Title
Constructing sparse and fast mean reverting portfolios

Permalink
https://escholarship.org/uc/item/66t0n3qj

Author
Long, Xiaolong

Publication Date
2014

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

Constructing Sparse and Fast Mean Reverting Portfolios

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Mathematics

by

Xiaolong Long

Dissertation Committee:
Professor Knut Sølna, Co-chair
Professor Jack Xin, Co-chair
Associate Professor Long Chen

2014

Dedication

To my father Baijing Long, my mother Yanhua Wang and my cousin Qing Wang.

Table of Contents

List of Figures **vi**

List of Tables **vii**

Acknowledgements **viii**

Curriculum Vitae **ix**

Abstract of the Dissertation **x**

Introduction **1**

1 Basic Definitions and Theories **5**

1.1 Generalized Eigenvalue and Eigenvector 5

1.2 Vector Autoregressive Model 6

1.2.1 Model Form 6

1.2.2 Stationarity 6

1.2.3 Estimation of ρ and β 7

1.3 Order of Integration and Cointegration 7

1.4 Difference of Convex Functions Programming 8

1.5 Ornstein-Uhlenbeck Process 9

1.6 LASSO and Least Angle Regression 11

2 Proxies of Mean Reverting Property **13**

2.1 Predictability 13

2.2 Direct OU estimator 14

3 Sparse Optimization Problem and Related Algorithms **16**

3.1 Sparse Optimization Problem 16

3.2 Algorithms 17

4 Optimization Problem Based on the Ratio of l_1 and l_2 Norms **18**

4.1 Motivation 18

4.2 Constraint Problem 18

4.2.1 Formulation 18

4.2.2 Algorithm 19

4.3 Penalized Optimization Problem 21
4.3.1 Formulation 21
4.3.2 Algorithm 22
4.4 Analysis of the ratio of l_1 and l_2 penalty 23

5 Optimization Problem Based on the l_1 Norm and Prior Knowledge 32
5.1 Motivation 32
5.2 Formulation 33
5.3 Theory of Recovering the Global Optimizer 33
5.4 Algorithms 35
5.4.1 Algorithm for Finding r_* 35
5.4.2 Algorithm for Solving (5.1) 36
5.4.3 Algorithms for Solving (5.4) 37

6 Recover the Fastest Mean Reverting OU Process 39
6.1 Motivation 39
6.2 Formulation 1: Recover the OU Process with an Abnormal Starting Value . 39
6.3 Formulation 2: Recover the OU Process with a Jump Process 40
6.4 Methodology 40

7 Numerical Tests 42
7.1 Comparison of Two Mean Reverting Proxies 42
7.2 Tests of the Ratio of l_1 and l_2 Norms Approach 45
7.2.1 U.S. Swaps Data 45
7.2.2 S&P 500 Data 48
7.2.3 Discussion of the l_1/l_2 penalty 49
7.2.4 Comments on the Ratio of l_1 and l_2 Norms 50
7.3 Tests of the Methods in Chapter 5 51
7.3.1 In-Sample Tests on U.S. Swaps 51
7.3.2 In-Sample Tests on 100 Stocks 53
7.3.3 Out-of-Sample Tests on the U.S. Swaps 55
7.3.4 Out-of-Sample Tests on High Dimensional Simulated Data 56
7.4 Numerical Results of Recovering Fast Mean Reverting Process 58
7.4.1 Recover the OU Process with an Abnormal Starting Value 58
7.4.2 Recover the OU Process with a Jump Process 60

8 Conclusion 66

BIBLIOGRAPHY 68

APPENDICES 71
.1 More Numerical Results of the Methods in Chapter 5 71
.2 Estimation of the VAR(1) model 72

List of Figures

1.1 Sample paths of two Ornstein-Uhlenbeck processes 10

7.1 Trading Opportunities 44

7.2 Comparison of the performance of sparse mean reverting portfolios solved under different proxies 45

7.3 In-sample tests on the whole data set of the U.S. swaps. The x-axis shows the value of γ and the y-axis shows the estimated mean reversion coefficients and the trading opportunities 47

7.4 In-sample v.s. out-of-sample tests on the U.S. swaps. The x-axis shows the value of γ and the y-axis shows the average estimated mean reversion coefficients and the average trading opportunities 48

7.5 Comparison of the performance of sparse mean reverting portfolios of S&P 500 stocks. 49

7.6 One set of solutions on the S&P 500 data. The Y-axis represents the coefficients. 50

7.7 The values of the penalty term of two solutions sets 51

7.8 Mean reversion coefficients under different l_1 constraints 55

7.9 Cardinalities under different l_1 constraints 56

7.10 Trading Opportunities 57

7.11 Solutions under different l_1 constraints 58

7.12 In-Sample vs. Out-of-Sample Tests: λ 59

7.13 In-Sample vs. Out-of-Sample Tests: Trading Opportunities 60

7.14 Solutions under different l_1 constraints 61

7.15 Original Path v.s. Recovered Path When $k = 20$ and $n = 99$. 62

7.16 Original Path v.s. Recovered Path When $k = 20$ and $n = 639$. 63

7.17 Sample paths of the stochastic processes 64

7.18 Recovered paths with $k = 14$ and $n = 450$. The estimated mean reversion coefficients of recovered paths are close to the true parameter 300. The estimated mean reversion coefficient of original path is 294.9149. 64

7.19 The l^1 norms of recovered paths for different i. i is from 2 to 450. 65

List of Tables

7.1 Solutions obtained by the exhaustive search method 52

7.2 Solutions of method A, B, C and D obtained under $\|x\|_1 \leq 2.05$ 54

7.3 Recovering OU process with Jumps 63

1 Solutions of method A, B, C and D obtained under $\|x\|_1 \leq 1.5$ 71

2 Solutions of method A, B, C and D obtained under $\|x\|_1 \leq 1.8$ 72

3 Solutions of method A, B, C and D obtained under $\|x\|_1 \leq 2.1$ 72

4 Solutions of method A, B, C and D obtained under $\|x\|_1 \leq 2.4$ 73

5 Solutions of method A, B, C and D obtained under $\|x\|_1 \leq 2.7$ 73

Acknowledgements

I would like to express my gratitude to Professor Knut Sølna. Your guidance and assistance on both my research and career was invaluable. I am especially grateful to you for introducing me to the field of financial mathematics and providing me with opportunities to be physically exposed to this sector. Your patience and commitment as an advisor makes you a great mentor. I would like to express my special appreciation and thanks to my co-advisor Professor Jack Xin. Your continuous and insightful advice on my research project makes it move forward smoothly. Your financial support provides a good balance between my research and teaching duty. Without your both supervision and constant help, this dissertation would not have been possible.

A special thanks to my committee member Professor Long Chen. You are both a teacher and a friend to me. I have learnt so much from your computational PDE course. My deepest appreciation goes to all my teachers, instructors and faculty members at University of California Irvine. You have either provided me with the knowledge and skills or other necessary supports to make this dissertation happen.

I would like to thank Penghang Yin and Shuai Zhang for your helpful discussions on my research project, and all my classmates and friends in the University. You have made my Ph.D. study productive and life colorful.

I am indebted to University of California Irvine for its generous offer of scholarship in the past five years. My dissertation would not exist without its full support.

Last but not least, my gratitude goes to my parents whose selfless support and understanding are always an encouragement to my efforts to make progress in this dissertation writing and my overall life.

The work was partially supported by NSF grant DMS-1222507 while the author was co-advised by Professor Jack Xin, who thanks Dr. Wuan Luo for introducing the topic of mean reverting portfolios.

Curriculum Vitae

Xiaolong Long

B.S. in Applied Mathematics, Beijing Normal University, 2009

M.S. in Mathematics, University of California, Irvine, 2011

M.S. in Statistics, University of California, Irvine, 2013

Ph.D. in Mathematics, University of California, Irvine, 2014

ABSTRACT OF THE DISSERTATION

Constructing Sparse and Fast Mean Reverting Portfolios

By

Xiaolong Long

Doctor of Philosophy in Mathematics

University of California, Irvine, 2014

Professor Knut Sølna, Professor Jack Xin, Co-chairs

We study the problem of constructing sparse and fast mean reverting portfolios based on a set of financial data arising in convergence trading. The problem is formulated as a generalized eigenvalue problem with a cardinality constraint [4]. We develope a new proxy of mean reversion coefficient, the direct OU estimator, which can be used for both stationary and non-stationary data. In addition, we introduce two different methods to enforce the sparsity of the solutions instead of predetermining the cardinality. One method uses the ratio of l_1 and l_2 norms and the other one uses l_1 norm and prior knowledge. We analyze various formulations of the resulting non-convex optimization problems and develop efficient algorithms to solve them on portfolio sizes as large as hundreds. By adopting a simple convergence trading strategy, we test the performance of our sparse mean reverting portfolios on both generated and historical real market data. In particular, the l_1 norm regularization method gives robust results on large out-of-sample data set. We formulated a new type of problems for recovering fastest mean reverting process. It is a generalization of recovering sparse element in a subspace. From the numerical tests, we successfully recovered the hidden fastest OU process.

Introduction

Convergence trade is a trade designed to benefit from the assumption that the difference of the prices of two assets will fluctuate around a certain level. The deviations from this level are temporary and thus investors can build an appropriate trading strategy when they observe the deviations and expect to profit by the amount of convergence. Mean reversion has received a significant amount of attention as a classic indicator of predictability in financial markets and is sometimes apparent, for example, in equity excess returns over long horizons. Classical methods include co-integration [6] and canonical correlation analysis [11].

Cointegration is a statistical property of time series variables. It tests whether we could build a stationary time series as a linear combination of a set of non-stationary time series. In convergence trade, investors want the values of the portfolio to be stationary since it is easy to determine trading signals for the trading strategy. Canonical correlation analysis aims at maximizing the correlation of the linear combinations on two sets of random vectors. This can help investors find portfolios that share a common stochastic drift.

The risk of a convergence trade is that the expected convergence does not happen, or that it takes too long such that it possibly diverges before converging. Therefore, it is important to quantify how fast the portfolio will converge and what an optimal portfolio could be constructed based on this criterion. In addition, a sparse portfolio may be preferred in convergence trading since sparsity typically means less transaction costs. However, there will be a trade-off between the sparsity and the convergence rate.

An optimization framework in constructing sparse mean reverting portfolios is proposed in [4]. We briefly review the formulation here:

> ”Suppose that S_{ti} is the value at time t of an asset S_i with i=1,2...n and t=1,2,...n, we want to form a portfolio P_t of these assets with coefficients x_i, and assume it follows an Ornstein-Uhlenbeck process given by:
>
> $$dP_t = \lambda(\mu - P_t)dt + \sigma dW_t \qquad with \qquad P_t = \sum_{i=1}^{n} x_i S_{ti}$$
>
> where $\lambda > 0$, $\sigma > 0$ and μ are parameters and W_t is a standard Brownian motion. The objective here is to maximize the mean reversion coefficients λ of P_t by adjusting the portfolio weights x_i under the constraints that $\sum_{i=1}^{n} x_i^2 = 1$ and that the cardinality of x, i.e. the number of non-zero coefficients in x, remains below a given $k > 0$.”

The author of [4] found a proxy of the mean reversion coefficient, discussed methods for parameter estimation and formulated an optimization problem with sparsity constraint. Two algorithms have been proposed to solve this problem: greedy search and semidefinite relaxation. However, the greedy search often fails to produce the optimal solution and the semidefinite relaxation method is costly and unwieldy for large size portfolios with hundreds of assets. The results were mainly tested to small size data sets (8 or 14 assets). It shows that the mean reversion coefficient increases as the cardinality increases.

Some recent published articles are also focused on this problem. Cuturi et al [3] replaces the cardinality constraint by the variance constraint in order to improve the profits during arbitrage opportunities and [8] discussed more details on parameter estimation and trading strategies.

In our work, we developed two optimization methods in using norms to enforce sparsity of the portfolio.

The first method uses the ratio of l_1 and l_2 norms as a penalty function assuming limited prior information of the assets. Such a penalty term arises in non-negative matrix factorization (NMF), blind deconvolution, and sparse representation in coherent dictionaries, [12], [13], [14],[7], [22]. For example, the non-negative least squares (NNLS) problem under such penalty takes the following form [7]:

$$\min_{x\geq 0} \|\mathbf{X}x - \mathbf{Y}\|_2^2 + \gamma P(x)$$

where $P(x) = \frac{\|x\|_1}{\|x\|_2}$, $\mathbf{X}$ is an $m \times n$ matrix and $\mathbf{Y}$ is an $m \times 1$ vector. For our problem, we formulate the following minimization:

$$(1) \qquad \min_{x\neq 0} \frac{x^T A x}{x^T B x} + \gamma \frac{\|x\|_1}{\|x\|_2}$$

where A and B are positive definite matrices.

Due to the non-convexity of (1), finding a global minimum is challenging. To improve this aspect of global optimization, we shall incorporate a recent variant of simulated annealing, the so called intermittent diffusion method with discontinuous diffusion coefficient [2]. The combined local minimization of (1) and random search for global minimum is however expensive for large size portfolio computation.

The second method uses l_1 norm and our partial knowledge on the collection of assets. We reformulate the problem as a quadratic program:

$$f(r) = \max_{x(i)=1, \|x\|_1 \leq m} x^T B x - r x^T A x$$

where A and B are both positive definite matrices.

Non-convexity still exists and a special algorithm designed to optimize a difference of convex functions can be applied to overcome this difficulty in combination with convex algorithms such as the least angle regression. The resulting algorithm is the most efficient and can handle portfolios with hundreds of assets in seconds. To our best knowledge, this is the most significant computational advance to date.

Difference of convex functions(DC) programming are extensively studied in [10], [15], [19], [20]. The method aims to minimize a function $g - h$ where both g and h are convex functions on the whole space. This method works on a large set of optimization problems. In most literature of DC programming, an algorithm to find a local optima is used. Numerically however, the local algorithm often yields a global minimum.

The rest of the dissertation is structured as follows. In Chapter 1, we provide a reference on the important concepts, definitions and theories that we used in our work. In Chapter 2, we discussed two proxies of mean reversion coefficient. In Chapter 3, we present the framework built by D'Aspremont and discuss known algorithms in solving the sparse mean reverting portfolios problem. In Chapter 4, we formulate two types of optimization problems based on the ratio of l_1 and l_2 norms, discuss the algorithms to solve them and analyze how the penalty term works in enforcing sparsity. In Chapter 5, we formulate an optimization problem based on the prior knowledge and l_1 norm, prove the theory of recovering the global optimizer and present various algorithms. In Chapter 6, we discussed a new type of problems which can recover the fastest mean reverting OrnsteinUhlenbeck process. In Chapter 7, we presented numerical results of all the problems/algorithms in the previous chapters.

Chapter 1

Basic Definitions and Theories

In this chapter, we will discuss the basic definitions and theories that used through this work.

1.1 Generalized Eigenvalue and Eigenvector

Let A be a square matrix. If there is a non-zero vector v and a scalar λ such that:

$$Av = \lambda v$$

then λ is called the eigenvalue of A with corresponding eigenvector v. The possible values of λ must satisfy the following equation:

$$\det(A - \lambda I) = 0$$

In our work, we define the generalized eigenvalue and eigenvector in the following way. Let A and B be a square matrix. If there is a non-zero vector v and a scalar λ such that:

$$Av = \lambda Bv$$

then λ is called the generalized eigenvalue of A and B with corresponding eigenvector v.

The possible values of λ must satisfy the following equation:

$$\det(A - \lambda B) = 0$$

1.2 Vector Autoregressive Model

Vector autoregressive model (VAR) is an extension of univariate autoregressive model (AR). It is used to capture the linear interdependencies among multiple time series. In this section, we will give a brief introduction on VAR(1) and AR(1) models, i.e. we only consider the first lag term.

1.2.1 Model Form

Definition 1.2.1 *The AR(1) model is defined as*

$$X_t = c + \rho X_{t-1} + \epsilon_t$$

where c is a constant and ϵ_t is white noise.

In this dissertation, we will assume that ϵ_t follows a Gaussian distribution with mean 0 and variance σ^2 and $c = 0$.

Definition 1.2.2 *The VAR(1) model is defined as*

$$X_t = X_{t-1}\beta + c + \epsilon_t$$

where X_t and c are $1 \times n$ vectors, β is an $n \times n$ matrix, and ϵ_t are i.i.d $N(0, \Sigma)$.

1.2.2 Stationarity

There are two types of stationarity for time series: strong stationarity and weak stationarity.

Definition 1.2.3 *Given t_1, t_2,..., t_n, if the joint distribution of X_{t_1}, X_{t_2},..., X_{t_n} is the same as the distribution of $X_{t_1+\tau}$, $X_{t_2+\tau}$,..., $X_{t_n+\tau}$ for all n and τ, the X_t is a strong stationary process.*

Since the requirements of the strong stationarity is too strict, the weak stationarity is usually used.

Definition 1.2.4 *If $E(X_t)$ and $Var(X_t)$ don't depend on t and the covariance between X_t and $X_{t+\tau}$ is only a function of τ, then X_t is a weak stationary process.*

For AR(1) model, if the noise is Gaussian distributed, then the strong stationarity is equivalent to weak stationarity.

Theorem 1.2.1 *For AR(1) model, if $|\rho| < 1$, it is stationary. For VAR(1) model, if the solutions to $det(I - \beta z) = 0$ lie outside the unit circle or the eigenvalues of β all lie inside the unit circle, then it is stationary.*

1.2.3 Estimation of ρ and β

For AR(1) and VAR(1) models, we can use the least square estimation.

AR(1):

$$\hat{\rho} = \frac{\sum_{i=0}^{T-1} X_i X_{i+1}}{\sum_{i=0}^{T-1} X_i^2}$$

VAR(1):

$$\hat{\beta} = (\sum_{i=0}^{T-1} X_i^T X_i)^{-1} (\sum_{i=0}^{T-1} X_i^T X_{i+1})$$

1.3 Order of Integration and Cointegration

Order of integration, denoted $I(d)$, shows the minimum number of differences required to obtain a covariance stationary series. We will mainly discuss $I(0)$ and $I(1)$ in this dissertation.

Definition 1.3.1 *X_t is integrated of order 0 if X_t is stationary.*

Definition 1.3.2 *X_t is integrated of order 1 if $X_t - X_{t-1}$ is integrated of order 0.*

Definition 1.3.3 *Suppose $Y_t = (y_{1t}, y_{2t}, ..., y_{nt})$ is a vector of $I(1)$ time series. Y_t is cointegrated if there exists a $n \times 1$ vector β such that $Y_t \beta$ is I(0).*

1.4 Difference of Convex Functions Programming

Difference of Convex Functions(DC) Programming are introduced by Pham Dinh Tao and extensively developed by Le Thi Hoai An and Pham Dinh Tao [10], [15], [19]. This method aims at solving the problem of minimizing a function $g - h$ where both g and h are both convex functions on the whole space.

From the theory below, we can see that this method works on a large set of optimization problems.

Theorem 1.4.1 *The following three types of problems are equivalent:*

- $\sup\{f(x) : x \in C\}$, *where* f *and* C *are convex;*
- $\inf\{g(x) - h(x) : x \in \mathbb{R}^n\}$, *where* g *and* h *are convex;*
- $\inf\{g(x) - h(x) : x \in C, f_1(x) - f_2(x) \leq 0\}$, *where* g, h, f_1, f_2 *and* C *are all convex;*

The authors studied the conditions for global and local optimality in DC programs. However, there is not an efficient general algorithm to search for the global optima. In most literatures of DC programming, an algorithm to find a local optima is used. According to their numerical results, the local algorithm often yields the global optima. This algorithm is implemented in the following way:

Algorithm 1 DC Algorithm

```
Choose $x^0$ in $\mathbb{R}^n$
repeat
  Set $y^k$ in $\partial h(x^k)$
  Set $x^{k+1}$ in $\partial g(y^k)$ that in most cases leads to solving a convex program:
      $\inf\{g(x) - x^T y^k : x \in \mathbb{R}^n\}$
until convergence
```

The $\partial g(x^0)$ is called a different of g at x^0. It is defined in this way:

Definition 1.4.1 *Define an* ϵ*-subgradient of* g *at* x_0 *to be*

$$\partial_\epsilon g(x^0) = \{y \in \mathbb{R}^n : g(x) - g(x^0) \geq (x - x^0)^T y - \epsilon\}$$

Definition 1.4.2

$$\partial g(x^0) = \bigcap_{\epsilon > 0} \partial_\epsilon g(x^0)$$

To illustrate the idea of DCA, we will first define the primal and dual problem.

The primal problem:

$$\inf\{g(x) - h(x) : x \in \mathbb{R}^n\} \tag{1.1}$$

The dual problem:

$$\inf\{h^*(y) - g^*(y) : y \in \mathbb{R}^n\} \tag{1.2}$$

where $h^*(y)$ is called the conjugate of$h(x)$ and it is defined as

$$h^*(y) = \sup\{x^T y - h(x), x \in \mathbb{R}^n\}$$

Le Thi Hoai An gives an explanation of this algorithm: the DCA constructs two sequences $\{x^k\}$ and $\{y^k\}$. They are the candidates to be optimal solutions of primal and dual programs respectively. In addition, the sequences $\{g(x^k) - h(x^k)\}$ and $\{h(y^k) - g(y^k)\}$ are decreasing. At the step k, we uses the affine minorization $h_k = h(x^k) + (x - x^k)^T y^k$ to approximate the second component h at a neighbourhood of x^k to obtain a convex program whose the solution set is nothing but $\partial g(y^k)$. Similarly, the second DC component g^* of the dual DC program is replaced by its affine minorization $g_k^*(y) = g^*(y^k) + (y - y^k)^T x^{k+1}$ at a neighbourhood of y^k to give birth to the convex program whose $\partial h(x^{k+1})$ is the solution set.

1.5 Ornstein-Uhlenbeck Process

Ornstein-Uhlenbeck process plays an important role in constructing sparse and fast mean reverting portfolios. We will use the mean reversion coefficient, λ, to test the performance of a portfolio. Therefore, it is helpful to have a short review of this stochastic process.

Consider an Ornstein Uhlenbeck process:

$$dP_t = \lambda(\mu - P_t)dt + \sigma dW_t \tag{1.3}$$

where μ, λ and σ are parameters and λ is called mean reversion coefficient.

Using Itô's lemma, we get

$$P_t = P_0 e^{-\lambda t} + \mu(1 - e^{\lambda t}) + \int_0^t \sigma e^{\lambda s} dW_S$$

From this, we could find that the mean and variance of P_t are:

$$E(P_t) = P_0 e^{-\lambda t} + \mu(1 - e^{\lambda t}) \xrightarrow{t \to \infty} \mu$$

$$Var(P_t) = \frac{\sigma^2}{2\lambda}(1 - e^{-2\lambda t}) \xrightarrow{t \to \infty} \frac{\sigma^2}{2\lambda}$$

The mean reversion coefficient λ determines rate of convergence. Figure (1.1) compares two sample paths of two different OU processes with same parameters $P_0 = 1.3$, $\mu = 1.5$ and $\sigma = 2$ but different λ.

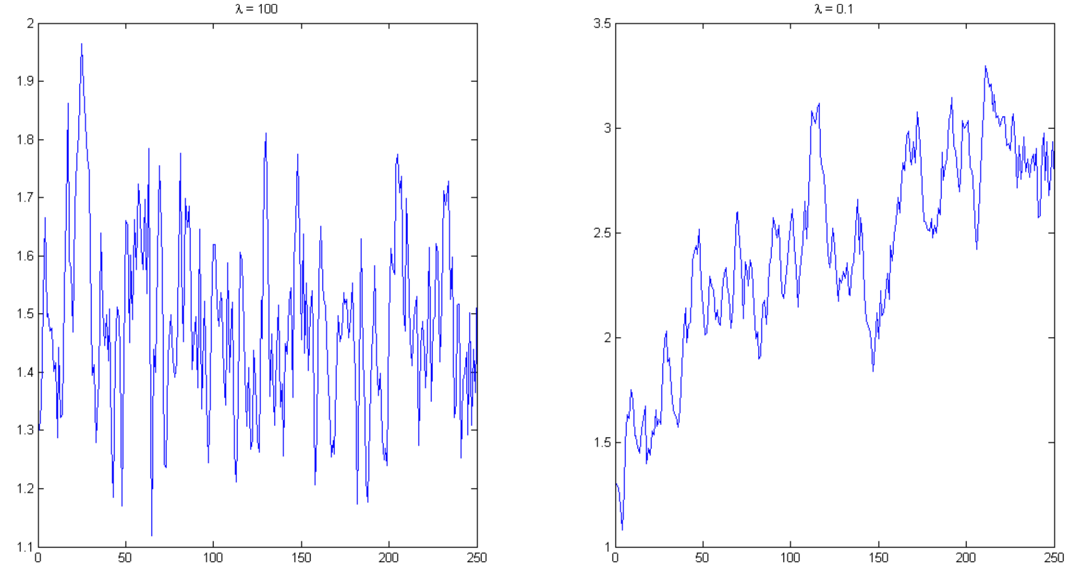

Figure 1.1: Sample paths of two Ornstein-Uhlenbeck processes

We could estimate the parameters of an OU process by linear regression. By writing

(1.3) in a discrete form, we have

$$P_t - P_{t-1} = \lambda\mu\Delta t - \lambda P_{t-1}\Delta t + \sigma dW_t \tag{1.4}$$

which is equivalent to the simple linear regression:

$$y = a + bx + \epsilon_t$$

Therefore, we could estimate all the parameters by regressing $P_t - P_{t-1}$ on P_{t-1}. Then we can recover $\hat{\lambda}$ as $-\frac{\hat{b}}{\Delta t}$, $\hat{\mu}$ as $\frac{\hat{a}}{\hat{\lambda}\Delta t}$ and $\hat{\sigma}$ as $\frac{\hat{\sigma}(\epsilon_t)}{\sqrt{\Delta t}}$. The estimation of λ is scale invariant.

1.6 LASSO and Least Angle Regression

Lasso (least absolute shrinkage and selection operator) is a regularized version of least squares, which uses the constraint that the L1-norm of the parameter vector is no greater than a given value. Robert Tibshirani proposed this method in [21] and later Efron, Bradley, et al [5] developed Least Angle Regression(LARS), a new model selection algorithm which can be used for implementing the Lasso.

The Lasso problems are defined in the following way:

Definition 1.6.1 *Let $X = \{x_1, x_2, ..., x_m\}$ be a $n \times m$ matrix and y be a $n \times 1$ vector. We could consider the columns of X represent covariates and y are the predicted responses. The Lasso estimate of $\hat{\beta}$ is*

$$\hat{\beta} = \text{argmin}\|X\hat{\beta} - y\|_2^2 \ \textit{subject to}\|\hat{\beta}\|_1 \leq t$$

This problem can be solved by quadratic programming techniques. However, a modified version of LARS is an easier and more efficient algorithm to solve the Lasso problem. In [5], the authors proved that the LARS algorithm yields all Lasso solutions with only one condition. Robert Tibshirani summarized the LARS algorithm in his website (http://statweb.stanford.edu/∼tibs/lasso/simple.html).

Algorithm 2 LARS

```
Start with all coefficients $\hat{\beta}_j$ equal to zero;
Find the covariate $x_j$ most correlated with $y$;
Increase the coefficient $\hat{\beta}_j$ in the direction of the sign of its correlation with $y$. Take residuals $r = y - X\hat{\beta}$ along the way. Stop when some other covariate $x_k$ has as much correlation with $r$ as $x_j$ has.
Increase $(\hat{\beta}_j, \hat{\beta}_k)$ in their joint least squares direction, until some other covariate $x_m$ has as much correlation with the residual $r$.
Continue until: all covariates are in the model
```

For the LARS algorithms, we only need m steps to obtain the full set of solutions. Normally, in each step, one new covariate will be added. The entire sequence of steps in the LARS algorithm with $m < n$ covariates requires $\mathrm{O}\big(m^3 + nm^2\big)$.

Chapter 2

Proxies of Mean Reverting Property

In [3], the authors discussed three different criteria to measure how fast a portfolio is mean-reverting. They are predictability, the portmanteau statistics and the crossing statistics. In [16], the authors reviewed some of the most applied hedging methods and the test statistics to judge whether a portfolio can be used for hedging. We would refer the reader [3],[1],[16] for more details. In our paper, we mainly consider two proxies. One is predictability and the other one is called the direct OU estimator. To our knowledge, no one has used the second proxy before.

2.1 Predictability

The idea of predictability of a time series is first derived in [1]. They consider a stationary autoregressive model:

$$S_t = S_{t-1}\beta + c + Z_t, \quad S_t \in \mathbb{R}^n \tag{2.1}$$

where S_{t-1} is the lagged portfolio process, $c \in \mathbb{R}^n$, $\beta \in \mathbb{R}^{n\times n}$ and Z_t is a vector of i.i.d Gaussian noise with zero mean and a covariance matrix Σ, independent of S_{t-1}. The condition of stationarity of S_t is that all the eigenvalues of β lies inside the unit circle.

In univariate case,

$$\mathbf{E}[S_t^2] = \mathbf{E}[(S_{t-1}\beta)^2] + \mathbf{E}[Z_t^2]$$

which can be rewritten as $\sigma_t^2 = \sigma_{t-1}^2 + \Sigma$. Box & Tiao (1977) measure the *predictability* of stationary series by:

$$\nu = \frac{\sigma_{t-1}^2}{\sigma_t^2}$$

d'Aspremont [4] propose to use this measure of predictability as a proxy for the mean reversion parameter λ.

In multivariate case, consider a portfolio $P_t = S_t x$ with weights $x \in \mathbb{R}^n$, then by multiplying both sides of (2.1) by x, we get

$$S_t x = S_{t-1}\beta x + cx + Z_t x$$

and we can measure its predictability as:

$$\nu_1(x) = \frac{x^T \beta^T \Gamma \beta x}{x^T \Gamma x}$$

where Γ is the covariance matrix of S_t. This proxy is first introduced in [4]. Minimizing the predictability $\nu_1(x)$ corresponds to maximizing λ.

2.2 Direct OU estimator

Based on the result in Chapter 1, maximizing the estimated mean reversion coefficient λ corresponds to minimizing the estimated slope of $\hat{b}$. Note that

$$b = \frac{Cov(P_t - P_{t-1}, P_{t-1})}{Var(P_{t-1})} = \frac{Cov(P_t, P_{t-1})}{Var(P_{t-1})} - 1$$

Let's replace P_t by $S_t x$, we have:

$$b = \frac{Cov(S_t x, S_{t-1}x)}{Var(S_{t-1}x)} - 1$$

We define the direct OU estimator as:

$$\nu_2(x) = \frac{Cov(S_t x, S_{t-1}x)}{Var(S_{t-1}x)} \tag{2.2}$$

If S_t is stationary, then we can rewrite (2.2) in the following form:

$$\nu_2(x) = \frac{x^T(Cov(S_t, S_{t-1}) + Cov(S_t, S_{t-1})^T)x}{2x^T Var(S_{t-1})x}$$

Here we assume that the covariance matrix of S_t is positive definite and we have enough observations of the assets' values to get a full rank estimation of the covariance matrix. Minimizing the predictability $\nu_2(x)$ corresponds to maximizing λ.

This proxy captures similar property of a time series as the portmanteau statistics in [3] and the direct Dickey-Fuller statistics in [16]. One thing to note is that the proxy (2.2) works even if we don't assume that the asset prices S_t are stationary. The only assumption needed is that the linear combinations of those assets $S_t x$ is stationary. This phenomenon is called cointegration [6]. This helps us relax the assumption. According to our numerical tests, the portfolios constructed under this proxy have a better performance in trading.

Though the estimation of $Cov(S_t, S_{t-1}) + Cov(S_t, S_{t-1})^T$ is not guaranteed to be positive definite, our numerical tests show that in most cases it is still true. Even if this condition fails, we could use the estimation of $Cov(S_t, S_{t-1}) + Cov(S_t, S_{t-1})^T + \rho Var(S_{t-1})$ as the matrix in the quadratic form of the numerator where ρ is a tuning parameter. By choosing an appropriate ρ, the matrix will be positive definite and it is equivalent to the optimization problem with $\rho = 0$.

Chapter 3

Sparse Optimization Problem and Related Algorithms

3.1 Sparse Optimization Problem

In the previous chapter, we discussed two proxies of the mean reversion coefficient. Note that both of them can be written as

$$\frac{x^T Ax}{x^T Bx}$$

where A and B are $n \times n$ positive definite matrices and we will make this assumption for the rest of the work. If we do not care the cardinality of x, then minimizing these proxies are the same as a generalized eigenvalue problem. In order to obtain a sparse solution, d'Aspremont in [4] proposed the following sparse optimization problem:

$$\begin{aligned} \min \quad & x^T Ax/x^T Bx \\ s.t. \quad & \|x\|_0 \leq k \\ & \|x\|_2 = 1, \end{aligned} \tag{3.1}$$

This problem has been proved to be NP-hard [17]. When the dimension of the problem is large, we cannot expect to find the optimal solution. Several methods of solving 3.1 have been proposed in [4] and [8]. We will give a brief summary here.

3.2 Algorithms

- Exhaustive search method: it tests all $\frac{n!}{k!(n-k)!}$ possible combinations of assets and find the smallest generalized eigenvalue and eigenvectors. This method gives us an optimal solutions and works very well when n is small. However, it will be extremely slow when n is large.

- Greedy search: denote I_k as the support of the solution and set $I_k = \emptyset$ initially. Each time, we pick one asset form $\{1, 2, ..., n\} \setminus I_k$ such that it has smallest objective among all the other choice. Then we add it to I_k and repeat this procedure for k times. This method gives a sub-optimal solution.

- Truncation method: first we solve the unconstrained problem and find x_{opt}. Then we find the index set J_k of the largest k components of $|x_{opt}| = (|x_1|, ..., |x_n|)^T$. Then we solve the generalized eigenvalue problem on the set J_k by taking the corresponding part of matrices A and B. It gives a sub-optimal solution and it is the fastest among all the listed methods since it only requires solving the generalized eigenvalue problem twice.

- Semidefinite relaxation method: this method reformulates the problem (3.1) as a semidefinite program. It provides sub-optimal solutions. The computational complexity is lower than the exhaustive search method but is higher than that of the greedy search. For more details, we refer the readers to [4].

Chapter 4

Optimization Problem Based on the Ratio of l_1 and l_2 Norms

4.1 Motivation

One popular method in handling a cardinality constraint is to use a norm penalization. Standard choices include l_1 and l_p $(0 < p < 1)$. In our case, we would prefer using $\frac{\|x\|_1}{\|x\|_2}$ since our problem (1) is scale invariant. The scale constraint is required for $\|x\|_1$ and $\|x\|_p$ penalties otherwise they cannot enforce sparsity. The reason is that we could simply decrease the penalty by decreasing the scale of all the elements of x. For analysis of sparsity promoting properties of ratio of l_1 and l_2 norms, we refer to [22].

4.2 Constraint Problem

4.2.1 Formulation

We could reformulate the problem (3.1) in the following way:

$$\begin{array}{rl} \min_x & x^T Ax/x^T Bx \\ s.t. & \frac{\|x\|_1}{\|x\|_2} \leq m \\ & x \neq 0, \end{array} \tag{4.1}$$

Using the similar method as [4] , we found that this problem can be also expressed as a semidefinite programming problem.

By setting $X = xx^T$, then the problem (4.1) is equivalent to

$$\begin{array}{ll} \min_X & \mathrm{trace}(AX)/\mathrm{trace}(BX) \\ s.t. & \frac{1^T|X|1}{\mathrm{trace}(X)} \leq m^2 \\ & rank(X) = 1 \\ & X \succeq 0 \end{array}$$

where $|X|$ means we take the absolute value for each entry of X.

Then by changing of variables:

$$Y = \frac{X}{\mathrm{trace}(BX)}, \quad z = \frac{1}{\mathrm{trace}(BX)}$$

and dropping the rank constraint, the previous problem can be written as a semidefinite programming problem:

$$\begin{array}{ll} \min_Y & \mathrm{trace}(AY) \\ s.t. & 1^T|Y|1 \leq m^2 z \\ & \mathrm{trace}(Y) - z = 0 \\ & \mathrm{trace}(BY) = 1 \\ & Y \succeq 0 \end{array} \tag{4.2}$$

If we set $\mathrm{Card}(x) = k = m^2$, this is exactly the semidefinite relaxation in [4].

4.2.2 Algorithm

Since we have relaxed the problem to a semidefinite program, we could use many well-established package like SeDuMi and Yalmip to solve it. The only difficulty is handling the constraint:

$$1^T|Y|1 \leq m^2 z$$

It corresponds to 2^{n^2} constraints. When the dimension of the problem is small, we could write out a set of linear constraints that cover all the possible signs of x. This method is impossible to implement when n is large. A classical method to tackle this issue is starting with a smaller set of linear constraint(s) and increasing the set if the current solution fails the original constraint. Tibshirani proposed this method in solving the LASSO type problem [21].

For illustration, we consider a simple example: $|x| \leq t$ where $x = (x_1, x_2)^T$. Therefore, $|x| \leq t$ is equivalent to the following 4 linear inequalities:

$$x_1 + x_2 \leq t \tag{4.3a}$$

$$x_1 - x_2 \leq t \tag{4.3b}$$

$$-x_1 + x_2 \leq t \tag{4.3c}$$

$$-x_1 - x_2 \leq t \tag{4.3d}$$

If we want to minimize some function $f(x)$ subject to $|x| \leq t$, we start with the problem of minimizing $f(x)$ subject to 4.3a. We can get a minimizer x^*. If $|x^*| \leq t$, then it is also the minimizer under the constraint $|x| \leq t$. If not, then we add a new constraint: $sign(x_1^*)x_1 + sign(x_2^*)x_2 \leq t$. Now we will solve the optimization problem with two linear constraints. We repeat this process until the solution satisfies $|x| \leq t$.

For our problem, we proposed the following algorithm:

Algorithm 3 Solve (4.2)

```
Input the parameters $A$, $B$ and $m$;
Set an initial $X_0$ which corresponds to the solution without sparsity constraint;
Constraint set = {Null}
Calculate $Y = \frac{X_0}{\text{trace}(BX_0)}$ and $z = \frac{1}{\text{trace}(BX_0)}$
while $1^T|Y|1 > m^2 z$ do
    Add sign($Y$) to the constraint set;
    Solve the problem (4.2) by reducing the first $2^{n^2}$ constraints to the current constraint set;
    Update $X_0$, $Y$ and $z$
end while
```

The last issue is how to recover x from X. We can express X as:

$$X = \sum_{i=1}^{n} \lambda_i q_i q_i^T$$

where λ_i are eigenvalues with corresponding eigenvector q_i and λ_i is a non-increasing sequence. Therefore, we could pick the first eigenvector q_1 and select the largest $k = m^2$ entries in absolute values.

4.3 Penalized Optimization Problem

4.3.1 Formulation

We could reformulate the problem (3.1) in the following way:

$$\min_{x \neq 0} \frac{x^T A x}{x^T B x} + \gamma \frac{\|x\|_1}{\|x\|_2} \tag{4.4}$$

where γ is a tuning parameter.

Comparing with the problem (3.1), the advantage of the problem (4.4) is that it does not specify the cardinality beforehand. This could be closer to the reality. In addition, we could consider the l_1 norm in the numerator as a way to quantify the uniform transaction costs.

4.3.2 Algorithm

The major difficulty comes from the non-convexity of the problem. Most optimization algorithms will only provide a local minimizer. In our problem, the performance of local minimizers may vary a lot. Therefore, it is important to develop an algorithm to find the global minimizer. The intermittent diffusion(ID) algorithm proposed in [2] can be helpful. The main idea of this algorithm is "to add intermittent, instead of continuously diminishing, random perturbations to the gradient flow generated by the objective, so that the trajectories can quickly escape from the trap of one minimizer and then approach others."

In addition, we also need to approximate the derivative of the non-smooth function

$$x^T Ax/x^T Bx + \gamma \frac{\|x\|_1}{\|x\|_2}$$

The non-smoothness is due to the term $\|x\|_1$. A common trick in handling this problem is using

$$|x_i| \approx \sqrt{x_i^2 + \epsilon}$$

where ϵ is a small tuning parameter.

Last but not the least, to satisfy the conditions of ID algorithm so that a global minimizer exists in a bounded set, we also add a penalty function $p(x, \theta, \xi, \zeta)$ to $f(x)$:

$$p(x, \theta, \xi, \zeta) = \sum_i u(x_i, \theta, \xi, \zeta)$$

where

$$u(x_i, \theta, \xi, \zeta) := \begin{cases} \xi(x_i - \theta)^\zeta, & x_i > \theta \\ 0, & |x_i| \leq \theta \\ \xi(x_i - \theta)^\zeta, & x_i < -\theta \end{cases}$$

In this way, we can make the objective to be infinity when $|x|$ is approach to infinity.

Finally, our objective becomes:

$$F(x) = \sum_i u(x_i, \theta, \xi, \zeta) + x^T Ax/x^T Bx + \gamma \frac{\|x\|_1}{\|x\|_2} \tag{4.5}$$

Algorithm 4 Solve (4.4)

```
Input $A$, $B$ and $\gamma$;
Set $\alpha$ as the scale for diffusion strength, $\kappa$ the scale for diffusion time and the total number of realizations $N$.
Set the initial state $x_0$ as the minimizer of $\frac{x^TAx}{x^TBx}$ with the constraint that $\|x_0\|_2 = 1$, i.e. the generalized eigenvector associated with the smallest eigenvalue
Find a local minimizer $\hat{x}_0$ of problem (4.4) given $x_0$ and set $X_{opt} = \hat{x}_0$.
for $i = 1$ to $N$ do
    Generate two positive random numbers $d$, $s$ within $[0,1]$ by uniform distribution and let $\sigma := \alpha d$ and $T := \kappa s$.
    Solve the stochastic equation for $t \in [0,T]$
        $dx(t,\omega) = -\bigtriangledown^s F(x(t,\omega))dt + \sigma dW(t), \quad x(0,\omega) = X_{opt}$
    where $\bigtriangledown^s F$ is the gradient of the objective 4.4 and record the final state $x_T := x(T,\omega)$.
    Find a local minimizer $\hat{x}_i$ of problem (4.4) by line search algorithm with starting point $x_T$.
    $X_{opt} = \hat{x}_i$ if $f(\hat{x}_i) < f(X_{opt})$.
end for
```

4.4 Analysis of the ratio of l_1 and l_2 penalty

In this section, we want to prove some useful properties of the solutions of problem (4.4).

We will define $f(x,\gamma)$, $L(x)$ and $P(x)$ as:

$$f(x,\gamma) = \frac{x^TAx}{x^TBx} + \gamma\frac{\|x\|_1}{\|x\|_2}$$

$$L(x) = \frac{x^TAx}{x^TBx}$$

$$P(x) = \frac{\|x\|_1}{\|x\|_2}$$

We denote the set of minimizers of the problem (4.4) as $x(\gamma)$ and we will set their l_2 norm as 1. When γ is fixed, the existence of the minima of $f(x,\gamma)$ is due to its continuity on the unit sphere and the compactness of the unit sphere. Since x^TAx/x^TBx is scale invariant, any vector on the same direction yields the same value. If there are different directions that yield the same $f(x,\gamma)$, then we always prefer those with the smaller ratio of l_1 norm and l_2 norm. Therefore, the set of the optimizers, $x(\gamma)$, has a unique value of the function $P(\cdot)$

on this set (and then the unique value of $L(\cdot)$). Mathematically, it can be defined in the following way:

$$X(\gamma) = \{x \in \mathbb{R}^n : \|x\|_2 = 1, f(x,\gamma) = \min_{z \neq 0, \|z\|_2 = 1} \{\frac{z^T A z}{z^T B z} + \gamma \frac{\|z\|_1}{\|z\|_2}\}\}$$

$$x(\gamma) = \{x \in X(\gamma) : P(x) = \min\{P(y) : y \in X(\gamma)\}\}$$

In the following proofs, we will use these notations:

$$B(x,\delta) \equiv \{x' : \|x' - x\|_2 < \delta\}$$

$$S_k \equiv \{x \in \mathbb{R}^n : \|x\|_2 = 1, \|x\|_0 \leq k\}$$

$$S^n \equiv \{x \in \mathbb{R}^n : \|x\|_2 = 1\} = S_n$$

$$d(U,V) = \min\{\|x - y\|_2 : x \in U,\ y \in V,\ U \textit{ and } V \textit{ are compact sets in } \mathbb{R}^n\}$$

Lemma 4.4.1 *Given $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$ are both $n \times n$ positive-definite matrices, then for any $\epsilon > 0$, there exist a number $\gamma(\epsilon)$, such that for any $\gamma \geq \gamma(\epsilon)$, $\|x - e\|_2 < \epsilon$, where $x(\gamma)$ is the set of optimal solutions of problem (4.4) given γ, $x \in x(\gamma)$, e is a 1-sparse vector and $\|e\|_2 = 1$.*

Proof. First, note that the problem (4.4) is scale invariant, therefore, we could constrain the problem on the sphere $\|x\|_2 = 1$. All the vectors in our proof are with l_2 norm 1.

Note that the 1-sparse minimizers of $L(x)$ on the sphere $\|x\| = 1$ are the vectors $\pm e_i$ with all 0s except for a 1 in the ith coordinate. The i is determined by

$$i = \arg\min_i \{\frac{a_{ii}}{b_{ii}}\}$$

Without loss of generality, we will assume that $i = 1$ and this corresponds to a unique minimizer, i.e.

$$\frac{a_{11}}{b_{11}} < \frac{a_{ii}}{b_{ii}}, \quad \textit{for all } i \neq 1$$

Note that both $f(x,\gamma)$ and $L(x)$ are even functions of x, we could further restrict our region on $S_+ \equiv \{x = (x_1, ..., x_n)^T : x_1 \geq 0 \ \textit{ and } \ \|x\|_2 = 1\}$.

Since $L(x)$ is continuous on S_+, then $\exists \delta_i$, for any point x in the region $\{x : \|x \pm e_i\| < \delta_i\} \cap S_+$, $L(x) \geq L(e_1) = \frac{a_{11}}{b_{11}}$, for $i = 2, 3, ..., n$.

Denote $D_i = \{x : \|x \pm e_i\| < \delta_i\}$ for $i = 2, 3, ..., n$, then for all the points $x \in S_+ \cap (\cup_{i=2}^n D_i)$, $L(x) \geq L(e_1)$ and $\|x\|_1/\|x\|_2 \geq 1$. Therefore, they cannot beat e_1 for any γ.

For any $\epsilon > 0$, let $D_1 = \{x : \|x - e_1\| < \epsilon\}$, then $S_+(\epsilon) \equiv S_+ \setminus (\cup_{i=1}^n D_i)$ is a closed and bounded set. Therefore, $\gamma(\epsilon)$ must satisfy the following inequality:

$$f(e_1, \gamma(\epsilon)) \leq f(x, \gamma(\epsilon)), \quad \forall x \in S_+(\epsilon).$$

This is equivalent to

$$\frac{a_{11}}{b_{11}} + \gamma(\epsilon) \leq L(x) + \gamma(\epsilon)\frac{\|x\|_1}{\|x\|_2}, \quad \textit{for any } x \in S_+(\epsilon).$$

And it implies:

$$\gamma(\epsilon) \geq (\frac{a_{11}}{b_{11}} - L(x))/(\frac{\|x\|_1}{\|x\|_2} - 1), \quad \textit{for any } x \in S_+(\epsilon).$$

The last line holds, since for any $x \in S_+(\epsilon)$, $\frac{\|x\|_1}{\|x\|_2}$ is guaranteed to be greater than 1. Note that the function $(\frac{a_{11}}{b_{11}} - L(x))/(\frac{\|x\|_1}{\|x\|_2} - 1)$ is well-defined and continuous on $S_+(\epsilon)$. Therefore, we can set $\gamma(\epsilon)$ as:

$$\gamma(\epsilon) = \max_{x \in S_+(\epsilon)} (\frac{a_{11}}{b_{11}} - L(x))/(\frac{\|x\|_1}{\|x\|_2} - 1)$$

□

Lemma 4.4.1 indicates that an almost 1-sparse solution can always be recovered by increasing the value of γ.

Now consider the optimization problems (3.1). Denote $\{\lambda_k\}_{k=1}^n$ as the minimums of the problem (3.1) given parameter k. Obviously, the following inequalities hold:

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_{n-1} \geq \lambda_n$$

The equal sign could hold by considering the following two matrices:

$$A = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 2 \end{bmatrix}, \qquad B = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$$

$(1, 0)^T$ is a generalized eigenvector of A and B and it corresponds to the smallest generalized eigenvalue. Therefore, $\lambda_1 = \lambda_2$. Using the similar structure, we could construct two matrices of any dimension n such that the $\lambda_1 = \lambda_n$. In this trivial case, the ratio of l_1 and l_2 is useless in searching for sparse solutions.

We want to consider the non-trivial cases, i.e. we will assume that the generalized eigenvalues of the matrices/sub-matrices of A and B satisfies the following condition:

$$\text{(H1)} \qquad \lambda_1 > \lambda_2 > ... > \lambda_{n-1} > \lambda_n$$

Lemma 4.4.2 *Given $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$ are both $n \times n$ positive-definite matrices, suppose $\{\lambda_k\}_{k=1}^n$ are the minimums of the problem (3.1) under the parameter k and they satisfy (H1). Then for $k \geq 2$, $\exists\ \delta > 0$ and $\gamma_k > 0$, such that for any $0 \leq \gamma \leq \gamma_k$, a minimizer $x \in x(\gamma)$ of (4.4) satisfies the following inequality:*

$$\|x - w\|_2 \geq \delta, \qquad \forall w \in S_{k-1}$$

Proof. By the same reason, we could constrain the problem on the sphere $\|x\|_2 = 1$. All the vectors in our proof are on the unit ball in $\mathbb{R}^n$.

Note that S_{k-1} is the union of finite many compact sets in $\mathbb{R}^n$, so it is also compact.

For $k \geq 2$, denote the x_k as the generalized eigenvector associated with λ_k. $\|x_k\|_1$ must be strictly greater than 1 based on (H1).

Note that the function $L(x)$ is continuous on the unit ball in $\mathbb{R}^n$. Due to (H1) and the continuity property, for any point $y \in S_{k-1}$, there exists $\delta(y)$ such that for any $y_0 \in B(y, \delta(y))$, $L(x_k) + c \leq L(y_0)$, where c is a constant between 0 and $\lambda_{k-1} - \lambda_k$.

Since $U \equiv \cup_{y \in S_{k-1}} \{x : \|x - y\| < \delta(y)\}$ is an open set, then $S^n \setminus U$ is a closed set. Note that $(S^n \setminus U) \cap S_{k-1} = \emptyset$ and both are compact, then

$$\delta = d(S^n \setminus U, S_{k-1}) > 0. \tag{4.6}$$

Since for any γ, we have

$$L(x) + \gamma \leq f(x)$$

Therefore, γ_k only need to satisfy the following inequality:

$$f(x_k, \gamma_k) \leq L(x) + \gamma_k, \qquad \textit{for any } x \in U$$

It's equivalent to:

$$\lambda_k + \gamma_k \|x_k\|_1 \leq L(x) + \gamma_k, \qquad \textit{for any } x \in U$$

which implies:

$$\gamma_k \leq (L(x) - \lambda_k)/(\|x_k\|_1 - 1), \qquad \textit{for any } x \in U$$

Therefore, we could choose γ_k as $c/(\|x_k\|_1 - 1)$. For any $\gamma \leq \gamma_k$, x_k beats all the points in U and therefore the minimizer $x(\gamma)$ must be in $S^n \setminus U$. By (4.6),

$$\|x(\gamma) - w\|_2 \geq \delta \qquad \textit{for any } w \in S_{k-1}$$

□

Lemma (4.4.2) tells us that there exists γ_k such that when $\gamma \leq \gamma_k$ the minimizer of the problem (4.4) is **at least** k-sparse for $k \geq 2$. Then, immediately we could get the next corollary.

Corollary 4.4.1 *A necessary condition to have an almost k-sparse minimizer of the problem (4.4) given γ is that there exist γ_k and γ_{k+1} such that $\gamma_k > \gamma_{k+1}$.*

Lemma 4.4.3 *Suppose $x(\gamma)$ is the set of minimizers of the problem (4.4), then $P(x(\gamma))$ is non-increasing in γ.*

Proof. For any arbitrary γ_1 and γ_2, assume $\gamma_1 < \gamma_2$. We want to show that $P(x(\gamma_1)) \geq P(x(\gamma_2))$. Since $x(\gamma_1)$ and $x(\gamma_2)$ are two minimizers, so we have

$$L(x(\gamma_1)) + \gamma_1 P(x(\gamma_1)) \leq L(x(\gamma_2)) + \gamma_1 P(x(\gamma_2)) \tag{4.7a}$$

$$L(x(\gamma_2)) + \gamma_2 P(x(\gamma_2)) \leq L(x(\gamma_1)) + \gamma_2 P(x(\gamma_1)) \tag{4.7b}$$

(4.7a) and (4.7b) implies

$$L(x(\gamma_1)) - L(x(\gamma_2)) \leq \gamma_1 (P(x(\gamma_2)) - P(x(\gamma_1))) \tag{4.8a}$$

$$L(x(\gamma_1)) - L(x(\gamma_2)) \geq \gamma_2 (P(x(\gamma_2)) - P(x(\gamma_1))) \tag{4.8b}$$

Then we have:

$$\gamma_2 (P(x(\gamma_2)) - P(x(\gamma_1))) \leq L(x(\gamma_1)) - L(x(\gamma_2)) \leq \gamma_1 (P(x(\gamma_2)) - P(x(\gamma_1))) \tag{4.9}$$

This implies

$$(\gamma_2 - \gamma_1)(P(x(\gamma_2)) - P(x(\gamma_1))) \leq 0$$

By the assumption that $\gamma_1 < \gamma_2$, therefore $P(x(\gamma_1)) \geq P(x(\gamma_2))$. □

We can consider $f(x, \gamma)$ as a continuous function of both x and γ defined on U:

$$f(x, \gamma) : U \equiv S^n \times [0, L] \to R$$

where S^n is the unit ball in $\mathbb{R}^n$. Therefore, U is a closed and bounded set. In the following proofs, we assume that $x(\gamma)$ only has two vectors x and $-x$. They are in the opposite directions. We will call this condition (H2). Under H2, it is harmless to use $x(\gamma)$ to denote both the set or the vectors. We will abuse this notation for the proofs below.

Lemma 4.4.4 *Suppose $x(\gamma)$ is the set of minimizers of the problem (4.4) under parameter γ. Assume that $X(\gamma)$ is 1-dimension. Let U be a closed set in $\mathbb{R}^n$, $U \cap S^n \neq \emptyset$ and $x(\gamma) \cap U = \emptyset$. Then there exist $\delta_1 > 0$ and $\delta_2 > 0$ such that for any $x \in U \cap S^n$ and any $\gamma_0 \in (\gamma - \delta_1, \gamma - \delta_2)$,*

$$f(x, \gamma_0) > f(x(\gamma), \gamma_0)$$

Proof. Since $X(\gamma)$ is 1-D, then we have:

$$f(x(\gamma),\gamma) < f(x,\gamma) \quad \forall x \in U \cap S^n$$

Note that $U \cap S_n$ is a compact set in $\mathbb{R}^n$. Therefore, there exist f_L and P_U such that:

$$f_L = \min_{x \in U \cap S_n} \{f(x,\gamma)\} > f(x(\gamma),\gamma)$$

$$P_U = \max_{x \in U \cap S_n} P(x)$$

We will only be interested in the case that $P_U > P(x(\gamma))$, since δ_1 could be any number if this is not true.

For δ_1, we want to find a range of $\Delta\gamma > 0$ such that:

$$f(x(\gamma),\gamma) - \Delta\gamma P(x(\gamma)) < f(x,\gamma) - \Delta\gamma P(x) \qquad \forall x \in U \cap S^n$$

Then we only need $\Delta\gamma > 0$ satisfies:

$$f(x(\gamma),\gamma) - \Delta\gamma P(x(\gamma)) < f_L - \Delta\gamma P_U$$

It is equivalent to:

$$\Delta\gamma(P_U - P(x(\gamma)) < f_L - f(x(\gamma),\gamma)$$

which implies:

$$\Delta\gamma < (f_L - f(x(\gamma),\gamma))/(P_U - P(x(\gamma))$$

Therefore, we could set $\delta_1 = (f_L - f(x(\gamma),\gamma))/(P_U - P(x(\gamma))$.

Similarly, for δ_2, we want to find a range of $\Delta\gamma > 0$ such that:

$$f(x(\gamma),\gamma) + \Delta\gamma P(x(\gamma)) < f(x,\gamma) + \Delta\gamma P(x) \quad \forall x \in U \cap S^n$$

Then we only need $\Delta\gamma > 0$ satisfies:

$$f(x(\gamma),\gamma) + \Delta\gamma P(x(\gamma)) < f_L + \Delta\gamma$$

Thus,

$$\Delta\gamma(P_U - 1) < f_L - f(x(\gamma),\gamma)$$

and we have

$$\Delta\gamma < (f_L - f(x(\gamma),\gamma))/(P_U - 1)$$

Therefore, we could set $\delta_2 = (f_L - f(x(\gamma),\gamma))/(P_U - 1)$. □

Corollary 4.4.2 *Suppose $x(\gamma)$ is the set of minimizers of the problem (4.4) under parameter γ and assume that $X(\gamma)$ satisfies H2. If $\|x(\gamma)\|_0 = k > 1$, then there exist δ_1 and δ_2 such that for any $x \in S_{k-1}$ and any $\gamma_0 \in (\gamma - \delta_1, \gamma - \delta_2)$,*

$$f(x,\gamma_0) > f(x(\gamma),\gamma_0)$$

Proof. S_{k-1} is a closed set and $x(\gamma) \cap S_{k-1} = \emptyset$. □

Corollary 4.4.3 *Suppose $x(\gamma)$ is the unique minimizer of the problem (4.4) under parameter γ and assume that $X(\gamma)$ satisfies H2. If $\|x(\gamma)\|_0 = k > 1$, then for any ϵ, there exist $\delta_1 > 0$ and $\delta_2 > 0$ such that for any $\gamma_0 \in (\gamma - \delta_1, \gamma - \delta_2)$,*

$$f(x,\gamma_0) > f(x(\gamma),\gamma_0) \quad \forall x \in S_n \setminus S_k(\epsilon)$$

where

$$S_k(\epsilon) \equiv \cup_{x \in S_k} B(x,\epsilon)$$

Proof. $S_k(\epsilon)$ is an open set, so $S_n \setminus S_k(\epsilon)$ is a compact set. $x(\gamma) \in S_k(\epsilon)$ and therefore, $x(\gamma) \notin S_n \setminus S_k(\epsilon)$. □

Corollary 4.4.2 and 4.4.3 indicate the local properties of the optimal solution given the parameter γ. Corollary 4.4.2 shows that in a neighbourhood of γ, the minimizer of problem (4.4) has at least k non-zero entries. Corollary 4.4.3 shows that in a neighbourhood of γ, the minimizer will not be far away from S_k.

Theorem 4.4.1 *Denote $U \subset S^n$ as a subset of feasible points of problem (4.4). Assume that $X(\gamma)$ satisfies H2 for all γ. Suppose we have the follow property:*

$$L(x) \leq L(y) \Rightarrow \|x\|_0 \geq \|y\|_0$$

Then $\|x(\gamma)\|_0$ is a non-increasing function of γ, where $x(\gamma)$ are the minimizers of problem (4.4) restricted on the subset U.

Proof. For any arbitrary γ_1 and γ_2, suppose that we have $\gamma_1 > \gamma_2 \geq 0$.

Then by the inequality (4.9) in lemma 4, we must have

$$L(x(\gamma_1)) - L(x(\gamma_2)) \leq \gamma_1(P(x(\gamma_2)) - P(x(\gamma_1))) \leq 0$$

Therefore, $L(x(\gamma))$ is non-increasing in γ.

Note that $P(x(\gamma_2)) = P(x(\gamma_1)) \iff L(x(\gamma_2)) = L(x(\gamma_1))$, otherwise one optimizer will beat the other one for all $\gamma \geq 0$.

By our assumptions, $\|x(\gamma)\|_0$ is a non-increasing function of γ if we restrict the problem (4.4) on U. □

Chapter 5

Optimization Problem Based on the l_1 Norm and Prior Knowledge

5.1 Motivation

In the previous chapter, we applied a stochastic algorithm in searching for a global optimizer of the problem (4.4). The major drawback of the algorithms is that we can not guarantee a global optimizer. Based on the theory of the intermittent diffusion(ID) algorithm, we could increase the probability of finding the global optimizer by increasing the number of realizations N. However, choosing N is problem dependent and it will be a difficult task when the dimension of the problem is large. We have to balance the efficiency and the performance of the solution.

Therefore, an efficient global optimization algorithm is desired. In addition, we may prefer a simpler norm to enforce sparsity, because it will lead to simpler algorithms. Last but not the least, we would like to keep the number of tuning parameters as few as possible.

5.2 Formulation

Note that the problem (3.1) is equivalent to:

$$\begin{aligned} \max \quad & x^T Bx/x^T Ax \\ s.t. \quad & \|x\|_0 \leq k \\ & \|x\|_2 = 1 \end{aligned}$$

Now, we use the l_1 norm to enforce the sparsity and consider the following problem:

$$f(r) = \underset{x(i)=1, \|x\|_1 \leq M}{\text{maximize}} \; x^T Bx - rx^T Ax \tag{5.1}$$

where i is a pre-determined fixed number. We could choose i based on our prior knowledge or our investment need. For example, we could choose i by selecting the asset which has the largest entry in absolute value by solving the unconstraint problem. The constraint $x(i) = 1$ helps the l_1 norm to enforce the sparsity and also simplify the problem to a quadratic program.

5.3 Theory of Recovering the Global Optimizer

We would like to show that

Theorem 5.3.1 *Let*

$$f(r) = \underset{x(i)=1, \|x\|_1 \leq M}{\mathit{maximize}} \; x^T Bx - rx^T Ax$$

where A and B are both positive definite matrices, then

a For any given $R > 0$, (5.1) is continuous on $0 \leq r \leq R$;

b $f(r)$ is an non-increasing function of r;

c There exists r_* such that $f(r_*) = 0$;

d Suppose the optimizer at r_* is x_*, then x_* is the optimizer of the following problem:

$$\begin{aligned} &\underset{x\neq 0}{\mathit{maximize}} && x^TBx/x^TAx \\ (5.2) \qquad &\mathit{subject\ to} && x(i) = 1 \\ & && \|x\|_1 \leq m \end{aligned}$$

Proof. a. For any given $R > 0$, since $f(x, r) = x^TBx - rx^TAx$ is continuous on the closed and bounded set $\{(x, r), x(i) = 1, \|x\|_1 \leq m, 0 \leq r \leq R\}$, all the optimizers are obtainable. Due to the uniform continuity, $f(r)$ is continuous.

b. Suppose $r_1 > r_2$, then for any feasible x, we must have

$$\begin{aligned} &(x^TBx - r_1x^TAx) - (x^TBx - r_2x^TAx) \\ &= (r_2 - r_1)x^TAx < 0 \end{aligned}$$

Since A is positive definite and $x \neq 0$. Therefore, it is non-increasing in r.

c. Notice that $f(0) > 0$. Since A is positive definite, there must exist an $R > 0$ such that $B - RA$ is negative definite. Therefore, $f(R) < 0$. Due to the continuity of $f(r)$, there exists r_* such that $f(r_*) = 0$.

d. Suppose there exists a feasible x_1 such that

$$r_1 := \frac{x_1^TBx_1}{x_1^TAx_1} > r_*$$

Then we must have:

$$
\begin{aligned}
0 \quad &= x_1^T B x_1 - r_1 x_1^T A x_1 \\
&= x_1^T B x_1 - r_* x_1^T A x_1 - (r_1 - r_*) x_1' A x_1 \\
&\leq x_*^T B x_* - r_* x_*^T A x_* - (r_1 - r_*) x_1^T A x_1 \\
&= -(r_1 - r_*) x_1^T A x_1 < 0
\end{aligned}
$$

Therefore, there is a contradiction. □

This theorem tells us that if we can find a root of $f(r)$ and the corresponding maximizer, then we have found the global maximizer of problem (5.2).

The conversion of the ratio minimization problem to a sequence of difference minimization problems has been proposed in solving the trace ratio optimization problem arising in machine learning and high dimensional data analysis [18]. Here in Theorem 5.3.1, we considered the additional l_1 constraint.

5.4 Algorithms

5.4.1 Algorithm for Finding r_*

We used a binary search algorithm.

Algorithm 5 Find r_*

```
Input the matrices $B$ and $A$, a threshold $\epsilon > 0$ for stopping, initial $x_0$, $r_{min}$,$r_{max}$, $i$ and $m$;
Set $r_0 = \frac{1}{2}(r_{min} + r_{max})$;
while $|x_0^T B x_0 - r_0 x_0^T A x_0| > \epsilon$ do
    Solve the quadratic program (5.1) with $r_0$ and obtain the maximizer $x_0$;
    if $x_0^T B x_0 - r_0 x_0^T A x_0 > \epsilon$ then
        $r_{min} = r_0$
    else
        $r_{max} = r_0$
    end if
    $r_0 = \frac{1}{2}(r_{min} + r_{max})$
end while
```

5.4.2 Algorithm for Solving (5.1)

The difficult part is solving the quadratic program (5.1). It is a non-convex optimization problem. Therefore, classical algorithms can not guarantee a global optimizer. In addition, there is scalability issue when the dimension of the problem increases.

One way to tackle this problem is treating the problem as a difference of convex functions and using the DC (difference of convex functions) algorithm.

First, we could reformulate the problem (5.1) in the following way by plugging in the constraint $x(i) = 1$:

$$\underset{\|x\|_1 \leq M'}{\text{minimize}}\; rx^T V x + c' x - x^T U x$$

where U and V are both positive definite matrices and $x \in \mathbb{R}^{n-1}$.

In addition, based on a standard method in DCA, we could change it to an unconstraint problem:

$$\underset{x}{\text{minimize}}\; rx^T V x + c' x - x^T U x + \chi_{\|x\|_1 \leq M'}(x) \tag{5.3}$$

where

$$\chi_{\|x\|_1 \leq M}(x) = \begin{cases} \infty & : \|x\|_1 > M' \\ 0 & : \|x\|_1 \leq M' \end{cases}$$

Let

$$\begin{aligned} g(x) &= rx^T V x + c' x + \chi_{\|x\|_1 \leq M'}(x) \\ h(x) &= x^T U x \end{aligned}$$

then the objective is a difference of g and h. We can use the following DC algorithm:

Algorithm 6 Solve (5.3) for a given r

```
Choose $x^0$ in $\mathbb{R}^{n-1}$;
repeat
    Set $y^k = 2Ux^k$;
    Solve the optimizer $x^{k+1}$ of the convex program:
    (5.4)    $\inf\{rx^TVx + c'x - x^Ty^k + \chi_{\|x\|_1 \le M'}(x), x \in \mathbb{R}^{n-1}\}$
until convergence
```

5.4.3 Algorithms for Solving (5.4)

The problem (5.4) can be considered as a quadratic program with l_1 constraint.

$$\begin{aligned} \min \quad & rx^TVx + c^Tx + x^Ty^k \\ s.t. \quad & \|x\|_1 \le M' \end{aligned} \tag{5.5}$$

It can also be rewritten as a least squares optimization with l_1 constraint. This problem has been well studied in the literature. Now we will present several options for solving it.

1 Use a similar method as algorithm (3). To handle the l_1 constraint, we start with a smaller set of constraint and add new constraints if the solution fails the original one. We refer to algorithm (7).

Algorithm 7 Solve (5.5)

```
Input the parameters $V$, $c$, $y^k$, $r$ and $M'$;
Set $x_0 = (2rV)^{-1}(c + y^k)$
Constraint set = {null}
while $\|x\|_1 > M'$ do
    Add sign($x$) to the constraint set;
    Solve the problem (5.5) by reducing the first $2^{(n-1)^2}$ constraints to the current constraint
    set;
    Update $x$
end while
```

2 Use $|x_i| \approx \sqrt{x_i^2 + e}$ to approximate the l_1 norm, where e is a small number. Now we could break the constraint into two cases. The first case is that the optimizer is in the interior. This implies the optimizer is $(2rV)^{-1}(c + y^k)$. Then we only need to check

whether it satisfies the l_1 norm constraint. If not, then the optimizer must satisfy the equality constraint. Therefore, we could use the method of Lagrange multipliers. The optimizer is the solution to the following non-linear system

$$\begin{cases} 2rV + c + y^k + \lambda\{\frac{x_i}{\sqrt{x_i^2+e}}\} = 0 \\ \sum_{i=1}^{n-1} \sqrt{x_i^2 + e} - M' = 0 \end{cases}$$

where $\{\frac{x_i}{\sqrt{x_i^2+e}}\}$ is a vector with the ith entry $\frac{x_i}{\sqrt{x_i^2+e}}$. This system can be solved by using the Newton's method.

3 A common trick in handling the absolute values is using non-negative variables. We can write $x = x^+ - x^-$ where x^+ and x^- are both non-negative variables. Therefore, problem (5.5) takes the following form:

$$\begin{array}{ll} \min & r(x^+ - x^-)^T V(x^+ - x^-) + c'(x^+ - x^-) + (x^+ - x^-)^T y^k \\ s.t. & \sum x^+ + \sum x^- \leq M' \end{array}$$

In this way, we double the number of variables but reduce to a single constraint. The problem can be solved by active set method.

4

- Reformulate it as a LASSO problem [21]. Notice that the objective of a LASSO problem is

$$\|\mathbf{X}x - \mathbf{Y}\|_2^2 + \lambda|x|_1 = x^T\mathbf{X}^T\mathbf{X}x - 2\mathbf{Y}^T\mathbf{X}x + \mathbf{Y}^T\mathbf{Y} + \lambda|x|_1$$

Therefore, by solving the following system for A and b, we could retrieve a LASSO type problem from (5.4):

$$rV = \mathbf{X}^T\mathbf{X} \qquad c + y = -2\mathbf{X}^T\mathbf{Y}$$

The first equation can be solved by Cholesky decomposition and then the second equation is easy to solve. Finally, we could apply the least angel regression algorithm (LARS) [5].

Chapter 6

Recover the Fastest Mean Reverting OU Process

6.1 Motivation

In this section, we formed a new type of problems. Suppose we have several times series which we believe they are the linear combinations of some hidden mean reverting processes. We would like to recover the fastest mean reverting process given these observations.

This problem can be considered an inverse problem of constructing sparse and mean reverting portfolios.

6.2 Formulation 1: Recover the OU Process with an Abnormal Starting Value

Assume that v_t, X_{t1}, X_{t2}, ..., X_{tk} satisfy the following OU processes:

$$dX_{tk} = -\lambda_k X_{tk} dt + \sigma_k dB_{tk} \qquad dv_t = -\lambda v_t dt + \sigma dB_t$$

where dB_{tk} are the independent noises. We assume that $\lambda >> \lambda_i$ for $i = 1, ..., k$. Suppose we have a data set of n periods of the linear combinations of these $k + 1$ OU processes, we wonder recover v_t and further find its mean reversion coefficient, i.e.

$$\begin{aligned} &\text{Given} \quad W = \text{span}\{v_t, X_{t1}, X_{t2}, ..., X_{tk}\}, \text{ for } \quad t = 0, ..., n, \ |v_0| >> 0 \\ &\text{Find} \quad v_t \end{aligned}$$

6.3 Formulation 2: Recover the OU Process with a Jump Process

Consider $k+1$ stochastic processes $\{v_t, X_{t1}, X_{t2}, ..., X_{tk}\}$. v_t is the one we would like to recover. We assume it follows:

$$dv_t = -\lambda v_t dt + \sigma dB_t + dN$$

where N counts the number of jumps that have occurred and the random jump magnitude is 1. We assume the jump process follows a Poisson process with parameter θ.

For $\{X_{t1}, X_{t2}, ..., X_{tk}\}$, we assume they are independent OU processes with relatively small mean reverting coefficients:

$$dX_{tk} = -\lambda_k X_{tk} dt + \sigma_k dB_{tk}$$

Our goal is still recover v_t given a space spanned by $\{v_t, X_{t1}, X_{t2}, ..., X_{tk}\}$, i.e.

$$\begin{aligned} &\text{Given} \quad W = \text{span}\{v_t, X_{t1}, X_{t2}, ..., X_{tk}\}, \text{for} \quad t = 0, ..., n \\ &\text{Find} \quad v_t \end{aligned}$$

6.4 Methodology

In order to solve the previous problems, we used a method proposed by Laurent Demanet and Paul Hand [9], which is used for recovering the sparsest element in a subspace.

This method is trying to solve the following problem: given an arbitrary basis of $W \subset \mathbb{R}^n$, find x, the sparsest nonzero element in W.

The authors transform this problem to n optimization problems:

$$\begin{aligned} &\min && \|z\|_1 \\ &\text{subject to} && z \in W, z(i) = 1 \end{aligned} \tag{6.1}$$

for each $1 \leq i \leq n$, where

$$W = \operatorname{span}\{w_1, w_2, ..., w_{k+1}\} = \operatorname{span}\{v, v_1, v_2, ..., v_k\}$$

i is fixed for a single problem and all the entries of v_i's are i.i.d. random variables with standard normal distribution. The goal is to find v given the basis $w_1, w_2, ..., w_{k+1}$. Note that $z \in W$ is equivalent to $z \perp W^{\perp}$ and therefore there will be $n - k - 1$ equations.

This problem can be considered as a quadratic program. The authors found the conditions for exact and stable recovery. For details, we would refer [9].

This method works on our problem, since it is just a generalization from a set of independent normal random vectors to a set of correlated normal random vectors.

Chapter 7

Numerical Tests

Our codes are implemented in Matlab R2011b. We used the optimization package YALMIP. Computations are performed on a Dell desktop with 8G RAM and 3.4 GHz i7 CPU. We have used two historical data sets. One is the U.S. daily swaps data for maturities 1Y, 2Y, 3Y, 4Y, 5Y, 7Y, 10Y and 30Y from July 3, 2000 until July 15, 2005. The data are obtained from **www.Economagic.com**. The total number of data is 1257×8. The data are in percent with two digits after the decimal point. The other one is the daily closed prices of S&P 500 companies. The data are collected from Yahoo finance. In order to obtain a large sample set, we only select those companies that remain on the list since July 2005. After this preselection procedure, we have 458 companies left. The data size in our numerical test is 2000×458.

7.1 Comparison of Two Mean Reverting Proxies

We first wanted to compare the performance of portfolio selection via two proxies that we discussed in Chapter 2.

By presetting the matrix β and the noise covariance matrix Σ in the VAR(1) model, we generated a data set of size 350×8 each time which means there are 350 observations for each asset and there are 8 assets in total. We used the first 100×8 samples as the training set and the rest as the test set. Next we made estimations of parameters and solved for the optimal solutions by the exhaustive search method based on the training set. We repeated our simulation 1000 times and then we compared the average of the estimated mean reversion coefficients of our sparse portfolios on both the training set and the test set.

Next we tested the performance of the sparse portfolios based on a simple convergence trading strategy. In most of the application of convergence trading, investors will consider two parameters μ and τ, where μ is the estimated average asset value and τ is the tolerance of mispricing. In [8], the authors developed a strategy that only takes advantages of underpricing of the portfolio. We generalized their strategy and took advantages of both the underpricing and overpricing of the portfolio. For simplicity, we also assumed that we have the ability to buy and sell assets without any transaction costs and we also have the ability to short sell. Since it will be difficult to calculate the return if we introduce the action of handling overpricing, we will use the number of trading opportunities to show the performance of the portfolio. The trading opportunity means the observation that the price converges after out-of-tolerance mispricing. We use K to denote the number of trading opportunities and P_t to denote the portfolio value at time t.

The trading strategy can be summarized as follows:

- If the observed sample $P_t > \mu + \tau$, we will sell our portfolio if we already hold one and $K = K + 1$. We will short this portfolio if we didn't short it before. Otherwise we perform no action.

- If the observed sample $P_t < \mu - \tau$, we will go long our portfolio if we already short one and $K = K + 1$. We will buy this portfolio if we didn't hold it before. Otherwise we perform no action.

- If the observed sample $\mu - \tau \leq P_t \leq \mu + \tau$, we will go long our portfolio if we already short one and $K = K + 1$. We will sell our portfolio if we already hold one and $K = K + 1$. Otherwise we perform no action.

Figure 7.1 will be helpful in understanding the trading strategy. The X-axis shows the time periods are from day 1 to day 60. The Y-axis shows the values of the portfolio. The green dashed line is $y = \mu$, the red solid line (overpriced bound) is $y = \mu + \tau$ and the teal solid line (underpriced bound) is $y = \mu - \tau$.

The trading opportunities will increase by 1 if we have a high price or low price before returning to the normal range.

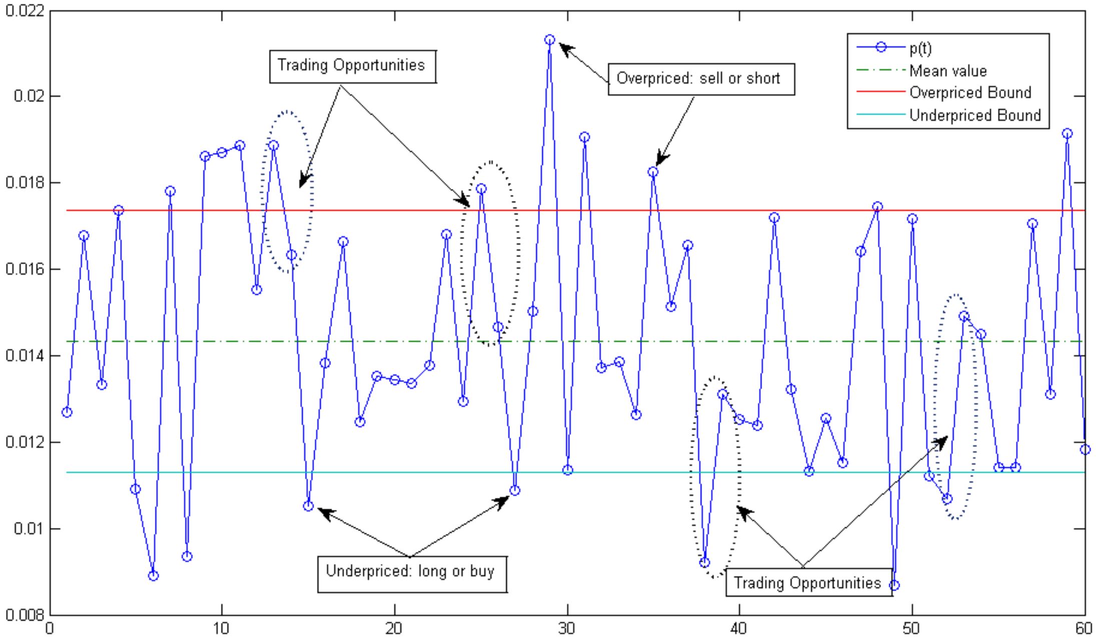

Figure 7.1: Trading Opportunities

The simulation test is similar as before. After we found the optimal solutions by the exhaustive search method, we tested the trading strategy only on the test set and counted the trading opportunities. We repeated 1000 times and calculated the averages. The estimation of μ and τ is based on the training set. We set μ as the sample mean and the τ as the sample standard deviation. The results are shown in Figure 7.2. We can see that the solutions under direct OU estimator always perform better than that under the predictability.

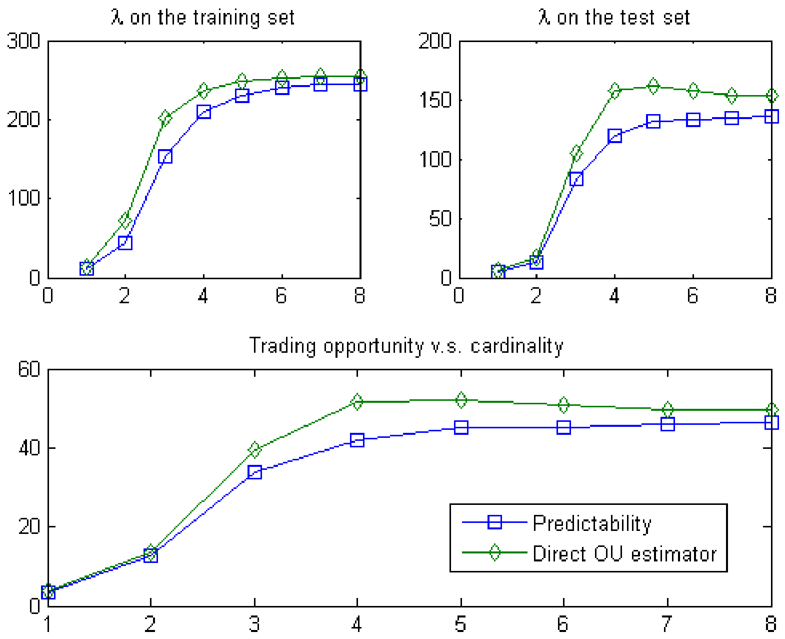

Figure 7.2: Comparison of the performance of sparse mean reverting portfolios solved under different proxies

7.2 Tests of the Ratio of l_1 and l_2 Norms Approach

In this section, we will present numerical results of solving the penalized optimization problem 4.4 based on historical data. We used two data sets. One is the U.S. swap rate data and the other one is a selected set of S&P 500 stocks.

One issue is determining the range of γ. We proposed a simple method by only calculating the solutions whose cardinalities are 1 and n and setting γ as the value to make $f(x_1, \gamma) = f(x_n, 0)$. This is computationally efficient. Using this method, we found that the range of γ for U.S. swap rate data is 0 to 1.5 and the range of γ for S&P 500 data is 0 to 0.8.

7.2.1 U.S. Swaps Data

In this section, we study the U.S. daily swap rate data for maturities 1Y, 2Y, 3Y, 4Y, 5Y, 7Y, 10Y and 30Y.

We performed the following tests on the U.S. swaps data.

First, we used the whole data set to estimate all the parameters and calculated the solutions for different γ. We found the minimizers by the algorithm in chapter 4 for different γ's between 0 and 1.5 with a step size 0.02. Then we calculated the estimated mean reversion coefficients and counted the trading opportunities of the whole period for all the minimizers. The results are shown in the figure 7.3.

Secondly, we used every 100 observations to estimate those parameters and also found the minimizers for different γ's between 0 and 1.5 with a step size 0.02. This time, we calculated the estimated mean reversion coefficients and counted the trading opportunities for both these 100 days and the next 100 trading days. The results are shown in the figure 7.4.

We solve the following problem:

$$x^T Ax/x^T Bx + \gamma \frac{\|x\|_1}{\|x\|_2}$$

by using intermittent diffusion algorithm (ID algorithm). After each iteration of the line search algorithm, we fix the l_2 norm to be 1.

In our tests of ID algorithm, we let $\alpha = 20$, $\kappa = 20$ and $N = 20$. In addition, to satisfy the conditions of ID algorithm, we also add a penalty function $p(x, \theta, \xi, \zeta)$ to $f(x)$:

$$p(x, \theta, \xi, \zeta) = \sum_i u(x_i, \theta, \xi, \zeta)$$

where

$$u(x_i, \theta, \xi, \zeta) := \begin{cases} \xi(x_i - \theta)^\zeta, & x_i > \theta \\ 0, & |x_i| \leq \theta \\ \xi(x_i - \theta)^\zeta, & x_i < -\theta \end{cases}$$

We set $\theta = 10$, $\xi = 2$ and $\zeta = 100$ in our test.

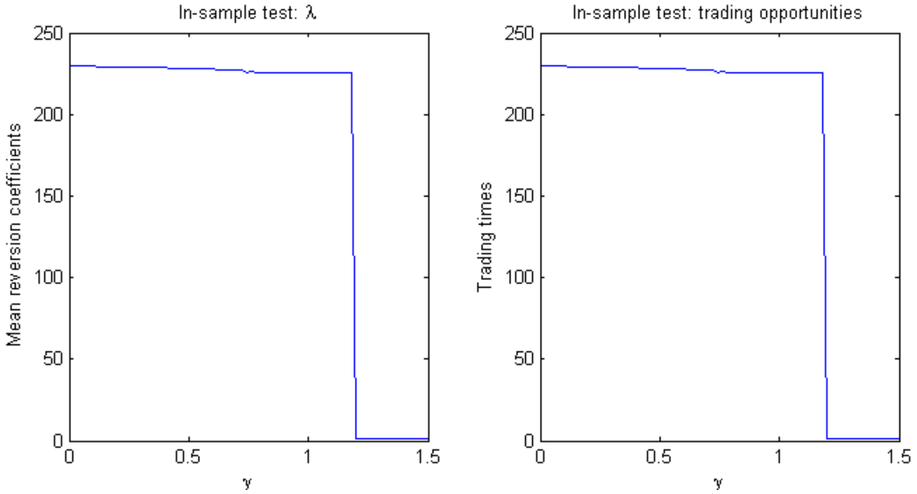

Figure 7.3: In-sample tests on the whole data set of the U.S. swaps. The x-axis shows the value of γ and the y-axis shows the estimated mean reversion coefficients and the trading opportunities

From the numerical results, we could see that as the γ increases the mean reversion coefficients and the trading opportunities have a decreasing trend for both in-sample and out-of-sample tests. There are big jumps in Figure 7.3 which means that we failed to recover some portfolios with intermediate cardinalities.

The curves in Figure 7.4 look smoother, since they are the average performance of portfolios on different data set. Normally, the in-sample performance is better than the out-of-sample performance. However, we could notice that we can still maintain about 60% of the performance.

When γ is close to 1.5, the minimizer will be an 1-sparse vector. This shows that the ratio of l_1 and l_2 norms indeed enforces extreme sparsity in our problem.

The ratio of l_1 and l_2 norms could be considered as l_2 normalized transaction costs. It encourages investors use a small number of portfolios and thus investors could spend less effort in keeping track of the prices of portfolios. One extreme case is that we compare an 1-sparse vector $x_1 = (0, 0, ..., 0, 0, 1)^T$ and a uniform vector $x_n = (\frac{1}{n}, \frac{1}{n}, ..., \frac{1}{n}, \frac{1}{n})^T$ under this penalty. They obtained the minimum and maximum of the ratios of l_1 and l_2 norms

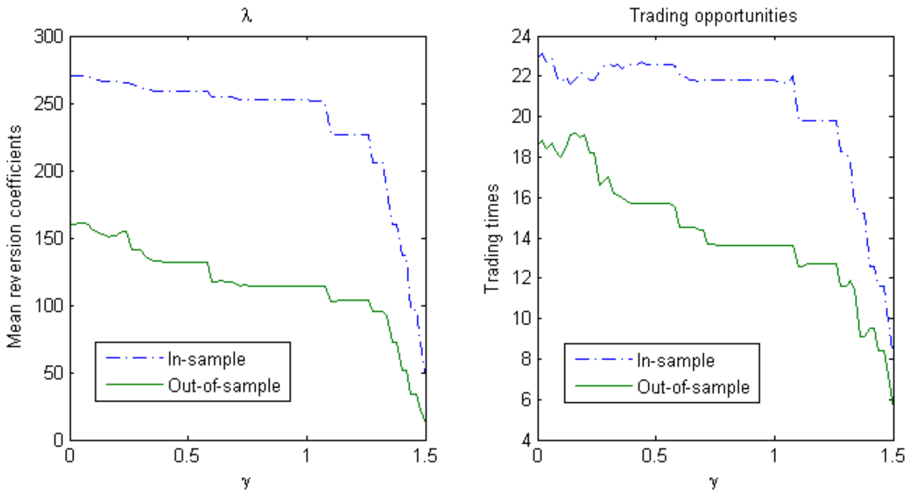

Figure 7.4: In-sample v.s. out-of-sample tests on the U.S. swaps. The x-axis shows the value of γ and the y-axis shows the average estimated mean reversion coefficients and the average trading opportunities

respectively. However, their l_1 norms are identical and if we believe that the transaction costs are consistent and only depending on the trading volumes, then the vectors have the same transaction costs. x_1 will be preferred under our penalty (given a large enough γ), since investors only need to keep track of one portfolio. Therefore, we think the ratio of l_1 and l_2 norms could be also considered as a penalty for the working load.

7.2.2 S&P 500 Data

In this section, we want to apply our model to the stock prices of S&P 500 companies. The data are collected from Yahoo finance.

Normally, the stock price is a non-stationary time series. However, their linear combinations could be stationary. Therefore, direct OU estimator will be more appropriate due to the lack of stationarity of the data.

We performed the similar tests as we did for the U.S. swap rate data. The numerical results are shown in figure 7.5. We could find that as γ increases the estimated mean reversion coefficients and trading opportunities are decreasing. In addition, figure 7.6 shows the minimizers under different γ's. From this plot, we can see that the ratio of l_1 and l_2

norms does enforce sparsity.

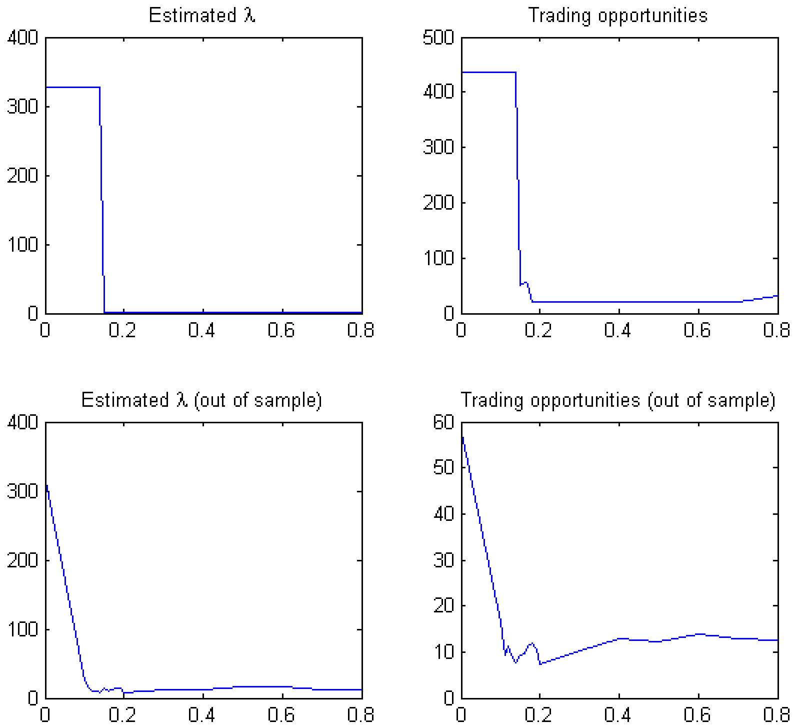

Figure 7.5: Comparison of the performance of sparse mean reverting portfolios of S&P 500 stocks.

7.2.3 Discussion of the l_1/l_2 penalty

We would like to use our numerical results to show the non-increasing property of $\frac{\|x(\gamma)\|_1}{\|x(\gamma)\|_2}$. It is shown in figure 7.7. Therefore, lemma 4.4.3 is numerically verified.

The downward trend of the cardinality of the minimizers are observed as γ increases. However, we also observed some jumps in-between. We attributes this phenomenon to two reasons:

- It is very difficult to check the conditions of theorem 4.4.1;
- The minimizers from ID algorithm are still sub-optimal, so they may not be the global minimizer.

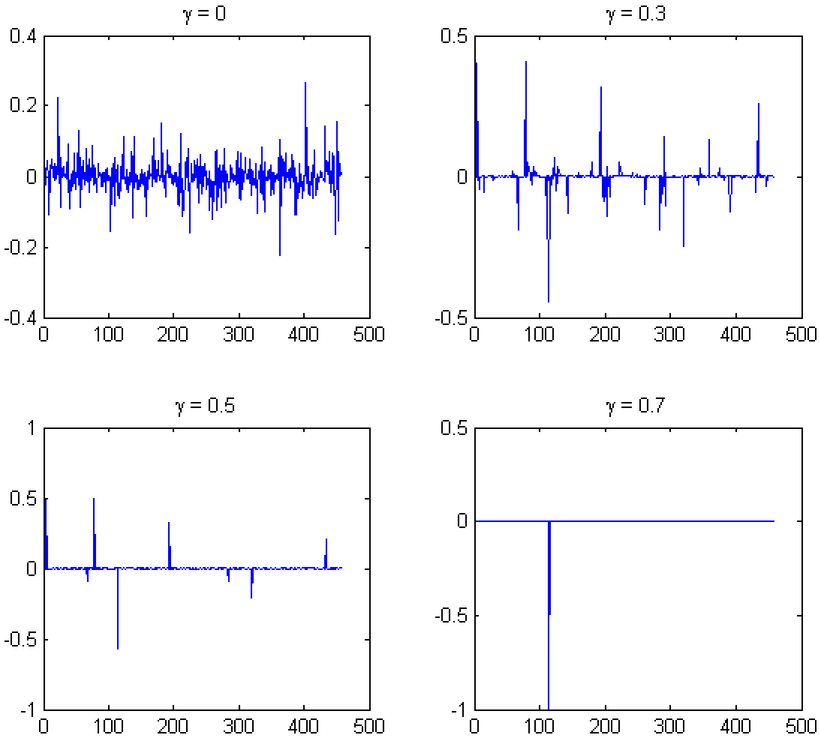

Figure 7.6: One set of solutions on the S&P 500 data. The Y-axis represents the coefficients.

Therefore a more efficient algorithm in solving for a global minimizer is needed.

We could notice that there is always a huge jump in the values of l_1/l_2. We tried to shrink the step size of γ but we still cannot recover more solutions with different cardinality. We think this is probably a certain property of the ratio penalty that it prefers extreme cases.

7.2.4 Comments on the Ratio of l_1 and l_2 Norms

The advantages of the ratio of l_1 and l_2 norms approach are: 1. It does not require any prior knowledge of the assets and therefore investors will not be misled by their previous experience; 2. It does not predetermine the cardinality. This approach also has its disadvantages: 1. It is difficult to recover a portfolio whose cardinalities are intermediate. We could encounter big jumps of the mean reversion coefficients and trading opportunities as we change the tuning parameter γ; 2. The algorithm is not very efficient. It takes more

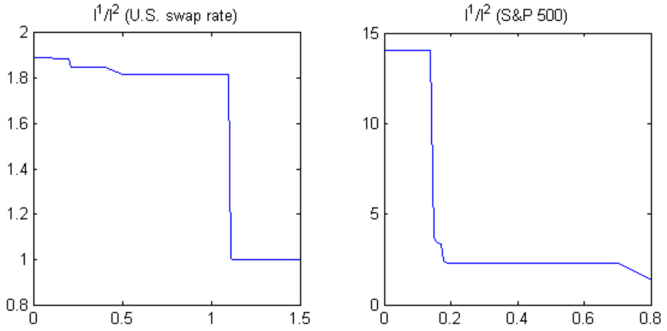

Figure 7.7: The values of the penalty term of two solutions sets

than 5 minutes in solving a problem of size 100 (100 different assets for us to choose). If we compare this speed with methods of the next section, it is relatively slow.

7.3 Tests of the Methods in Chapter 5

7.3.1 In-Sample Tests on U.S. Swaps

In order to test the performance of our methods, we need to set up a criterion. One choice is the solution obtained by the exhaustive search method. They serve as the reference global optima.

Table 7.1 shows the solutions obtained by the exhaustive search method based on the whole U.S. swaps data. The mean reversion coefficients, $\hat{\lambda}$'s, are estimated based on the whole data set.

Notice that the 4Y swap rate has the largest weight in absolute value for portfolios with cardinality of 4 to 8. Therefore, setting the weights of the 4Y swap to be 1 is a potentially good strategy.

The key part of the algorithms in Chapter 5 is how to solve the non-convex quadratic program 5.1 given r and i:

$$\underset{x(i)=1,\|x\|_1\leq m}{\text{maximize}}\; x^T Bx - rx^T Ax$$

$\|x\|_0$	1	2	3	4	5	6	7	8
1Y	0.000	0.000	0.000	0.000	0.000	0.000	-0.003	0.003
2Y	0.000	0.000	0.000	-0.111	-0.082	0.080	0.095	-0.094
3Y	0.000	0.000	0.000	0.522	0.425	-0.402	-0.426	0.425
4Y	0.000	0.673	0.452	-0.766	-0.754	0.732	0.735	-0.736
5Y	0.000	-0.740	-0.812	0.357	0.488	-0.527	-0.506	0.506
7Y	0.000	0.000	0.369	0.000	-0.077	0.130	0.116	-0.113
10Y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.002
30Y	1.000	0.000	0.000	0.000	0.000	-0.014	-0.012	0.012
$\hat{\lambda}$	1.35	5.27	102.30	204.12	226.38	229.18	229.65	229.66

Table 7.1: Solutions obtained by the exhaustive search method

We have mainly tested the following four methods:

A Directly use the active set method for quadratic programming and handle the l_1 constraint by using non-negative variables;

B Use the DC algorithm and Tibshirani's method, i.e. Algorithm 5 and 6 (option 2 in section 5.4.3);

C Use the DC algorithm and Non-negative variable method, i.e. Algorithm 5 and option 3 in section 5.4.3;

D Use the DC algorithm and the least angel regression (LARS) algorithm, i.e. Algorithm 5 and option 4 in section 5.4.3;

We have tested different l_1 levels and constructed different sparse mean reverting portfolios. We estimated the mean reversion coefficients on these portfolios. Here we only present one table of our portfolios constructed using different methods. For more numerical results, we refer to the tables in the appendix.

One thing to note is that LARS algorithm solves for a solution path under l_1 penalty multiplied by a scale parameter. Therefore, we picked the solution x from LARS results which is slightly over the l_1 constraint. This will not affect our ultimate goal: controlling the $\|x\|_0$, since the solution path of the LARS algorithm normally will increase at most one cardinality in each step. This may lead to an unfair comparison between the method D with methods A, B and C. However, we could compare their asset selection to see whether there is any advantage of one method over the other.

From the the table 7.2, we may find that the DC algorithm does help improve the performance of the results. However, it increases the computational costs a lot. Normally, Tibshirani's method is not as efficient as the Non-negative variable method and this is just as what Tibshirani states [21]. Surprisingly, the method D (LARS algorithm in conjunction with the DC algorithm) is the fastest method. Although its estimated mean reverting coefficient, λ, is the largest, it does not mean that it has advantages in picking assets. The reason is just that it slightly breaks the l_1 constraint as we stated before. However, we could see that all the methods have the same choice of assets. Therefore, we could say that they actually construct the same portfolio. After considering the computation efficiency, we conclude that the method D is our best choice.

Finally, we would like to compare our results with the table 7.1. Methods A, B, C and D suggest us picking the 1Y, 3Y, 4Y, 5Y and 30Y swaps. It is of cardinality 5. The best portfolio of cardinality 5 uses the 2Y, 3Y, 4Y, 5Y and 7Y swaps. Therefore, we could say that our methods have a correct rate 60% since they pick 3Y, 4Y and 5Y. In addition, we could see that these three swaps have the leading weights in the portfolio. Our methods select the main components of the optimal portfolio. We believe the difference of our portfolio and the optimal portfolio of the same cardinality is due to the difference between l_1 norm and the cardinality. If we normalize the weights of the optimal portfolio such that its 4Y swap is 1, then its l_1 norm will be 2.425 which is much larger then our l_1 constraint 2.05. Therefore, our portfolio is the optimal one under l_1 constraint.

7.3.2 In-Sample Tests on 100 Stocks

The efficiency of the method D is better shown for a large dimensional data. In the next test, we applied the method D to a data set of 100 stocks. These stocks are the first 100 S&P stocks in ticker symbols' alphabetical order from our pre-selected list of S& P 500 stocks. Therefore, the size of the matrices A and B is 100×100.

Method	A	B	C	D
1Y	0.021	0.020	0.020	0.019
2Y	0.000	0.000	0.000	0.000
3Y	-0.475	-0.468	-0.468	-0.423
4Y	1.000	1.000	1.000	1.000
5Y	-0.554	-0.560	-0.560	-0.630
7Y	0.000	0.000	0.000	0.000
10Y	0.000	0.000	0.000	0.000
30Y	0.001	0.001	0.001	0.037
λ	149.054	149.185	149.185	185.002
Time(s)	0.446	10.404	3.338	0.033

Table 7.2: Solutions of method A, B, C and D obtained under $\|x\|_1 \leq 2.05$

When the problem is of this size, we are not able to use the exhaustive search method to get the optimal solution for the middle cardinalities. Therefore, in order to set up a criterion, we will compare our results with the densest solution. This solution should give us the largest possible mean reversion coefficient based on the data set.

For our data set, the largest possible mean reversion coefficient is 109.97. In addition, we will set the weight of the No. 43 stock (ticker symbol: AMAT) to be 1, since it has the largest weight among all.

Figure 7.8, 7.9, 7.10 and 7.11 are the numerical results. The l_1 constraints are in the range of 3 to 16 with a step size 1. Figure 7.8 demonstrates the mean reversion coefficients of portfolios built under different l_1 constraints. The X-axis shows the level of l_1 norm and the Y-axis is the mean reversion coefficients. Figure 7.9 demonstrates the cardinality of those portfolios and figure 7.10 shows the trading opportunities. Figure 7.11 displays the weights of 100 stocks.

The average computational cost of the method D of all these tests is 4.928 seconds. This is a remarkable improvement over the other computational methods to date in terms of efficiency and speed.

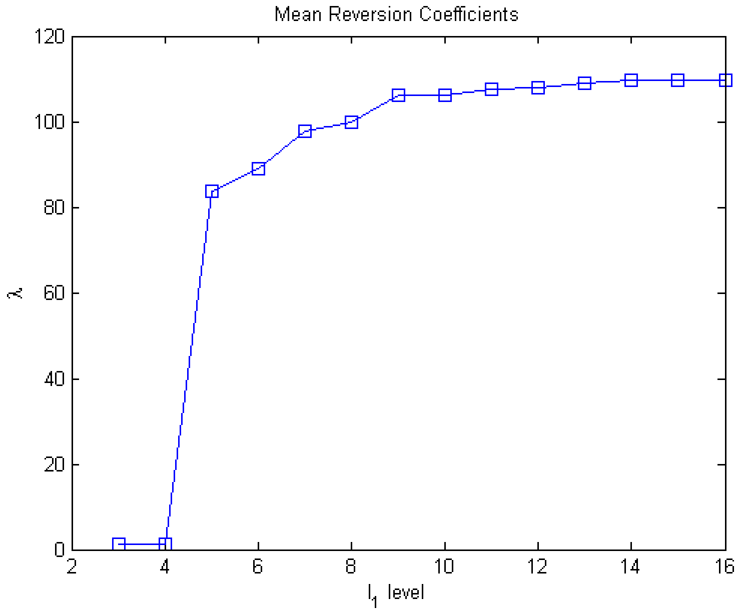

Figure 7.8: Mean reversion coefficients under different l_1 constraints

7.3.3 Out-of-Sample Tests on the U.S. Swaps

We also performed out-of-sample tests. We still worked on the U.S. swap data. Each time, we used 100 days data as the training data and built a sparse mean reverting portfolio. Then we estimated the mean reversion coefficients of the portfolio on these 100 days, next 50 days and next 100 days.

Figure 7.12 shows the numerical results of the average λ and Figure 7.13 shows the numerical results of the average trading opportunities. Both the X-axes are the l_1 level. The range is from 1.5 to 2.7 with a step size 0.1. For each level, we performed several tests and the Y-axis is the average of the estimated mean reversion coefficients and the trading opportunities of all these tests.

From Figure 7.12, we see that as expected the in-sample performance is better. Our portfolios could maintain about 70% of the in-sample mean reversion coefficients during the 50 out-of-sample days and maintain about 65% of the in-sample mean reversion coefficients during the 100 out-of-sample days.

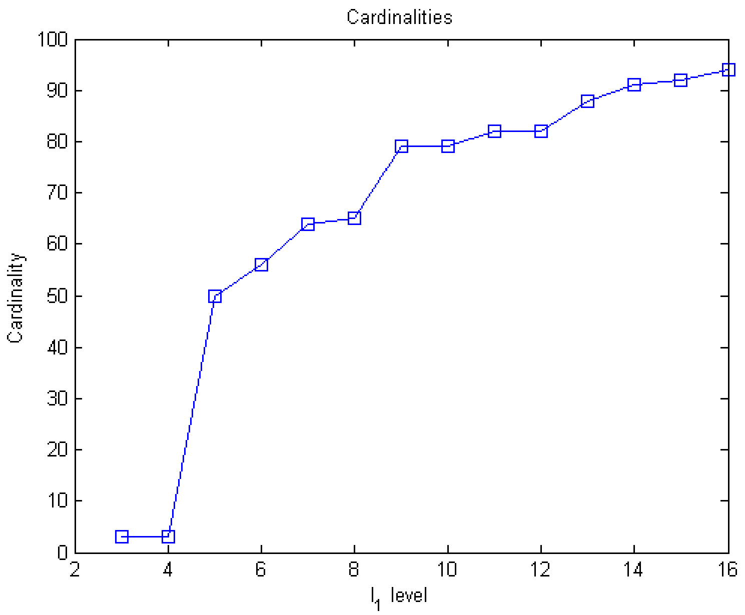

Figure 7.9: Cardinalities under different l_1 constraints

In order to make a fair comparison of the trading opportunities, we performed the following counts. We counted the trading opportunities of the last 50 days of the training period and compare it with the trading opportunities of the next 50 days. We counted the trading opportunities of the last 100 days of the training period and compare it with the trading opportunities of the next 100 days. In this experiment, the total length of trading days are identical for in-sample and out-of-sample tests. We find that in both cases our portfolios can maintain about 75% of the in-sample trading opportunities during the out-of-sample period.

7.3.4 Out-of-Sample Tests on High Dimensional Simulated Data

In this section, we perform an out-of-sample test on high dimensional simulated data.

By presetting the matrix β and the noise covariance matrix Σ in the VAR(1) model, we generated a data set of size 400×100. We consider this set as the training set. we estimated of the matrices A and B and solved for the optimal solutions by the least angle regression and DC algorithm based on the training set.

After this, we generated 400×100 observations based on the same VAR(1) model in each

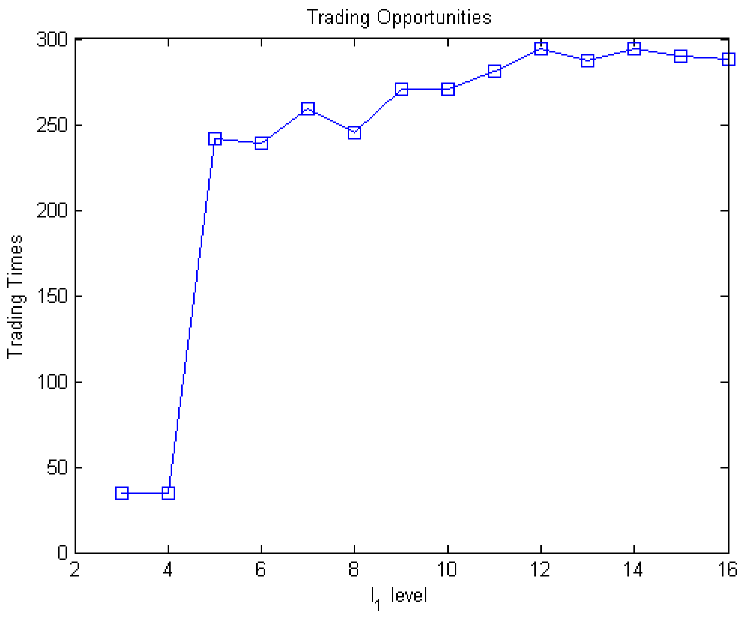

Figure 7.10: Trading Opportunities

trial. These are test sets on which to evaluate our constructed portfolios.

We generated 100 different test sets and then we compared the average of the estimated mean reversion coefficients and trading opportunities of our sparse portfolios on both the training set and the test set. When we counted the out-of-sample trading opportunities, we still used the mean and standard deviation of the training set, since we are not supposed to know the future mean or variance.

The results are shown in Figure 7.14. We notice that it is very hard to maintain a high level of mean reversion coefficients when the dimension of the problem is high. They are about 10% of the in-sample levels. However, the numbers of trading opportunities do not decrease such dramatically. They are about 40% of the original. In fact, this is more important since the profits of convergence trading strategy come directly from those trading opportunities.

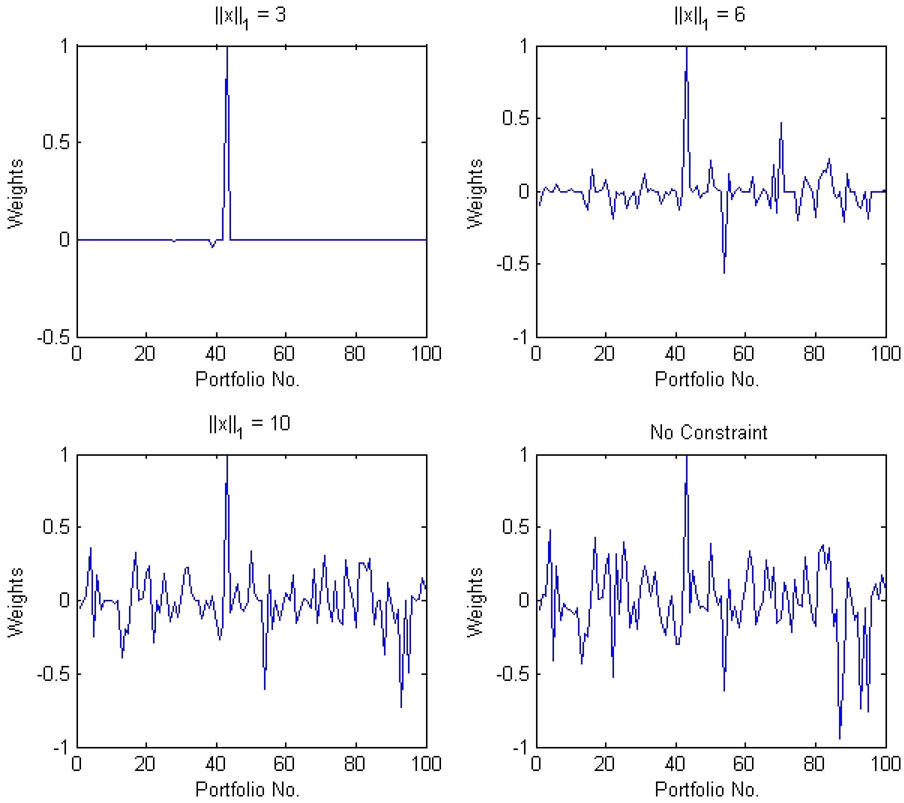

Figure 7.11: Solutions under different l_1 constraints

7.4 Numerical Results of Recovering Fast Mean Reverting Process

7.4.1 Recover the OU Process with an Abnormal Starting Value

We assume that v_t, X_{t1}, X_{t2}, ..., X_{tk} satisfy the following OU processes:

$$dX_{tk} = -\lambda_k X_{tk} dt + \sigma_k dB_{tk} \qquad dv_t = -\lambda v_t dt + \sigma dB_t$$

where dB_{tk} are the independent noises. For each process, we have n observations.

In our tests, we used the following parameters:

- $\sigma = \sigma_1 = ... = \sigma_k = 1;$

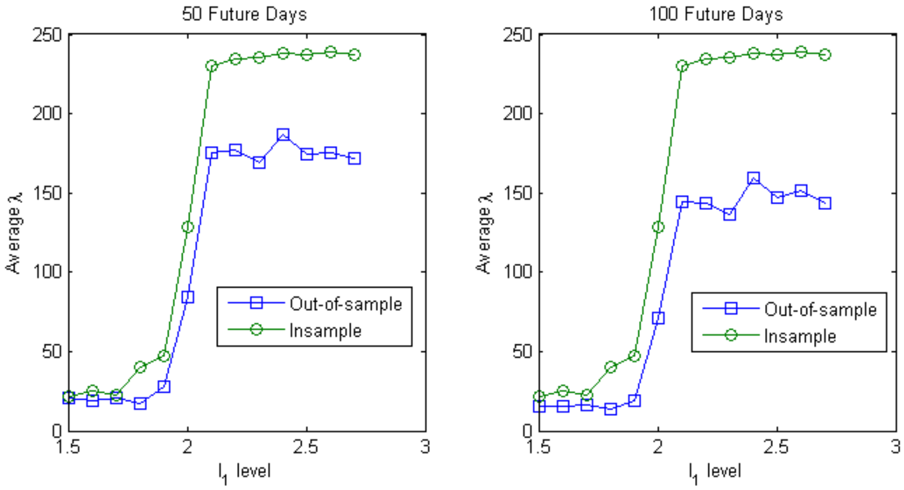

Figure 7.12: In-Sample vs. Out-of-Sample Tests: λ

- $v_0 = X_{01} = ... = X_{0k} = 1$, i.e. all the processes start at 1;
- λ_i's are some random integers between 1 to 7 and $\lambda = 200$;

In the original paper, since the position of i is unknown, the author will solve n of the following quadratic optimization problems for $1 \leq i \leq n$:

$$\begin{array}{ll} \min & \|z\|_1 \\ \text{subject to} & z^T \perp X_{tk}, z(i) = 1 \end{array} \tag{7.1}$$

In out test, we only need to solve for one problem, because we know that $i = 1$.

The results are shown in figure 7.15 and 7.16. The x-axis shows the time step and y-axis shows the value at each time step.

In figure 7.15, we present the recovery results with $k = 20$ and $n = 99$. According to [9], this k is still too large (they require $k \leq n/32$. However, the estimated mean reversion coefficients are already close. The original path is 191 and the recover path is 268.

A similar test result on a large data set is shown in figure 7.16. This time, we set $k = 20$ and $n = 639$. According to [9], this k is small enough. The estimated mean reversion coeffi-

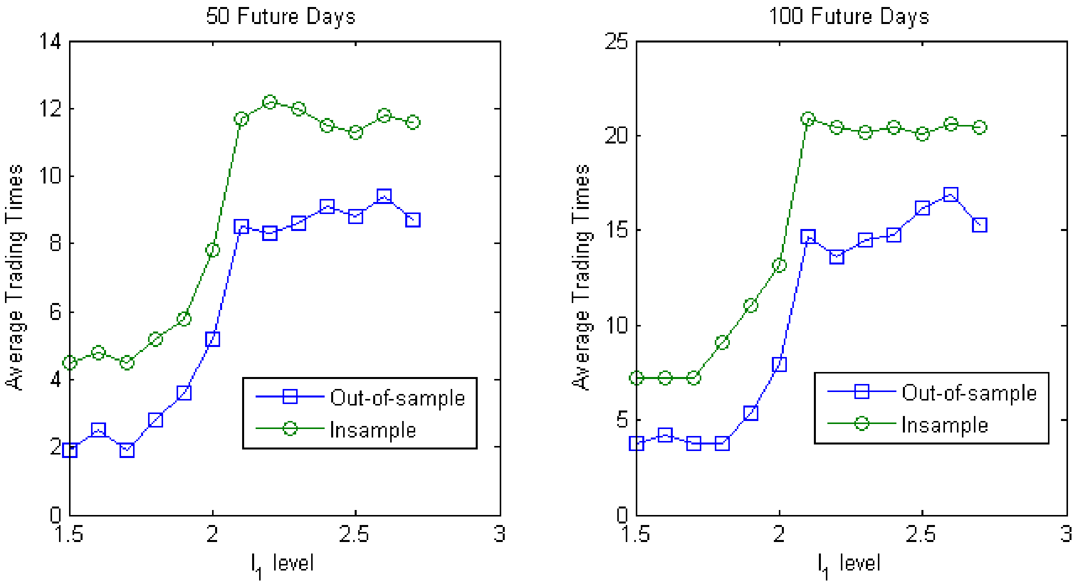

Figure 7.13: In-Sample vs. Out-of-Sample Tests: Trading Opportunities

cients are closer. The original path is 203 and the recover path is 197.

All these problems are solved by the YALMIP package.

7.4.2 Recover the OU Process with a Jump Process

In these tests, we assume it follows:

$$dv_t = -\lambda v_t dt + \sigma dB_t + dN$$

where N counts the number of jumps that have occurred and the random jump magnitude is 1. We assume the jump process follows a Poisson process with parameter θ.

For $\{X_{t1}, X_{t2}, ..., X_{tk}\}$, we assume they are independent OU processes with relatively small mean reverting coefficients:

$$dX_{tk} = -\lambda_k X_{tk} dt + \sigma_k dB_{tk}$$

where for each process, we have n observations.

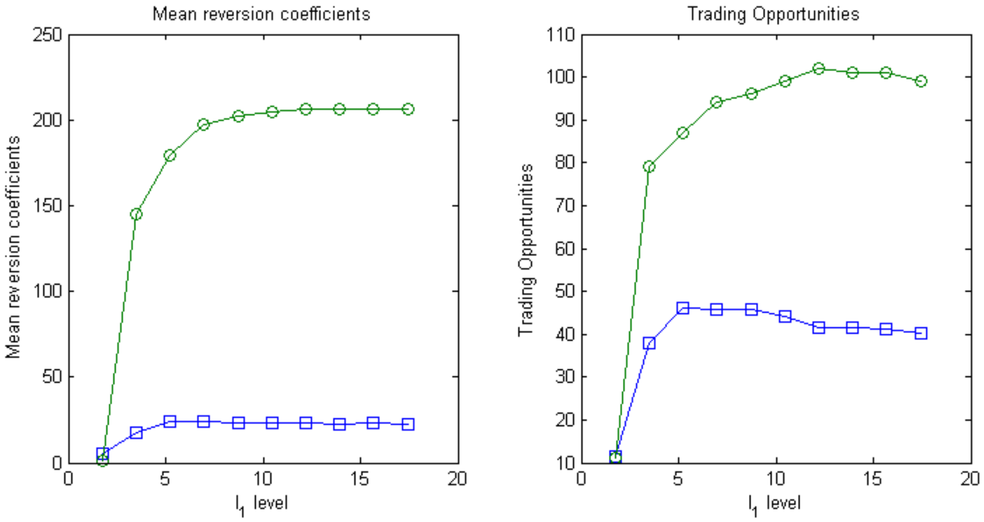

Figure 7.14: Solutions under different l_1 constraints

We used the following parameters:

- All the processes start at 0;
- $\sigma = \sigma_1 = ... = \sigma_k = 1$;
- $k = 14$ and the length of sample path is 450;
- λ_i's are some random integers between 1 to 7 and $\lambda = 300$;
- The parameter of the Poisson process is 10;

This time, since we do not know when and how many jumps happen, we will have to solve n of the following quadratic optimization problems for $1 \leq i \leq n$:

$$\begin{array}{ll} \min & \|z\|_1 \\ \text{subject to} & z^T \perp X_{tk}, z(i) = 1 \end{array} \tag{7.2}$$

Numerical results are presented in table 7.3, figure 7.17, 7.18 and 7.19.

In figure 7.17, we gave the plots of 4 sample paths in the data set, while in total, we have 14 paths. The path with jumps is the one we want to recover. From our simulation

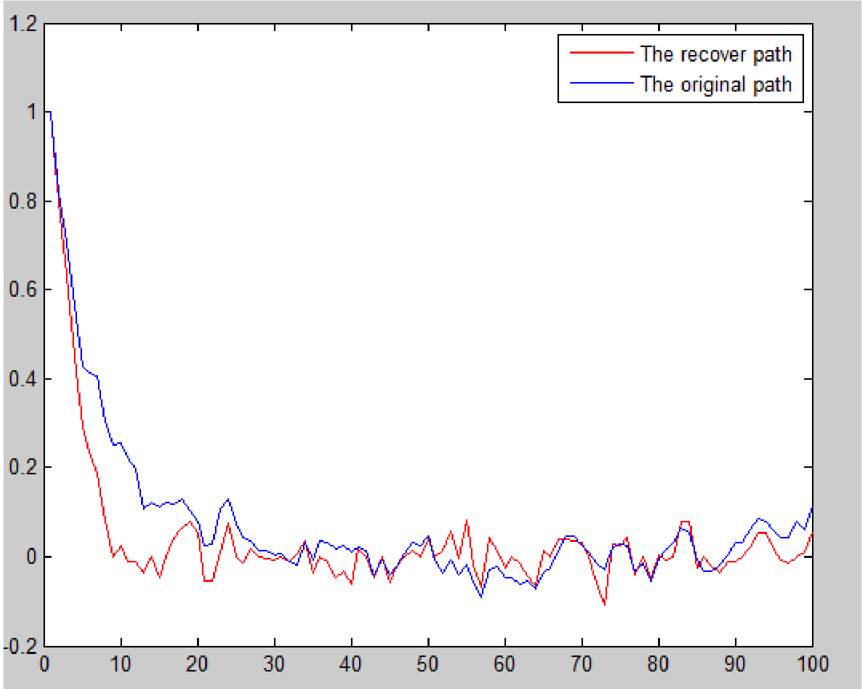

Figure 7.15: Original Path v.s. Recovered Path When $k = 20$ and $n = 99$.

results, that path have 5 jumps and they happen at time step 223, 370, 371, 423 and 437 (it is difficult to see 5 jumps since two jumps happen in a short time). Therefore, we should expect that we could get a good recovered path when we set i equal to these numbers. The numerical results are just as we expected. Please refer to table 7.3.

In figure 7.18, we gave three paths. The path with the best MR means that the path has the closest mean reversion coefficients to the simulation parameter $\lambda = 300$ among all the solutions obtained under different i. This path is recovered when $i = 437$. The path with the best l_1 norm means that the path has the smallest l_1 norm among all the solutions obtained under different i. This path is recovered when $i = 371$.

In figure 7.19, we presented the l_1 norms of the recovered sample paths under different i. The x-axis is the index i and the y-axis shows the corresponding the l_1 norms of the recovered sample path. We could found that it found all the right jumps. The l_1 norm is relatively low when we set i at those jumps.

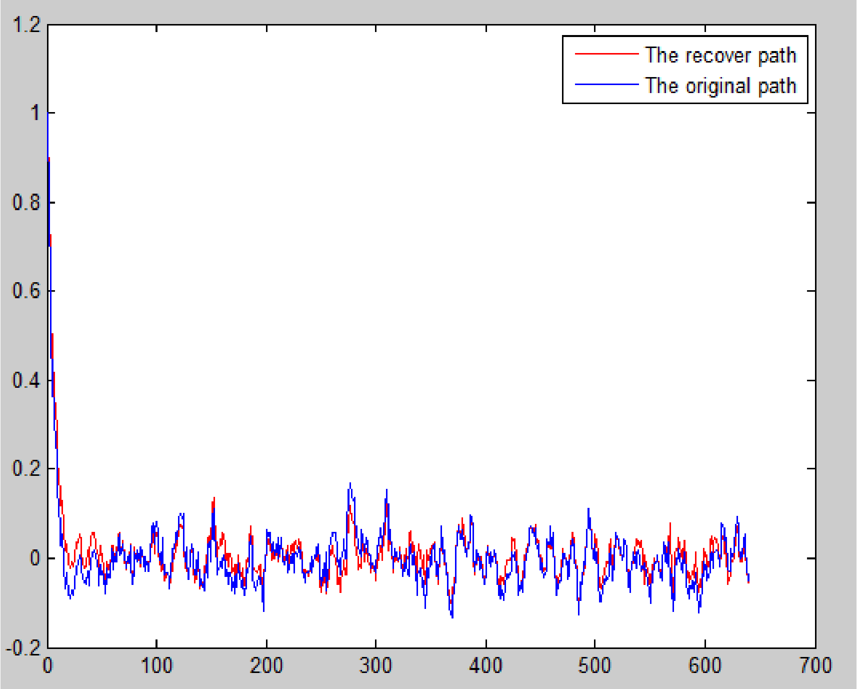

Figure 7.16: Original Path v.s. Recovered Path When $k = 20$ and $n = 639$.

Jump time	l^1 norm of recovered path	Estimated Mean Reversion Coefficient
223	27.54	400.63
370	27.53	409.32
371	16.63	368.81
423	28.95	370.47
437	27.46	325.30

Table 7.3: Recovering OU process with Jumps

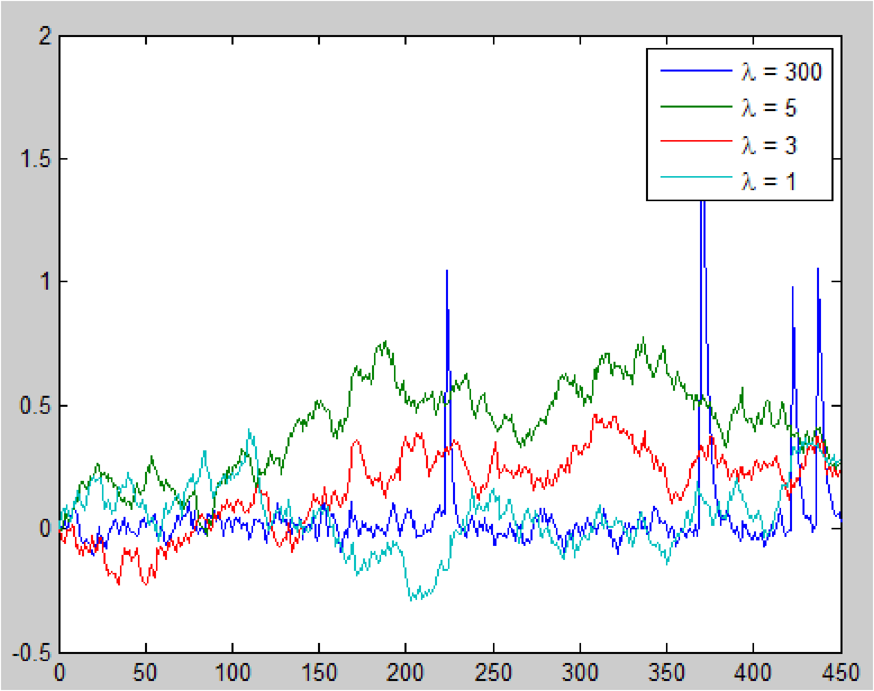

Figure 7.17: Sample paths of the stochastic processes

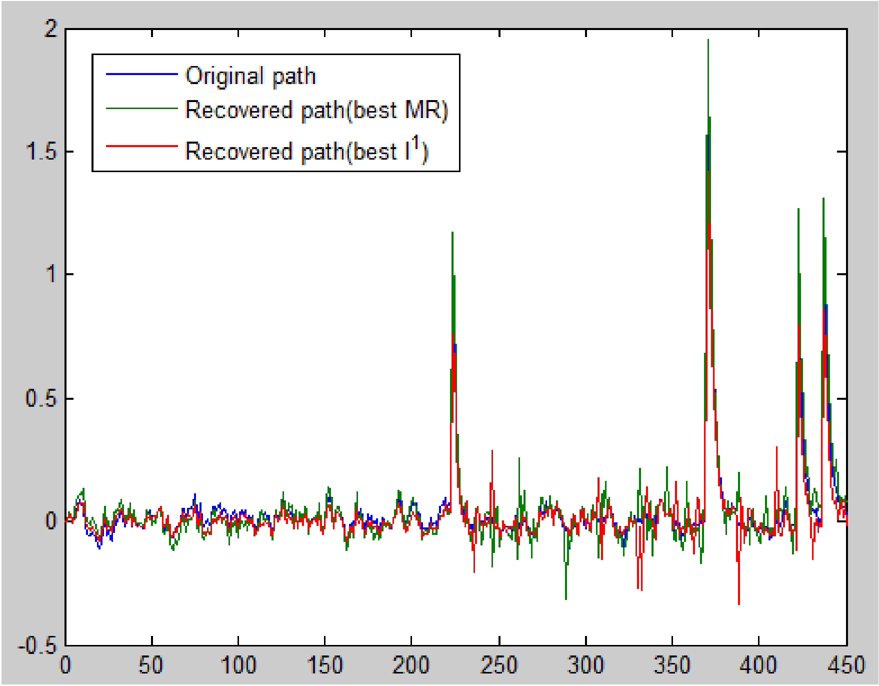

Figure 7.18: Recovered paths with $k = 14$ and $n = 450$. The estimated mean reversion coefficients of recovered paths are close to the true parameter 300. The estimated mean reversion coefficient of original path is 294.9149.

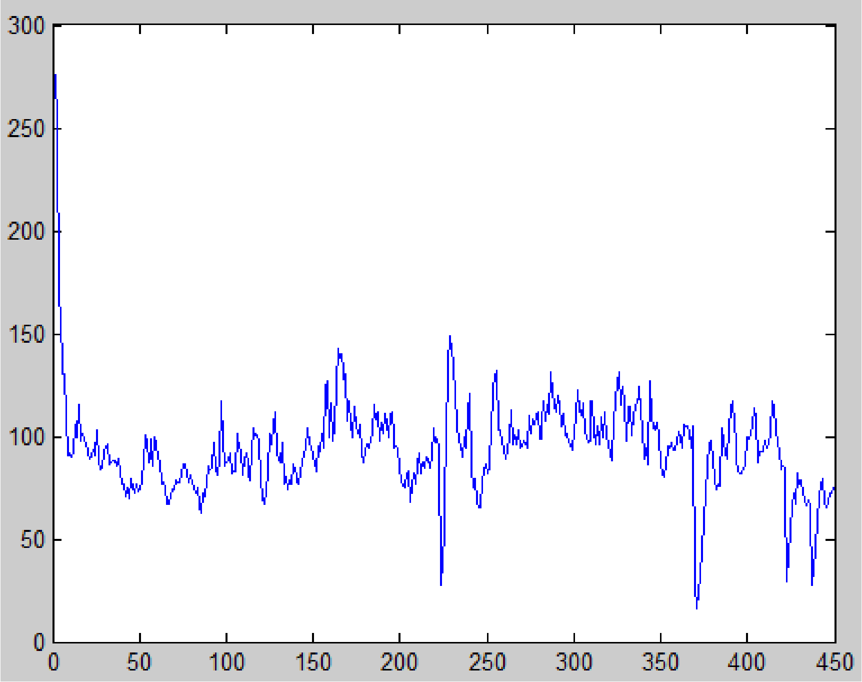

Figure 7.19: The l^1 norms of recovered paths for different i. i is from 2 to 450.

Chapter 8

Conclusion

In this work, we developed a new proxy of mean reversion coefficient: direct OU estimator. From numerical tests, the portfolios constructed under this proxy perform better in convergence trading than the portfolios constructed under predictability.

We developed several different types of optimization problems for building sparse mean reverting portfolios. All these problems do not predetermine the cardinality of the portfolio and we believe this is more realistic.

Without any prior knowledge of the assets, we used the ratio of l_1 and l_2 norms to enforce the sparsity. We studied the properties of the ratio of l_1 and l_2 norms and designed an algorithm in solving the penalized optimization problem.

With our prior knowledge of the assets, we only need the l_1 norm to enforce the sparsity and we found a way to simplify the problem to a non-convex quadratic program. We presented analysis on obtaining global minimizer and developed various algorithms for computing.

In our numerical tests, we applied our methods on both historical market data and simulated data. We compared the computation costs of different algorithms. Our numerical tests suggest that the combination of the least angle regression and the difference of convex functions algorithm is the best choice. We carried out efficient computation for portfolios with hundreds of assets.

We presented the in-sample and out-of-sample performance of the portfolios constructed under different algorithms and different problem settings. Normally, the in-sample performance is better than the out-of-sample performance, but the portfolios constructed under our methods can still maintain a high performance in the out-of-sample period. As the size of dimension increases, it becomes more and more difficult in maintaining the mean reversion coefficients. However, the trading opportunities can keep around 40% of the in-sample counts. We believe future work can be done in increasing the out-of-sample performance. The trading opportunities on in-sample and out-of-sample data share similar trends.

We formulated a new type of problems for recovering fastest mean reverting process. It is a generalization of recovering sparse element in a subspace. From the numerical tests, we successfully recovered the hidden fastest mean reverting OU process.

Bibliography

[1] George EP Box and George C Tiao. A canonical analysis of multiple time series. *Biometrika*, 64(2):355–365, 1977.

[2] Shui-Nee Chow, Tzi-Sheng Yang, and Haomin Zhou. Global optimizations by intermittent diffusion. *preprint*, 2013.

[3] Marco Cuturi and Alexandre D'aspremont. Mean reversion with a variance threshold. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 271–279, 2013.

[4] Alexandre d'Aspremont. Identifying small mean-reverting portfolios. *Quantitative Finance*, 11(3):351–364, 2011.

[5] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[6] Robert F Engle and Clive WJ Granger. Co-integration and error correction: representation, estimation, and testing. *Econometrica: journal of the Econometric Society*, pages 251–276, 1987.

[7] Ernie Esser, Yifei Lou, and Jack Xin. A method for finding structured sparse solutions to nonnegative least squares problems with applications. *SIAM Journal on Imaging Sciences*, 6(4):2010–2046, 2013.

[8] Norbert Fogarasi and János Levendovszky. Improved parameter estimation and simple trading algorithm for sparse, mean reverting portfolios. In *Annales Univ. Sci. Budapest., Sect. Comp*, volume 37, pages 121–144, 2012.

[9] Paul Hand and Laurent Demanet. Recovering the sparsest element in a subspace. *arXiv preprint arXiv:1310.1654*, 2013.

[10] R Horst and Nguyen V Thoai. Dc programming: overview. *Journal of Optimization Theory and Applications*, 103(1):1–43, 1999.

[11] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3-4):321–377, 1936.

[12] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

[13] Hui Ji, Jia Li, Zuowei Shen, and Kang Wang. Image deconvolution using a characterization of sharp images in wavelet domain. *Applied and Computational Harmonic Analysis*, 32(2):295–304, 2012.

[14] Dilip Krishnan, Terence Tay, and Rob Fergus. Blind deconvolution using a normalized sparsity measure. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 233–240. IEEE, 2011.

[15] An Le Thi Hoai and Pham Dinh Tao. Solving a class of linearly constrained indefinite quadratic problems by dc algorithms. *Journal of Global Optimization*, 11(3):253–285, 1997.

[16] Marcos Lopez de Prado and David Leinweber. Advances in cointegration and subset correlation hedging methods. *Journal of Investment Strategies (Risk Journals)*, 1(2):67–115, 2012.

[17] Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[18] Thanh T Ngo, Mohammed Bellalij, and Yousef Saad. The trace ratio optimization problem. *SIAM Review*, 54(3):545–569, 2012.

[19] Pham Dinh Tao and Le Thi Hoai An. Convex analysis approach to dc programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289–355, 1997.

[20] Pham Dinh Tao and Le Thi Hoai An. A dc optimization algorithm for solving the trust-region subproblem. *SIAM Journal on Optimization*, 8(2):476–505, 1998.

[21] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[22] Penghang Yin, Ernie Esser, and Jack Xin. Ratio and difference of l1 and l2 norms and sparse representation with coherent dictionaries. *CAM report 13-21, UCLA, 2013.*

APPENDICES

.1 More Numerical Results of the Methods in Chapter 5

Method	A	B	C	D
1Y	-0.478	-0.479	-0.481	-0.542
2Y	0.000	-0.004	0.000	0.000
3Y	0.000	0.002	0.000	0.000
4Y	1.000	1.000	1.000	1.000
5Y	0.000	0.002	0.000	0.000
7Y	0.000	0.004	0.000	0.000
10Y	0.022	0.015	0.019	0.000
30Y	0.000	0.002	0.000	0.000
λ	1.546	1.548	1.551	1.622
Time(s)	2.669	22.632	7.976	0.161

Table 1: Solutions of method A, B, C and D obtained under $\|x\|_1 \leq 1.5$

Method	A	B	C	D
1Y	0.000	-0.005	-0.033	0.000
2Y	-0.060	-0.050	0.000	0.431
3Y	-0.740	-0.746	-0.766	-1.358
4Y	1.000	1.000	1.000	1.000
5Y	0.000	0.000	0.000	0.000
7Y	0.000	0.000	0.000	0.000
10Y	0.000	0.000	0.000	0.000
30Y	0.000	0.000	-0.001	0.000
λ	1.631	1.636	1.664	5.826
Time(s)	2.340	21.656	7.772	0.124

Table 2: Solutions of method A, B, C and D obtained under $\|x\|_1 \leq 1.8$

Method	A	B	C	D
1Y	0.022	0.022	0.022	0.019
2Y	0.000	0.000	0.000	0.000
3Y	-0.447	-0.447	-0.447	-0.403
4Y	1.000	1.000	1.000	1.000
5Y	-0.603	-0.603	-0.603	-0.690
7Y	0.000	0.000	0.000	0.000
10Y	0.008	0.008	0.008	0.075
30Y	0.019	0.019	0.019	0.000
λ	186.939	186.939	186.939	207.122
Time(s)	0.251	1.222	0.473	0.011

Table 3: Solutions of method A, B, C and D obtained under $\|x\|_1 \leq 2.1$

.2 Estimation of the VAR(1) model

In [8], the authors discussed several methods in estimating β and Γ. In most cases, the number of the observations of assets values M is greater than the number of assets N. Under this case and previous assumptions, we could use the following estimates:

$$\hat{\beta} = (\sum_{t=2}^{M}(S_{t-1} - \bar{S}_1)^T(S_{t-1} - \bar{S}_1))^{-1}(\sum_{t=2}^{M}(S_{t-1} - \bar{S}_2)^T(S_t - \bar{S}_2))$$

$$\hat{\Gamma}_1 = \frac{1}{M-1}\sum_{t=1}^{M}(S_t - \bar{S})^T(S_t - \bar{S})$$

Method	A	B	C	D
1Y	-0.002	-0.002	-0.002	-0.003
2Y	0.118	0.118	0.118	0.119
3Y	-0.580	-0.580	-0.580	-0.559
4Y	1.000	1.000	1.000	1.000
5Y	-0.610	-0.610	-0.610	-0.716
7Y	0.041	0.041	0.041	0.175
10Y	0.041	0.041	0.041	0.000
30Y	-0.009	-0.009	-0.009	-0.018
λ	227.349	227.349	227.349	229.519
Time(s)	0.251	0.652	0.484	0.012

Table 4: Solutions of method A, B, C and D obtained under $\|x\|_1 \leq 2.4$

Method	A	B	C	D
1Y	-0.003	-0.003	-0.004	-0.003
2Y	0.121	0.121	0.123	0.121
3Y	-0.561	-0.561	-0.565	-0.562
4Y	1.000	1.000	1.000	1.000
5Y	-0.724	-0.724	-0.717	-0.719
7Y	0.191	0.191	0.185	0.184
10Y	-0.009	-0.009	-0.008	-0.006
30Y	-0.017	-0.017	-0.017	-0.017
λ	229.426	229.426	229.493	229.486
Time(s)	0.283	0.309	5.784	0.012

Table 5: Solutions of method A, B, C and D obtained under $\|x\|_1 \leq 2.7$

where

$$\bar{S} = \frac{1}{M}\sum_{t=1}^{M} S_t \qquad \bar{S}_1 = \frac{1}{M-1}\sum_{t=1}^{M-1} S_t \qquad \bar{S}_2 = \frac{1}{M-1}\sum_{t=2}^{M} S_t$$

Therefore, the matrices in problem 3.1 can be estimated as:

$$\hat{A} = \hat{\beta}^T \hat{\Gamma} \hat{\beta}, \qquad \hat{B} = \hat{\Gamma}$$

In [8], the authors also pointed out that by numerically solving a Lyapunov equation we could get another estimation of Γ which we will call it $\hat{\Gamma}_2$. They used the quantity $\|\hat{\Gamma}_1 - \hat{\Gamma}_2\|$ to measure the goodness of model fit. The matrix norm here is the largest singular value.