ridge={2,3,5}, CVaR={2,3,6}, then the union of this subsets is {2,3}, which means that there are two stocks that have been chosen in all subsets, and hence 2/3 = 66% of the stocks have been chosen in all four models on the 3rd iteration.

**Percentage of Same Elements in Subsets**

**Figure 3 - FTSE100**

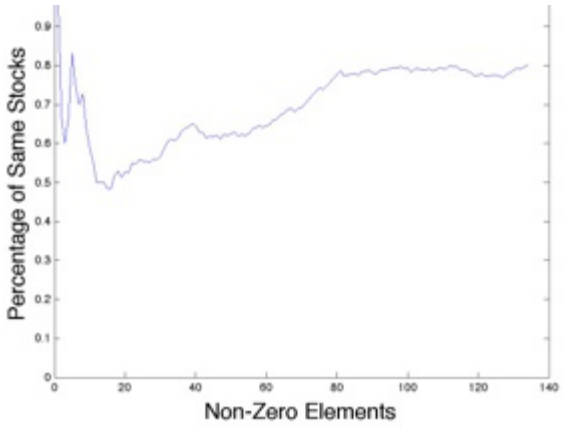
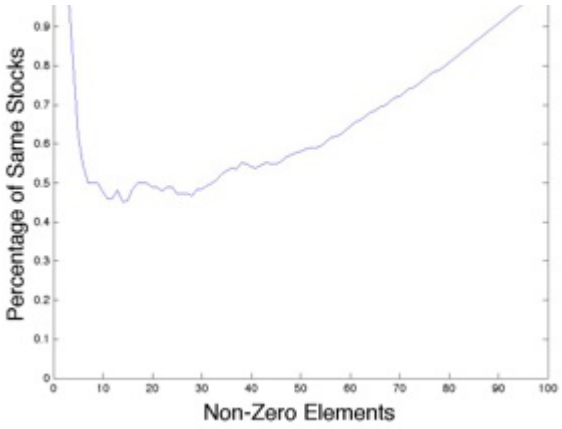



**Figure 4 - Monte Carlo**




It is possible to observe in figures 3 and 4 that for all our iterations, at least 50% of all the subsets consisted of the same stocks in all the four models at each iteration. This shows that there are stocks that there are certainly stocks that are more correlated to the market index, and these are selected no matter which regression model is used. This will be recalled later on in the paper.

The time taken for computing these results should be emphasized. Although the GFS approach taken made the computations of this section possible, it still did not account fully for the huge time required for computation. The FTSE100 dataset by itself consists only of 100 elements, and the Monte Carlo dataset of 200 – it took more than an hour and more than 6 hours to compute these respectively. Baring in mind that in the financial markets the smallest market indexes would consist of 100 stocks, with an average of 250, this approach would be infeasible as a day-to-day trading tool. Computational times will be revisited more in depth in section 4.1.3.

### *3.3.2 The Lasso sparse inducing model*

The Lasso model, as introduced previously, is a potential alternative to the L0-constraint implemented in the previous section which proposes a solution to achieve sparsity in our Index Tracking problem. This is one of the simplest and yet one of the most efficient sparsity inducing regression models. This model can be efficiently solved through the LARS, FISTA and several primal and dual approaches. Descendent approaches from the two latter ones, which are implemented by the CVX library that is used for the implementation. In the following figures it is possible to observe the behavior of the Lasso model as the value of $\lambda$ increases.

**Figure 5 - FTSE100**

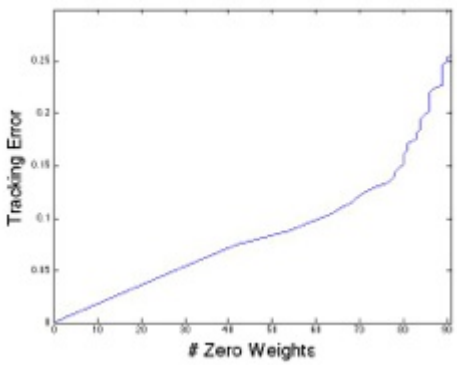
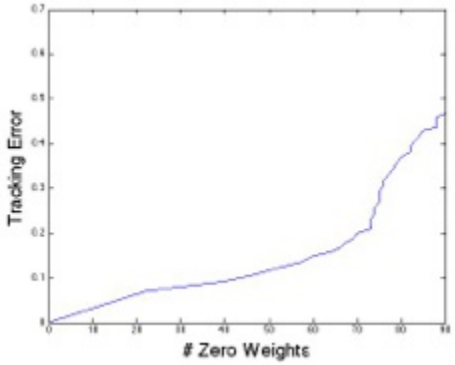



**Figure 6 - Monte Carlo**




The results shown in graphs 5 and 5 were as expected in terms of the behavior of the Lasso model – an increase in the number of zeros as $\lambda$ increases and vice-versa. These results are also shown in the table in figure 7. What should be noticed is the lower training error achieved compared to the L0-norm constrained approach implemented previously.

Once again, it is worth mentioning computation time. The the average length of time taken was 7.6657 seconds, with a standard deviation of 0.7138 seconds. Unlike the GFS approach, this algorithm has a much lower computation time as sparsity is induced with the actual implementation of the model without requiring any extra subset selection algorithms.

**Figure 7 - Zeros Errors Table**

| # Zeros | Tracking Error | # Zeros | Tracking Error |
|---|---|---|---|
| 0 | 0.0000 | 0 | 0.0000 |
| 31 | 0.0561 | 22 | 0.0714 |
| 42 | 0.0754 | 39 | 0.0909 |
| 54 | 0.0883 | 46 | 0.1051 |
| 59 | 0.0965 | 51 | 0.1199 |
| 63 | 0.1030 | 57 | 0.1350 |
| 65 | 0.1085 | 60 | 0.1476 |
| 68 | 0.1142 | 64 | 0.1587 |
| 69 | 0.1189 | 66 | 0.1670 |
| 71 | 0.1243 | 67 | 0.1766 |

## 3.4 Group-Level Regression Models

In this section we aim to obtain relevant results that show the effects of using stock grouping information of the stocks in portfolio optimization. There are several approaches in this section to observe these effects, as well as whether this use of extra information will have potential expansion for practical and real life financial problems.

In this paper group models are approached with two different methods – a group selection approach and a sparse group regression model.

### *3.4.1 Group Selection*

In this section the GFS individual feature concept is expanded into group selection, and a full-search algorithm is used instead of the greedy approach taken previously. This approach is implemented in our FTSE100 and Monte Carlo datasets; subsets of stocks are chosen through a group-level forward search method, where groups are chosen based on their compound training error.

This method is limited due to the large exponential time complexity of the full-search approach taken. However, the reason why this approach is taken is because grouping will allow for smaller numbers of elements, as instead of dealing with n stocks, we would now be dealing with m groups. This limitation should not be a problem in this implementation, as the number of sectors considered in our FTSE100 dataset, as well as the artificial groups obtained with our Spectral Clustering Algorithm is 9. A larger number of groups would however make the implementation of this approach infeasible.

**Figure 8 - FTSE100 (Sectors)**

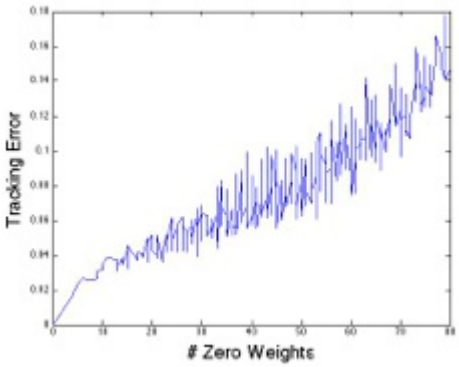



**Figure 9 - FTSE100 (Spectral)**

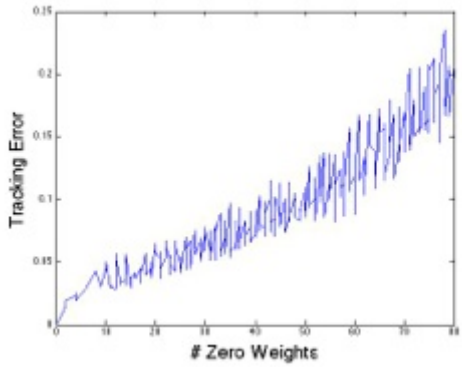
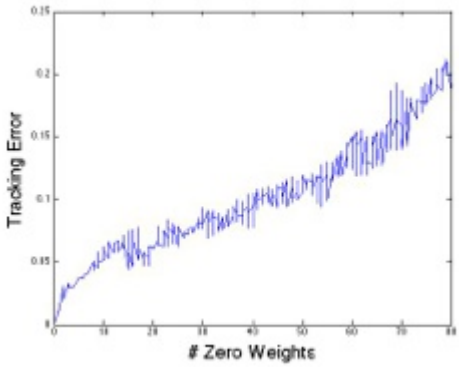



**Figure 10 - Monte Carlo (Spectral)**



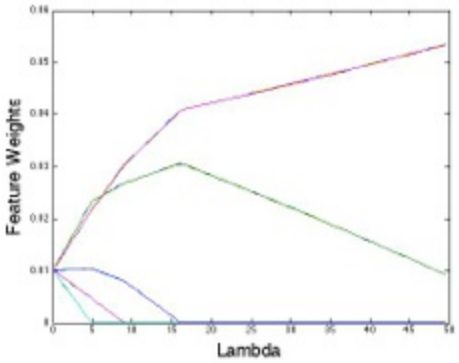

Figures [FIGURE] through [FIGURE] show a different pattern in the Zeros-Error - these aggressive variations in Tracking Error visible are caused by the full-search approach taken – by considering every single combination possible rather than only chose a local optimal, some combinations may have a higher training error even when they may have a lower sparsity. It is worth mentioning that the 'optimal' value would be at each local nadir in the graph line.

### 3.5 Sparse Group Models

Theory behind Sparse Group Models suggest that it is possible to control sparsity between and within groups [13]. In this section, the implementation will be initially focused on showing the effects of this model when applied to financial data.

### *3.5.1 Group Lasso Regression Model*

**Figure 11 - FTSE100**




Similar to our individual-feature Lasso implementation, we are interested in studying the behavior of the tracking error as the value of lambda increases – that is, as sparsity is induced throughout groups of stocks.

It is worth mentioning that, similar to the simple lasso model, as the value of lambda increases, more correlated features obtain higher weights, and less correlated features are pulled towards zero – the only difference is that in the case of this Group Lasso, features are groups consisting of equally weighted stocks. This can be observed in figure 11, where the individual feature-weights are plotted against a variable value of lambda. It is possible to observe how all the feature weights within a group are drawn towards an equal value, and groups of stocks that are less correlated to the index are pulled towards zero as lambda increases.

Now that an intuition has been provided on the general behaviour of this model, it is possible to focus on the actual implementation in financial data. Figures 12 and 13 consist of experiments that test the behavior of a variable Lambda against Tracking Error and Number of Zeros.

**Figure 12 - FTSE100**

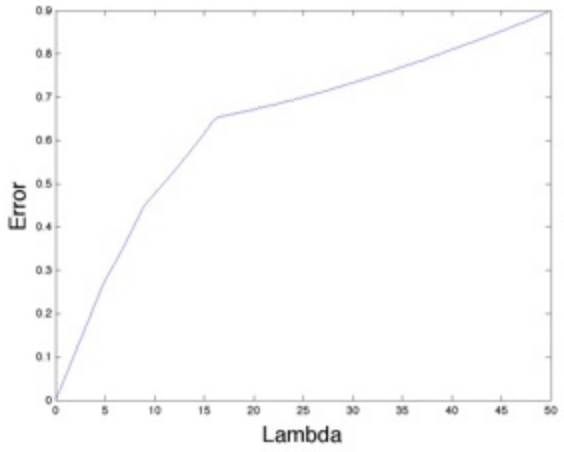



**Figure 13 - FTSE100**

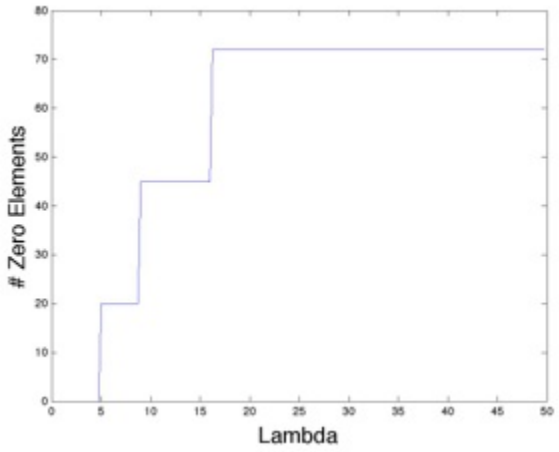



Figure 13 provides a visualization of the effect of the L2-norm constraint which induces sparsity between groups. With this in mind, it is possible to introduce a main limitation of this model: the L2-norm constraint imposed in the groups forces all the weights to be exactly the same within each group, and if one value in the

group becomes zero, then all the elements in such group will also become zero. Fortunately, it is possible to overcome this limitation through an L1-norm constraint, as it will be introduced in the next section.

### *3.5.2 Sparse Group Lasso*

As it was mentioned previously, a limitation present in the group lasso model is that the L2 constraint induces sparsity only between groups, but not within groups. What this means is that if one element in a group becomes zero, all the elements in that group would also become zero. The Sparse Group Lasso algorithm introduces back again the L1-norm constraint found in the simple feature Lasso model which induces sparsity within the groups (on feature level).

It should be noted that the values for $\lambda_1$ and $\lambda_2$ needed tweaking in order to obtain reliable results for the objectives of this paper. The behaviour of the model varies a lot depending on the value of these coefficients. This is why a strong understanding on the between- and within-group sparsity flexibility of this model in order to implement it in financial datasets effectively.

To begin our experiments, we observe the effects of a fixed feature sparsity coefficient ($\lambda_2 = 1$), over a variable group sparsity coefficient ($\lambda_1 += 0.00005$ per iteration) starting at zero, and increasing towards $\lambda_2$. These experiments will be initially applied to our FTSE100 dataset, with stock sectors as our grouping classification.

$\underline{\lambda_1 \text{ increasing towards a fixed } \lambda_2}$

**Figure 14 - FTSE100**

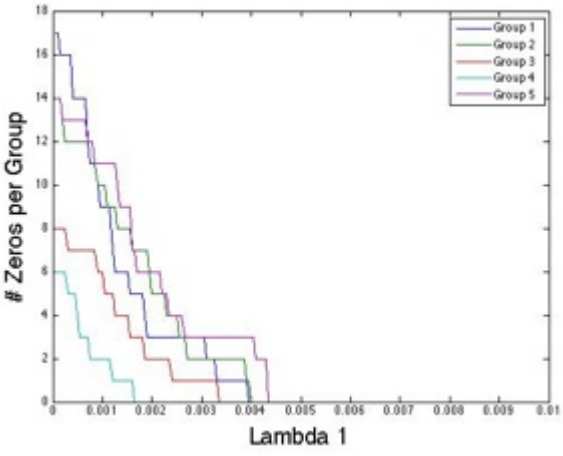
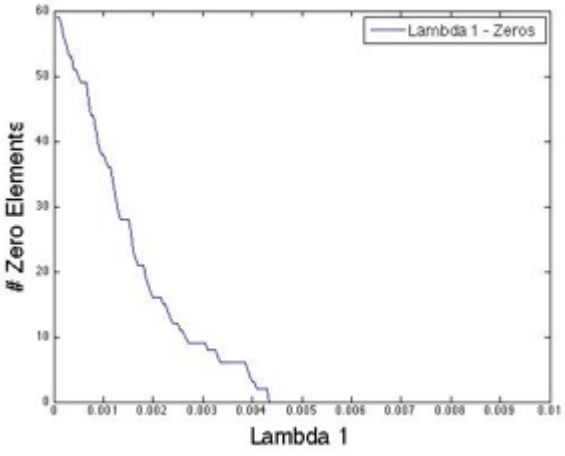



**Figure 15 - FTSE100**




In figures 14 and 15 above, it can be observed that as the value of $\lambda_1$ increases (i.e. group sparsity), we can observe that feature level sparsity decreases – this is due to values within groups being pushed towards a balance due to a stronger group lasso

coefficient. Given that the value of $\lambda_2$ does not change, there is no proportionate increase in feature-level sparsity to balance the group-sparsity introduced, which results in no sparsity at all, hence pushing all values towards a balanced weight of $1/n$. This behaviour is shown explicitly in figures 16 and 17 below.

$\underline{\lambda_1 \text{ increasing towards fixed } \lambda_2}$

**Figure 16 - FTSE100**

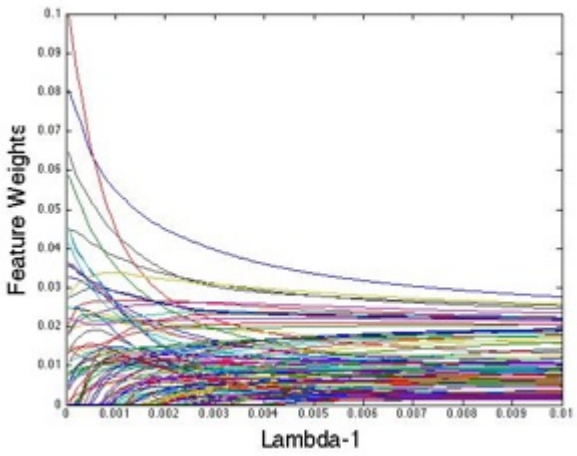
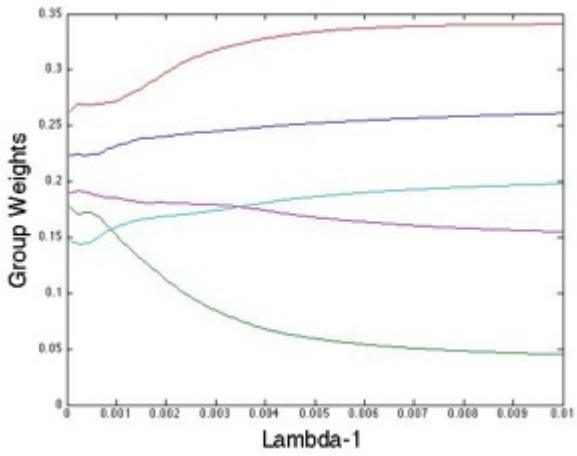



**Figure 17 - FTSE100**




The simple lasso behaviour can be observed on the first iterations of figure 16 on the left. It can also be observed that as $\lambda_1$ increases towards $\lambda_2$, zero weighted values obtain larger values and vice versa – tending towards 1/n as it was mentioned previously. Consequently, it should be intuitive that due to this, group weights will tend towards $p_g/n$ - this behaviour is shown in figure 17 on the right. From the small magnitude of increments in $\lambda_1$ on each iteration, it can be noticed that the group sparsity inducing coefficient is predominant, and a slight modification can cause a big variation in the behaviour of the model.

With this in mind, it is possible to state that a linearly proportionate increase in both $\lambda$'s is required for an effective application of this model, and the proportion chosen is crucial as stated in the model proposed in [13], especially when applied to financial datasets. This piece of knowledge allows for a more flexible approach towards Index Tracking, as these following experiments will study how this 'proportion' affects Tracking Error, allowing us to obtain more clear results on the effect of grouping data in financial datasets.

With these insights on the behaviour of the model we can approach our portfolio optimization problem more insightfully. In the application of Sparse Group Lasso in portfolio index tracking, our objective is to identify the groups and features that are most correlated that of the Market Index – this, while inducing sparsity between and within groups. From our previous definitions, we can observe that that this can be achieved with a with a proportionate amount of $\lambda_1$ and $\lambda_2$.

The following experiments of three different experiments where we observe the proposed approach of proportionate $\lambda$'s by having $\lambda_1 = \lambda_2/\Omega$, such that $\lambda_2 = [0.01,20]$ and $\Omega = \{ 500, 1000, 5,000, 10,000 \}$. Before presenting the results it is important to make a comparison against the behaviour of feature weights in the sparse models precented earlier. This is why we recall the feature weights behaviour of a Lasso model in figure 18, and the feature weights behaviour of a Group Lasso model in figure 19.

**Figure 18 – Lasso ($\lambda_1$=0)**

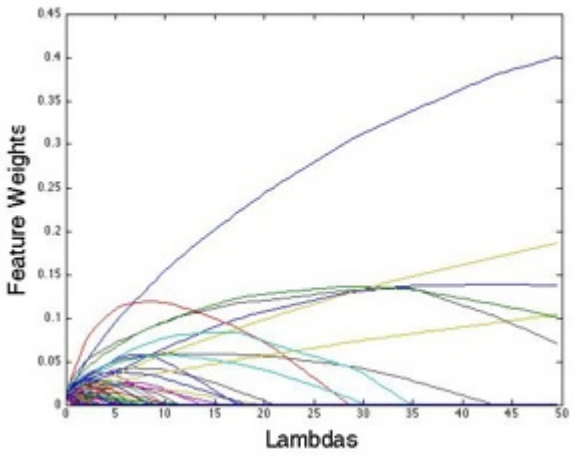
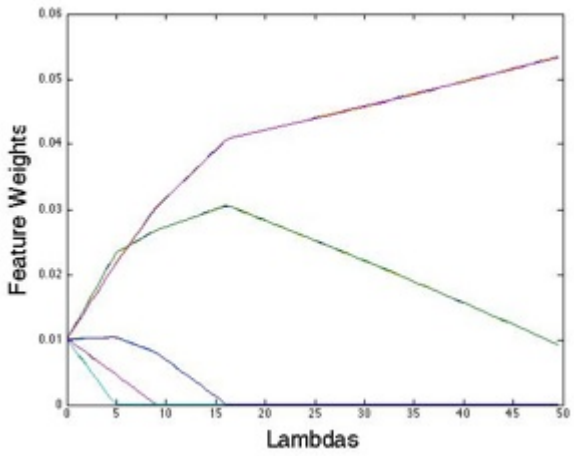



**Figure 19 - Group Lasso ($\lambda_2$= 0)**




These two models will provide a perspective on the effects of $\Omega$ in the FTSE100 financial dataset. With this in mind, feature weights are plotted against Lambda for different values of $\Omega$ in figures 20 to 23 below.

**Figure 20 - $\Omega$ = 10,000**

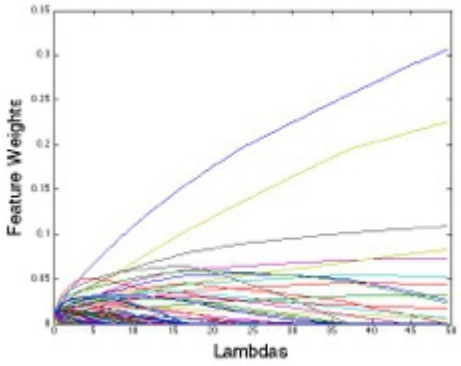



**Figure 21 - $\Omega$ = 5,000**

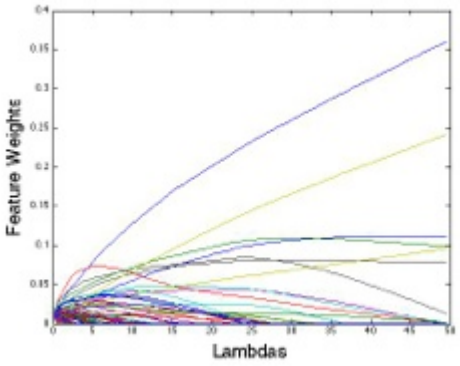


**Figure 22 - Ω = 1,000**

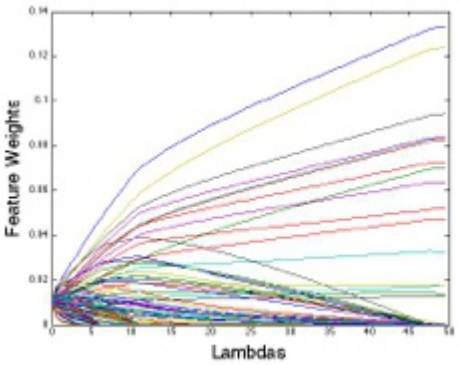
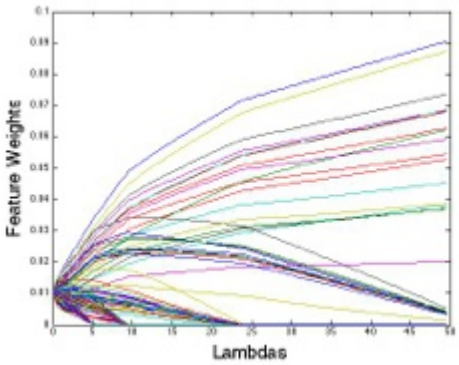



**Figure 23 - Ω = 500**




Figures 20 to 23 show the effects of a variable lambda on the feature weights for different values of Ω. The most important thing to observe is the transition between a group lasso model towards a simple lasso model.

It is also important to offer a perspective the behaviour of the group weights for different values of Ω – this is shown in figures 24 to 27 below.

**Figure 24 - Ω = 10,000**

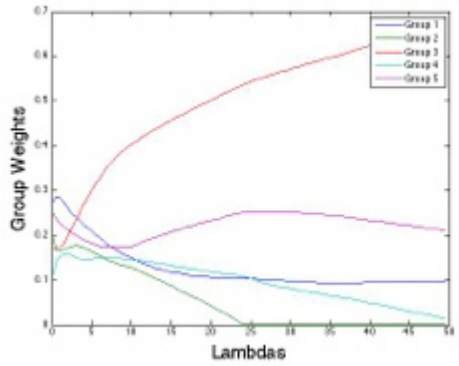



**Figure 25 - Ω = 5,000**

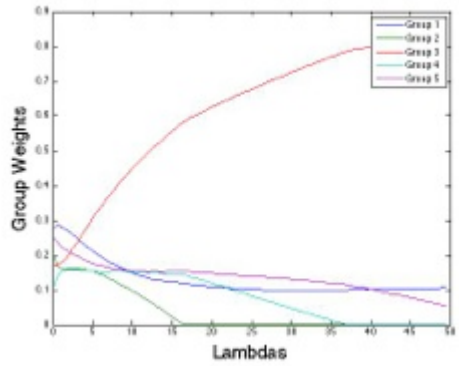



**Figure 26 - Ω = 1,000**

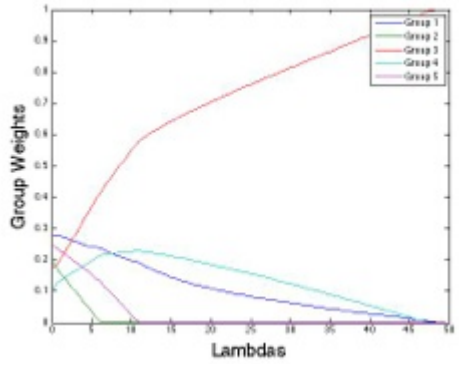
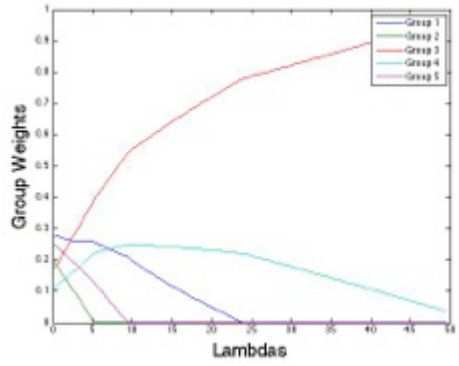



**Figure 27 - Ω = 500**



At a glance, between-group sparsity can be observed immediately on smaller values of Ω – whole groups are pulled to zero as Ω grows smaller. Likewise, the higher our value of Ω, the less our model takes groups into consideration.

Going even deeper, if we observe both, the group and feature weight for values $\lambda_2 = [5,10]$, we can observe that for higher values of Ω, groups that would have otherwise become zero in a group lasso model, are instead induced with within-group sparsity, allowing stocks that are most correlated to the index to stand out from the group while non-important stocks are kept at zero.

Another very important point we should consider in regards to this regression model is the effect of Ω in the tracking error. Figure [FIGURE] contains a table with the tracking error values for each of the values of Ω.

**Figure 28 – Effect of Ω**

| Ω = 500 | | Ω = 1,000 | | Ω = 5,000 | | Ω = 10,000 | |
|---|---|---|---|---|---|---|---|
| **# Zeros** | **Group Lasso** | **# Zeros** | **Error** | **# Zeros** | **Error** | **# Zeros** | **Error** |
| 0 | 0.0000 | 0 | 0.0000 | 0 | 0.0000 | 0 | 0.0000 |
| 0 | 0.1614 | 0 | 0.1880 | 31 | 0.2822 | 38 | 0.3119 |
| 0 | 0.2919 | 6 | 0.3153 | 47 | 0.4132 | 55 | 0.4483 |
| 20 | 0.3966 | 24 | 0.4225 | 54 | 0.5098 | 62 | 0.5507 |
| 45 | 0.4870 | 25 | 0.5167 | 58 | 0.5969 | 68 | 0.6414 |
| 45 | 0.5311 | 49 | 0.5782 | 64 | 0.6728 | 73 | 0.7127 |
| 45 | 0.5733 | 50 | 0.6171 | 66 | 0.7405 | 74 | 0.7808 |
| 45 | 0.6176 | 51 | 0.6515 | 74 | 0.8004 | 76 | 0.8447 |
| 45 | 0.6645 | 52 | 0.6844 | 75 | 0.8465 | 77 | 0.9065 |
| 45 | 0.7115 | 52 | 0.7166 | 75 | 0.8920 | 77 | 0.9686 |
| 72 | 0.7469 | 52 | 0.7486 | 77 | 0.9342 | 81 | 1.0270 |
| 72 | 0.7687 | 52 | 0.7802 | 77 | 0.9745 | 83 | 1.0673 |
| 72 | 0.7911 | 52 | 0.8114 | 78 | 1.0155 | 85 | 1.1019 |
| 72 | 0.8137 | 52 | 0.8431 | 80 | 1.0555 | 86 | 1.1355 |
| 72 | 0.8365 | 53 | 0.8777 | 80 | 1.0948 | 87 | 1.1660 |
| 72 | 0.8605 | 53 | 0.9127 | 84 | 1.1323 | 88 | 1.1954 |
| 72 | 0.8849 | 53 | 0.9483 | 84 | 1.1630 | 88 | 1.2260 |
| 72 | 0.9098 | 54 | 0.9843 | 84 | 1.1938 | 88 | 1.2583 |
| 72 | 0.9352 | 54 | 1.0210 | 85 | 1.2236 | 88 | 1.2928 |
| 72 | 0.9614 | 66 | 1.0576 | 85 | 1.2514 | 88 | 1.3288 |

The effect of Ω can be observed in the previous table – more aggressive movements in the number of zeros show a stronger group lasso behavior, while higher granularity in zeros shows a stronger simple lasso behavior.

# 4 Analysis

Now that the models in this paper have been tested and results have been collected, it is possible to compare their characteristics side by side and provide an insight on their potential applications in the financial world.

## 4.1 Quantitative Model Comparison

When it comes to portfolio optimization, and in specific index tracking, accuracy, sparsity, and speed are the most important attributes to consider financial tools for trading, investment, etc. In this section, quantitative analysis is made on the results obtained throughout the implementation of the models and approaches considered.

### *4.1.1 Accuracy*

The results for all models in the form of number of zeros against tracking error have been compiled, and can be found in the annex in the repository. In order to provide an intuitive perspective on these results, figures 29 and 30 below provide a summary of the results obtained for all the models in this paper for our FTSE100 dataset with grouping categories as both, sectors and spectral.

**Figure 29 - FTSE100 (Sectors)**

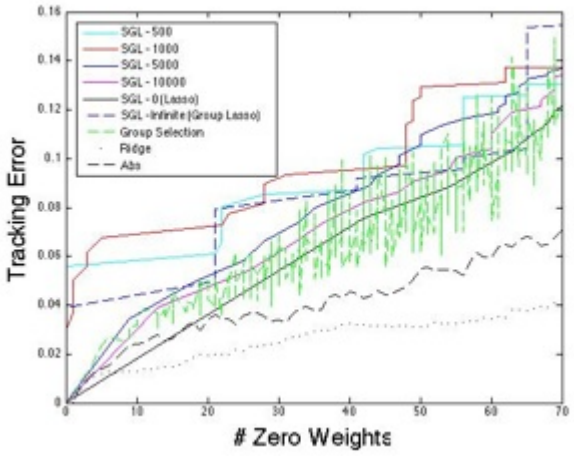
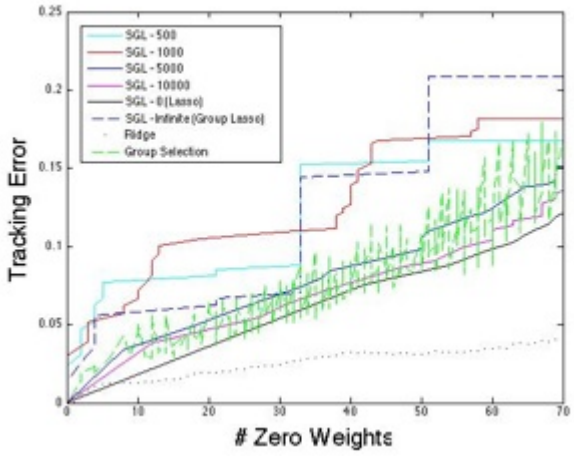



**Figure 30 – FTSE100 (Spectral)**




It was evident throughout this paper that the most accurate regression model, based in our results, is the *Ridge* (L0+L2-norm). These findings reassure the conclusions

stated in [1] and [2], where the L0+L2-norm model (i.e. the Ridge regression model with an L0-norm constraint) has proven to be a very accurate approach to Market Index Tracking. In terms of accuracy, the *Ridge* model was followed by the *Lasso*, *Group Selection* and *Sparse Group Lasso* with $\Omega = 10{,}000$. The *Abs* model was included as it will be discussed further. It should be noted that the GFS approach outperformed all the other approaches in terms of accuracy (except for the first 5 zero induced weights, where the Lasso model showed a slightly better performance).

In regards to the *Sparse Group Lasso* (SGL) and *Group Selection* approaches, figures 31 and 32 show a concise summary of the results obtained throughout this paper. Once again, the group selection approach has shown to be the most accurate model, followed by the Sparse Group Lasso with $\Omega = 10{,}000$ and $\Omega = 5{,}000$.

**Figure 31 – FTSE100 (Sectors)**

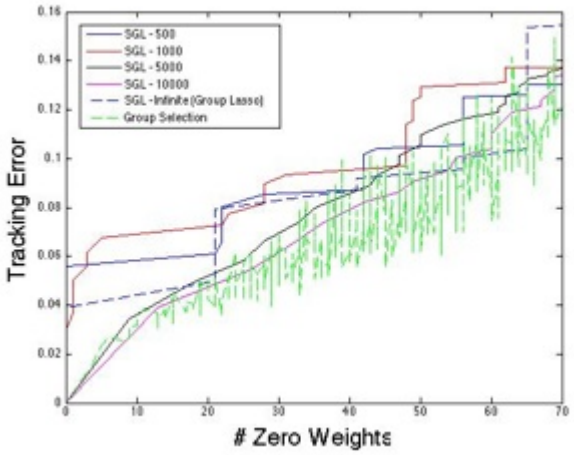
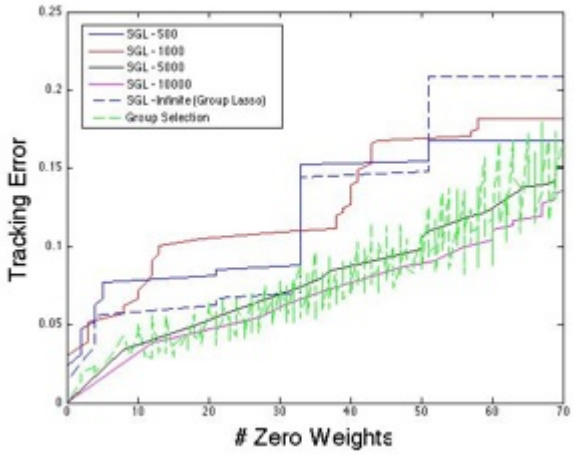



**Figure 32 – FTSE100 (Spectral)**




### *4.1.2 Sparsity*

One of the main concerns in this paper was to implement and study the effects of group and feature level sparsity through regression models when applied to portfolio optimization problems. The following experiments are applied to the FTSE100 dataset with Sectors as the input groups.

Based in our studies, it is important to analyze the results from both, feature and group perspectives. Initially, the results of the individual stocks for the FTSE100 dataset are be analyzed against an increasing sparsity. The first question introduced is the reason for the *Ridge* regression model being the best performing in terms of accuracy, and the reason for the Abs model being the worse. To provide an insight on the answer, figures [FIGURE] and [FIGURE] show feature weights of the *Abs* and *Ridge* models as sparsity increases in the chosen portfolio.

**Figure 33 - FTSE100 (ABS)**

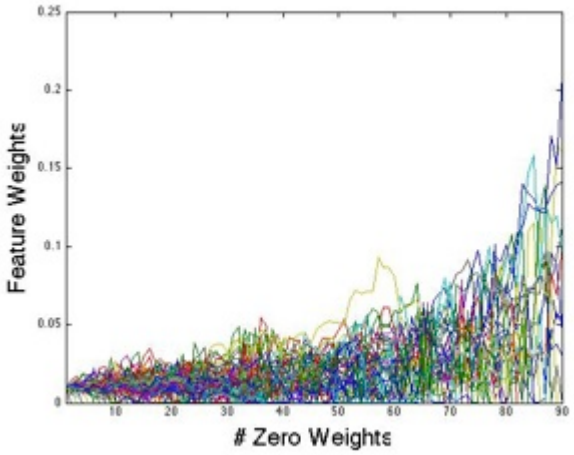
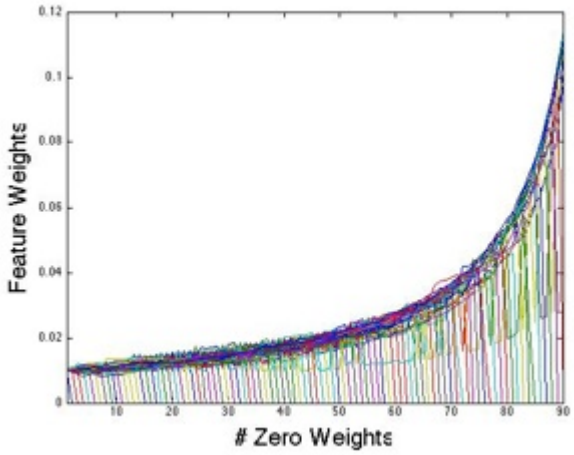



**Figure 34 - FTSE100 (RIDGE)**




At first sight, the behavior of the cost functions of the *Abs* and *Ridge* regression models are explicit in figures 33 and 34 above. While *Abs* tries to minimize the Tracking Error at all costs by adjusting varied weights, *Ridge* choses the stocks that represent the Index best, and pushes all the weights towards a balance due to the L2-norm constraint that penalizes the model on higher standard deviations. This shows that a balanced portfolio consisting of the most correlated subset of stocks will represent a market index better than a rather unbalanced one.

With this in mind, figures 35 and 36 show the behaviors of the *Lasso* and *Group Lasso* models against the individual weights of the stocks in the portfolio. The way sparsity is induced in these models is visible in both figures. Based on the characteristics of the *Ridge* model, it can be said that the *Lasso* model obtains more accurate than the Group Lasso as it allows for more correlated features to have higher weights than less correlated ones. This is however a limitation as well, as the *Ridge* model shows that a uniform value in most correlated stocks will obtain the best results. Although the group lasso provides the characteristic of uniformity, given that this constraint is applied to whole groups, less correlated stocks would also get a uniform value.

**Figure 35 - FTSE100 (LASSO)**

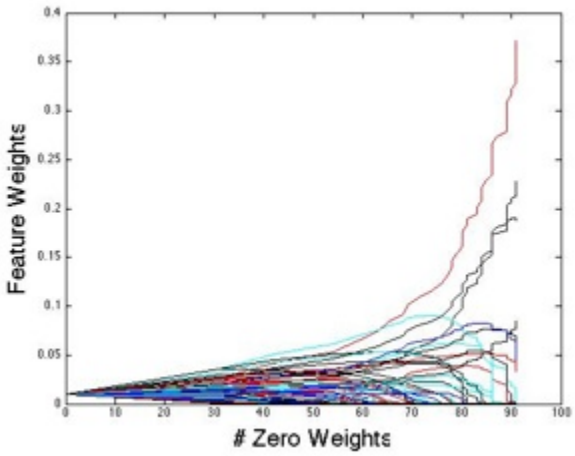
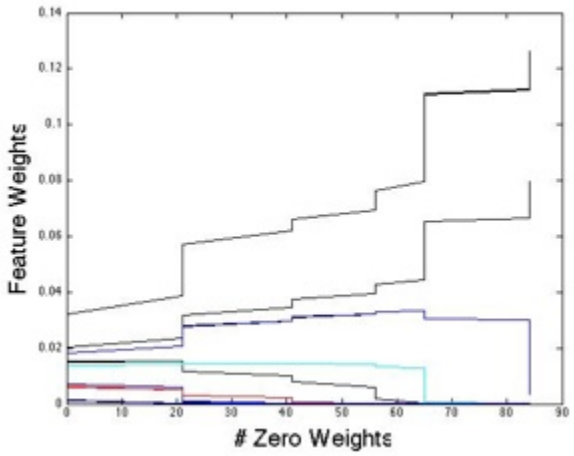



**Figure 36 - FTSE100 (GROUP LASSO)**

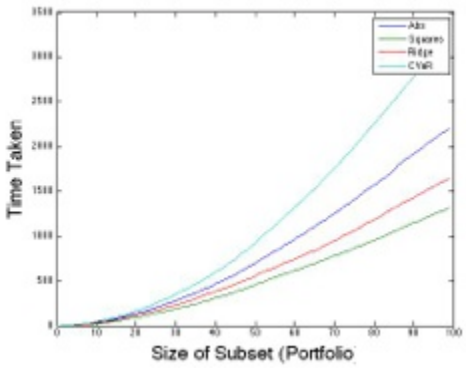



Theoretically, this is when the Sparse Group Model should provides the desired characteristics shown by the *Ridge* model – namely, choosing a subset of the most correlated values, and setting the weights almost uniformly. With this said, the Sparse Group Lasso model proposes to a proportionate amount of between/within group level sparsity, allowing more correlated features within groups to obtain higher values, and less correlated ones to become zero.

Based on the results obtained in this paper, however, this model did not prove to be a more accurate than the simple lasso or group lasso models alone. The observations made from the Sparse Group Lasso model where that when group sparsity is induced to a simple lasso model, the features of the lasso model can only approach towards the uniform group weight relative to their respective Group Lasso model weights (And vice versa). What this means is that between/within group sparsity introduced in the weights is constrained to the behavior of its respective Simple Lasso ($\lambda_1 = 0$) and Group Lasso($\lambda_2 = 0$), and sparsity obtained is only the transition paths of the weights between these.

### *4.1.3 Speed*

Speed in finance is always one a main concerns – financial institutions constantly strive to be the fastest in the market – to a point where nanoseconds are extremely important.




For this reason it is important to analyze the speed of the approaches taken. Starting with the GFS algorithm, it is possible to observe that all available stocks from the ones that have been chosen have to be run with a regression algorithm. This gives us a triangular number time complexity as $n + (n-1) + \cdots + 2 + 1 = n * (n+1)/2 = O(n\text{^}2)$. This behavior can be shown in figure [FIGURE], where we plot the elapsed time to execute the GSF algorithm for the FTSE100 index. This time complexity is quite a concern, as it adds up to the time complexity of the optimization algorithm.

The rest of our models are affected only by the time complexity of the algorithm used to solve the minimization problem. This is a characteristic that makes sparsity inducing models attractive – more specially, the Lasso and it's variations.

# 5 Expansion

Results in regards to accuracy were not as expected, however, the Sparse Group Lasso model does provide the flexibility required to induce between-group and within-group sparsity, and this model is currently being applied in numerous fields, including biology, aeronautics, etc. In this section we aim to propose a new concept to encourage further research on this model.

Before proposing the new model it is very important to emphasize that groupings have proven to discover deeper information in financial datasets, as they not only show how correlated groups interact with the data, but also allows us to use groupings as an extra dimension of data that allows for more insightful regression. Specific to finance, it's important to recall that these models impose portfolio diversification by nature. Given that with regular regression models, such as Abs, Squares, Ridge and CVaR, the only way to achieve this is through an algorithmic approach – this is when the SGL seems attractive, as its execution time is much faster, and offers more flexibility.

## 5.1 Multiple Sparse Group Regression (Proposed Concept)

In this section we would like to briefly propose a concept to encourage further research in the topic. The current sparse group model, is limited to only one category/type of groupings, and would not allow to consider multiple categories/types of groups. This section contains a suggestion for a model which aims to expand the concept of sparse group regression into more than one single category of groupings, for data that might consist of multiple correlated groups. It is important to emphasize that time did not allow for analyzing a proof for this model, and is more of a concept that aims to encourage further research in this area of Machine Learning.

To provide context, let's take a portfolio consisting of commodities, options currencies, stocks, etc. The financial instruments comprising this portfolio can be grouped not only by their nature (i.e. stocks, currencies, bonds, etc), but also by industry, sector, or based on historical return or volatility. The concept proposed aims to allow consideration on multiple grouping categories/types rather than just one.

## 5.2 Notation

In order to introduce this formula, we first need to introduce few concepts. We would like to recall the constraint proposed, $\left(\forall g \in \{1, \dots, m\}.\ \widehat{\widehat{\boldsymbol{\pi}}}\left[\sim \boldsymbol{\Psi}_{\boldsymbol{g}}, g\right] = 0\right) \Longrightarrow \left(\boldsymbol{\pi} = sum\left(\widehat{\widehat{\boldsymbol{\pi}}}, 1\right)\right)$.

A problem with introducing multiple categories of groups is that our constraint of $\widehat{\widehat{\boldsymbol{\pi}}}\left[\sim \boldsymbol{\Psi}_{\boldsymbol{g}}, g\right] = 0$ would be violated, as there would be overlaps between groups. An example of this would be groups of stocks by type of instrument (stocks, currencies, etc), which would overlap if these are grouped further into sectors or industry.

A simple solution for this would be to subdivide overlapping groups, which would keep the constraint imposed earlier, and would just give us a larger number of groups These stocks would all be considered as completely different groups, but to avoid this, a vector coefficient **w** can be introduced which would scale features from each group according to their groups. There should however exist a more elegant approach to this problem that could allow us to consider whole groups without having to break them into subsets of groups. This vector **w** will however be actually used in order to provide a scale to each category as required.

This model is based on the new notation that was presented when the group lasso model was introduced - Namely the formulation proposed introduced a constraint that allowed us to compute the formulation with only matrix multiplications – together with the model proposed by [12] defined as follows:

$$\min_{\overline{\boldsymbol{\pi}} \in \mathbb{R}^{\boldsymbol{n}}} \quad \left\| \boldsymbol{I} - \sum_{g=1}^{m} \boldsymbol{R}_g^{\mathrm{T}} \overline{\boldsymbol{\pi}}_g \right\|_2^2 + \lambda_1 \sum_{g=1}^{m} \sqrt{p_g} \left\| \overline{\boldsymbol{\pi}}_{\boldsymbol{g}} \right\|_2 + \lambda_2 \|\boldsymbol{\pi}\|_1$$

To begin with, we would like to briefly introduce very briefly a new notation. We will have K categories containing groups of size $m_k$, our indexing notation will be expanded to a matrix $\boldsymbol{\psi}$ consisting of rows containing the logical indexing for all groups in all categories, this can be defined as:

$$\boldsymbol{\psi} = \begin{bmatrix} | & | & | & | & | \\ \text{logical}(\boldsymbol{\Psi}_{\text{k,g}}^{\boldsymbol{T}}) & \dots & \text{logical}(\boldsymbol{\Psi}_{2,1}^{\boldsymbol{T}}) & \dots & \text{logical}(\boldsymbol{\Psi}_{\text{k,g}}^{\boldsymbol{T}}) \\ | & | & | & | & | \end{bmatrix}$$

Our new set $\boldsymbol{\Psi}_{k,g}$ denotes the indexes for the stocks contained in group g of category k. With this, we would be able to introduce our variable $\overline{\boldsymbol{\pi}}_{k,g} = \boldsymbol{\pi}\left[:, \boldsymbol{\Psi}_{k,g}\right]$, where $\overline{\boldsymbol{\pi}}_{k,g}$ would be the subset containing the stocks for group g in category k. Now, although our constraint will be removed, we can see form our previous observations that $\sum_{g=1}^{m} \boldsymbol{R}_g^{\mathrm{T}} \overline{\boldsymbol{\pi}}_g = sum(\boldsymbol{R}^{\mathrm{T}} \widehat{\widehat{\boldsymbol{\pi}}}, \boldsymbol{2}) = \boldsymbol{R}^{\mathrm{T}} \boldsymbol{\pi}$. What this tells us is that the weightings in

our original variable to be learned, $\boldsymbol{\pi}$, will never suffer any change in terms of logic structure even when groups are introduced. Before introducing the model we would like to add a placeholder to our second coefficient as $G = sum(\sqrt{}(p\text{^T}) * \sqrt{}(sum(\widehat{\overline{\boldsymbol{\pi}}}\text{^}2,2))) = \sum_{g=1}^{m} \sqrt{p_g} \left\| \overline{\boldsymbol{\pi}}_{\boldsymbol{g}} \right\|_2$. This would really simplify our formulation, and will allow us to revisit this coefficient further on, which is the most important one. This allows us to define and expand the following model:

$$\min_{\boldsymbol{\pi} \in \mathbb{R}^{\boldsymbol{n}}} \sum_{k=1}^{K} \left( \left\| \boldsymbol{I} - \sum_{g=1}^{m} \boldsymbol{R}_g^{\mathrm{T}} \overline{\boldsymbol{\pi}}_{k,g}) \right\| + \lambda * \mathrm{G} + \lambda_2 \|\overline{\boldsymbol{\pi}}_k\|_1 \right)$$

However, from what we mentioned earlier, the weights for each stock will be the same on any state of the execution. What this means is that $\boldsymbol{\pi}[:, \boldsymbol{\Psi}_{i,x}] = \boldsymbol{\pi}[:, \boldsymbol{\Psi}_{j,y}]$ if the stocks from group x in category I are the same than the stocks from group y in category j. What this proves is that $\sum_{g=1}^{m} \boldsymbol{R}_g^{\mathbf{T}} \overline{\boldsymbol{\pi}}_{k,g} = \boldsymbol{R}^{\mathrm{T}} \boldsymbol{\pi}$ will always hold. Likewise, $\sum_{k=1}^{K} \lambda_2 \|\overline{\boldsymbol{\pi}}_k\|_1 = \sum_{k=1}^{K} \lambda_2 \|\boldsymbol{\pi}\|_1 = \mathrm{K}\lambda_2 \|\boldsymbol{\pi}\|_1$, as the sum of absolute values should not vary on any category, as all categories will be composed of the same stocks. We can also omit the K value as we can assume $\lambda_2$ will be able to scale as required. This would leave us with:

$$\min_{\boldsymbol{\pi} \in \mathbb{R}^{\boldsymbol{n}}} \sum_{k=1}^{K} \|\boldsymbol{I} - \boldsymbol{R}^{\mathrm{T}} \boldsymbol{\pi})\| + \lambda * \mathrm{G} + \lambda_2 \|\boldsymbol{\pi}\|_1$$

Similar to our absolute sum of values, the first coefficient would always be the same, we could replace the summation $\sum_{k=1}^{K} \|\boldsymbol{I} - \boldsymbol{R}^{\mathrm{T}} \boldsymbol{\pi})\|$ simply with $K * \|\boldsymbol{I} - \boldsymbol{R}^{\mathrm{T}} \boldsymbol{\pi})\|$.

This now leaves us with the main and only coefficient that induces the Group sparsity effect desired – namely:

$$G = sum\left( \sqrt{p^{\mathrm{T}}} * \sqrt{sum(\widehat{\overline{\boldsymbol{\pi}}}^2, 2)} \right) = \sum_{g=1}^{m} \sqrt{p_g} \left\| \overline{\boldsymbol{\pi}}_{\boldsymbol{g}} \right\|_2$$

This is basically the scaled sum of the norms of each of our groups, which in our situation would expand to:

$$\sum_{k}^{K} \sum_{g=1}^{m} \sqrt{p_{k,g}} \left\| \overline{\boldsymbol{\pi}}_{\boldsymbol{k,g}} \right\|_2$$

Now can see that that our problem has reduced to a simple sum of L2-norm of all our individual groups for each category. In order to adapt this to our previous formulation, we would need to build a vector $\hat{p} = (p_{1,1}, \dots, p_{k,g}) \quad \forall k, g.\ k \in K \wedge g \in \{1, \dots, m\}$, and a function norms($\mathbf{v}$, $\boldsymbol{\Psi}$) where it returns the L2-norm for a matrix of a set of indexes.

### 5.3 MSGR Model definition

This would allow us to define our model as:

$$\min_{\boldsymbol{\pi}\in\mathbb{R}^n} \quad K\|\boldsymbol{I}-\boldsymbol{R}^{\mathrm{T}}\boldsymbol{\pi}\,)\| + \lambda_1 \boldsymbol{w}\sqrt{\hat{p}}\, norm(\boldsymbol{\pi},\boldsymbol{\psi}) + \lambda_2\|\boldsymbol{\pi}\|_1$$

Assuming that we could also superscript logical indexes, this formulation can be defined as our previously introduced formulation:

$$\min_{\boldsymbol{\pi}\in\mathbb{R}^n} \quad K\|\boldsymbol{I}-\boldsymbol{R}^{\mathrm{T}}\boldsymbol{\pi}\,)\| + \lambda_1\sqrt{\hat{p}} * \sqrt{sum(\boldsymbol{\pi}[\boldsymbol{\psi}]^2, 2)} + \lambda_2\|\boldsymbol{\pi}\|_1$$

We would also like to recall the **w** vector mentioned previously that would provide a scaling to each specific category as required – this would allow for specific categories to have a higher effect than others:

$$\min_{\boldsymbol{\pi}\in\mathbb{R}^n} \quad K\|\boldsymbol{I}-\boldsymbol{R}^{\mathrm{T}}\boldsymbol{\pi}\,)\| + \lambda_1\sqrt{\hat{p}} * \boldsymbol{w}\sqrt{sum(\boldsymbol{\pi}[\boldsymbol{\psi}]^2, 2)} + \lambda_2\|\boldsymbol{\pi}\|_1$$

This new formulation now takes into account overlapping groups which are scaled by our constant **w** vector in order to take group regression characteristics to the next level.

It should be noted that if only one category of the weights of **w** is set to 1 and the rest to zero, this would be the same effect than the normal Sparse Group Lasso model.

Once again, it is necessary to mention that for as of the time of writing, time did not allow for a mathematical proof on the correctness or convexity of this model. This model was only proposed from what was learned from the sparse models, and from the observations and results obtained in the implementation.

# 6 Conclusion

To conclude, in this paper we took a new approach to portfolio optimization by implementing sparse inducing models to a Market Index, and furthermore, study the effects of sparse group regression models in financial datasets. The results obtained provided great insight on sparse inducing models, as well as on the effects of feeding grouping information of financial instruments (i.e. Sector, Industry, etc.) to these models.

This paper also compared the results of these group-level regression models against an existing feature-level approach proposed in [1] that achieves sparsity through a greedy forward-search algorithm.

After analysing the results obtained throughout the implementation, the following conclusions were reached.

## 6.1 Feature Selection

This approach consisted of finding a subset of stocks from a Market Index by minimizing tracking error, which is calculated by using the Sum of Absolute Values (*Abs*), Least Squares (*Squares*), Ridge Regression (*Ridge*) or Conditional Value at Risk. Results proved to be very accurate to predict the Market Index – which supports the conclusions in [1] and [2] that inspired this research. The most accurate regression model in this paper proved to be the *Ridge* model (i.e. the L0+L2-norm constrained model[1]), which was followed by *CVaR*, *Squares*, and finally, *Abs*. From the observations, the behavior of the *Ridge* model consisted of selecting the most correlated stocks to the market index, and assigned balanced weightings to all the stocks chosen (imposed by the L2-norm constraint). The *Abs* model in the other hand proved to be the worse model, and although it showed very similar choice of stocks in the subset, it assigned a varied proportion to each stock, making its prediction less accurate. Speed was an issue with this model, as the time complexity of only the algorithm is squared on the number of features, and the minimization model is polynomial on both, the number of features and observations. This became a problem even for small Market Indexes, as computational time is infeasible for practical purposes.

## 6.2 Group Selection

Section [SECTION] covered the expansion of the greedy forward search approach proposed in [1] into a full-search group selection approach for financial datasets. Results showed an acceptable Index Market prediction, however, as with the previous model, this approach was very limited to time complexity – although it was

possible to implement this approach in this paper, if a situation arises with a slightly greater number of groups (i.e. greater than 15) the application of this approach would be impossible.

### 6.3 Lasso Model

In section [SECTION], the Lasso regression model implemented in financial datasets, and provided very insightful results on the potential of the application of sparse models in financial data. The tracking error from the predictions made by these model was acceptable, and the execution time was only a fraction of the approaches taken previously, which makes this an attractive alternative for practical applications in the finance. These results encouraged further research in these type of models, including the Group and Sparse Group Lasso models.

### 6.4 Sparse Group Lasso

The Sparse Group Lasso implemented in section [SECTION] provided a lot of insight on the effects caused when groupings of stocks are taken into consideration. This model provides flexibility allowing for sparsity at group level while still allowing for feature-level sparsity. After analyzing the results obtained when implementing this model in our Index Tracking problem, it was observed that the model did not prove to be as accurate as it's Simple Lasso nor Group Lasso counterparts, however it allowed for a lot of flexibility when it came to between-/within-group level sparsity.

### 6.5 Effects of Group and Sparse models in Financial Datasets

It is possible to state that sparse models prove to have potential for practical implementation in financial markets. Observations can show that the exceptional execution speed of the sparse inducing models, accounted for the small lack in prediction accuracy. While sparse models didn't prove to be as accurate as model selection approaches, their time complexity consists of only the one from the sparse regression algorithm [16].

Finally, it was observed that it is possible to control sparsity within and between groups in financial datasets when stock sector information is fed to the Sparse Group models. It was observed that the models weren't as accurate as their pure lasso or group lasso counterparts, the flexibility provided to control between-group and within-group sparsity, together with the fast execution time, will certainly prove useful in practical applications such as portfolio risk diversification.

Finally, an experimental model is proposed in section 5, which suggests taking into account multiple groupings/categories in regression as opposed to just one. This experimental model is referred to as the 'Multiple Sparse Group Lasso model'.

# 7 References


[1] Takeda, A., Niranjan, M., Gotoh, J. Y., & Kawahara, Y. (2013). Simultaneous pursuit of out-of-sample performance and sparsity in index tracking portfolios. *Computational Management Science*, *10*(1), 21-49.

[2] Markowitz, H. (1952). Portfolio selection*. *The journal of finance*, *7*(1), 77-91.

[3] Philippe J. (1996) Risk2: Measuring the Risk in Value at Risk. *Financial Analysts Journal* , Vol. 52, No. 6 (Nov. - Dec., 1996), pp. 47-56

[4] Rockafellar, R. T. (1997). *Convex analysis* (Vol. 28). Princeton university press.

[5] Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of risk*, *2*, 21-42.

[6] Takeda, A., Gotoh, J. Y., & Sugiama, M. (2010, August). Support vector regression as conditional value-at-risk minimization with application to financial time-series analysis. In *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on* (pp. 118-123). IEEE.

[7] Xiao, J. Y. (2001). Return to RiskMetrics: the evolution of a standard. *RiskMetrics Group*.

[8] Brodie, J., Daubechies, I., De Mol, C., Giannone, D., & Loris, I. (2009). Sparse and stable Markowitz portfolios. *Proceedings of the National Academy of Sciences*, *106*(30), 12267-12272.

[9] Yuan, M. (2007). Model selection and estimation in the Gaussian graphical model. Biometrika.

[10] Benjamin F. (Jan., 1996) The Journal of Business, Vol. 39, No. 1, Part 2:Supplement on Securty Prices, pp. 139-190.

[11] Stephen L. Meyers. The Journal of Finance, Vol 28, No. 3 (Jun., 1973), pp. 695-705

[12] Friedman, J. Hastie, T. Tibshirani, R. (2010, Feb). A note on the group lasso and a sparse group lasso. Cornell University.

[13] J. Hastie T. Tibshirani, R. (2013). Vol. 22, No. 3, A Sparse Group Lasso (pp. 231-245)

[14] Beck, A. Teboulle, M. (March 2009). A Fast Iterative Shrinkage-

Thresholding Algorithm for Linear Inverse Problems. *SIAM J. Img. Sci.* 2, 1 183-202.

[15] Francis B. Rodolphe J. Julien M. (2011). Optimization with Sparsity-Inducing Penalties (Foundations and Trends(R) in Machine Learning). Now Publishers Inc., Hanover, MA, USA.

[16] Least Squares Optimization with L1-Norm Regularization (2005, Dec).