Noname manuscript No.
(will be inserted by the editor)

Three l_1 based nonconvex methods in constructing sparse mean reverting portfolios

Dedicated to the memory of our good friend and colleague Ernie Esser

Xiaolong Long · Knut Solna · Jack Xin

the date of receipt and acceptance should be inserted later

Abstract We study the problem of constructing sparse and fast mean reverting portfolios. The problem is motivated by convergence trading and formulated as a generalized eigenvalue problem with a cardinality constraint [6]. We use a proxy of mean reversion coefficient, the direct Ornstein-Uhlenbeck (OU) estimator, which can be applied to both stationary and nonstationary data. In addition, we introduce three different methods to enforce the sparsity of the solutions. One method uses the ratio of l_1 and l_2 norms and the other two use l_1 norm. We analyze various formulations of the resulting non-convex optimization problems and develop efficient algorithms to solve them for 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 synthetic and historical real market data. In particular, the l_1 regularization method, in combination with quadratic program formulation as well as differ-

This work was partially supported by NSF grants DMS-1222507, DMS-1522383, and IIS-1632935.

Xiaolong Long
8700 Pershing Dr., Playa Del Rey, CA 90293
Tel.: +1-814-321-6383
E-mail: longxiaolongbnu@gmail.com

Knut Solna
Department of Mathematics, University of California at Irvine, Irvine, CA 92697
Tel.: +1-949-824-3154
Fax: +1-949-824-7993
E-mail: ksolna@math.uci.edu

Jack Xin
Department of Mathematics, University of California at Irvine, Irvine, CA 92697
Tel.: +1-949-824-5309
Fax: +1-949-824-7993
E-mail: jxin@math.uci.edu

ence of convex functions and least angle regression treatment, gives fast and robust performance on large out-of-sample data set.

To appear in Journal of Scientific Computing (JOMP).

Keywords Mean reversion · sparse estimation · convergence trading · l_1 and l_2 norms

Mathematics Subject Classification (2000) 62P05 · 90C20 · 90C26

1 Introduction

Convergence trade is a trade designed to benefit from the phenomenon that the price of a portfolio may fluctuate around a certain level. Since the deviations from this level are temporary, investors can build an appropriate trading strategy when they observe the deviations and expect to profit by the amount of convergence. Ideally, convergence trading will be market-neutral and investors will always make profits if the convergence happens. However, the risk of convergence trade is that the expected convergence does not happen, or that it takes too long so that it possibly diverges before converging. Therefore, it is important to quantify how fast the portfolio will converge and how an optimal portfolio could be constructed based on this criterion. In addition, a sparse portfolio may be preferred in convergence trade since sparsity typically means less transaction costs. However, there will be a trade-off between the sparsity and the convergence rate. This makes constructing such a fast mean reverting portfolio from a set of assets a challenging problem.

Classical methods include cointegration [9] and canonical correlation analysis [13], but researchers did not consider sparsity constraints when they applied these theories. A new optimization framework for constructing sparse mean reverting portfolios is proposed in [6]. The author uses the idea of predictability [2] as a proxy for the rate of mean reversion. By presetting the desired cardinality, an optimization problem is formulated and it is essentially a sparse generalized eigenvalue problem. In [5], the authors replace the cardinality constraint by the variance constraint in order to improve profits during arbitrage opportunities, while in [11], the authors discuss more details about parameter estimation and trading strategies. In [30], the authors propose a method called DC-PCA which solves for sparse generalized eigenvectors using the difference of convex algorithms. However, the authors only test their algorithm for principal component analysis.

Based on [6], the sparse optimization problem can be formulated as

$$\begin{array}{ll} \min_x & x^T A x / x^T B x \\ s.t. & \|x\|_0 \leq k \\ & \|x\|_2 = 1, \end{array} \tag{1}$$

where $\|x\|_0$ is the number of the non-zero entries of x and A and B are both positive definite matrices.

Without the cardinality constraint, the problem is a generalized eigenvalue problem. The goal is to find the unit vector x, such that

$$Bx = \lambda Ax$$

where λ has the largest possible value. When the matrix A is the identity matrix, then the problem reduces to finding the first principal component. Sparse principal component analysis problems are well-studied, see for instance [7,17,18,29,34]. In [29], the authors discuss 8 different formulations of the sparse PCA problem and introduce alternating maximization (AM) method to solve them efficiently.

In our work, we take the approach set forth in [6] as a starting point and develop three optimization methods that use l_1 based norms to enforce the sparsity of the portfolio.

The first method extends the AM method to the generalized eigenvalue problem. We use l_1 constraint to enforce sparsity. The problem formulation is

$$\begin{array}{ll} \max_x & x^T Bx \\ s.t. & \|x\|_1 \leq \sqrt{s} \\ & x^T Ax \leq 1, \end{array}$$

The second 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 [10,14,16,19,32]. For example, the non-negative least squares (NNLS) problem under such penalty takes the following form [10]:

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

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:

$$\min_{x \neq 0} \frac{x^T Ax}{x^T Bx} + \gamma \frac{\|x\|_1}{\|x\|_2} \tag{2}$$

where A and B are positive definite matrices and γ is a nonnegative tuning parameter. The first term of the objective function of (2) models predictability and the second term promotes sparsity. The details are in section 4. The l_1 norm regularization technique for variational problems under non-convex (orthogonality/unit ball) constraints has been actively developed lately to construct compactly supported eigen-functions [1,20,25,26]. In fact (2) is simply imposing l_1 regularization on the unit ball in l_2.

Due to the non-convexity of (2), finding a global minimum is challenging. To deal with this aspect of global optimization, we shall incorporate a recent variant of simulated annealing, the so called intermittent diffusion method with discontinuous diffusion coefficient [4]. The combined local minimization of (2) and random search for global minimum is however expensive for large size portfolio computation.

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

$$f(r) = \max_{x_i=1, \|x\|_1 \leq m} x^T Bx - rx^T Ax \tag{3}$$

where A and B are both positive definite matrices, r is a nonnegative number, x_i is the ith entry of the vector x, and the choice of (i, m) will be addressed later. The ratio minimization problem:

$$\min_{x_i=1, \|x\|_1 \leq m} \frac{x^T Ax}{x^T Bx}$$

will be shown to be equivalent to finding the positive root of $f(r)$.

Non-convexity still exists and a special algorithm designed to minimize a difference of convex functions [12] can be applied to overcome this difficulty in combination with convex algorithms such as the least angle regression [8]. The resulting algorithm is faster than the above methods and can handle portfolios with hundreds of assets. The combined difference of convex function and least angle regression method for root finding of $f(r)$ is our main contribution for computing large size (over 100's assets) portfolios with a significant speedup over other non-convex methods.

The paper is organized as follows. As background, we restate the problem of sparse and fast mean reverting portfolio proposed in [6] (section 2.1) and discuss two proxies of the mean reversion coefficient (section 2.2). One of them is the direct OU estimator which we found to work better. In section 3, we present the optimization problem in finding the desired portfolios and give a brief introduction to existing methods for solving it. In sections 4, 5 and 6, we introduce three new approaches in constructing portfolios. We formulate new optimization problems and develop the algorithms to solve them. Numerical experiments are presented in section 7. We perform in-sample tests and out-of-sample tests on both synthetic data and historical market data.

Before we introduce the portfolio problem, let us summarize some notations.

- $\langle x, y \rangle = x^T y$;
- For $x \in \mathbb{R}$, $(x)_+ = \max(x, 0)$;
- $\chi_A(\mathbf{x}) = \begin{cases} 0 & : \mathbf{x} \in A \\ \infty & : \text{else} \end{cases}$;

2 Background

2.1 Sparse and fast mean reverting portfolio problem

We briefly review the formulation in [6]. Suppose that S_{ti} is the value at time t of the ith asset with $i = 1, 2..., n$ and $t = 1, 2, ..., l$, 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 coefficient λ of P_t by adjusting the portfolio weights x_i under the normalization $\sum_{i=1}^{n} x_i^2 = 1$. In addition, we want to limit the number of assets in the portfolio, i.e. we want to balance the number of nonzeros of x_i's and the mean reversion coefficient.

2.2 Proxy of the mean reversion coefficients

In [5], the authors discuss three different criteria to measure how fast a portfolio is mean-reverting. They are predictability, the portmanteau statistic and the crossing statistic. In our paper, we mainly consider two proxies. One is predictability and the other one is called the direct OU estimator.

2.2.1 Predictability

The idea of predictability of a time series is first derived in [2]. They consider a stationary vector autoregressive (VAR) model:

$$S_t^T = S_{t-1}^T \beta + Z_t^T \tag{4}$$

where the column vector $S_t = (S_{t1}, ..., S_{tn})^T$, $\beta \in \mathbb{R}^{n \times n}$, Z_t is a vector of i.i.d Gaussian noise with zero mean and a covariance matrix Σ, independent of S_{t-1}.

In the 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:

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

In [6], d'Aspremont proposed to use this measure of predictability as a proxy for the mean reversion parameter λ in an Ornstein-Uhlenbeck process.

In the multivariate case, consider a portfolio $P_t = S_t^T x$ with weights $x \in \mathbb{R}^n$, then by multiplying both sides of (4) by x, we get

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

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. Minimizing the predictability $\nu_1(x)$ corresponds to maximizing λ.

2.2.2 Direct OU estimator

The 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, we will need an estimator for the mean reversion coefficient and will use this estimator as its proxy value.

Consider an Ornstein-Uhlenbeck process:

$$dP_t = \lambda(\mu - P_t)dt + \sigma dW_t \tag{5}$$

We can estimate the parameters of an OU process by linear regression. We refer the readers to [15,33] for more details. By writing (5) in a discrete form, we have

$$P_t - P_{t-\Delta t} = \lambda\mu\Delta t - \lambda P_{t-\Delta t}\Delta t + \sigma\Delta W_t$$

Note that it can be written as:

$$P_t = \lambda\mu\Delta t + P_{t-\Delta t}(1 - \lambda\Delta t) + \sigma\Delta W_t$$

It is equivalent to the linear regression:

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

Therefore, we could estimate all the parameters by regressing $P_t - P_{t-\Delta t}$ on $P_{t-\Delta t}$. 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 $\frac{\hat{\sigma}(\epsilon_t)}{\sqrt{\Delta t}}$.

Maximizing the estimated mean reversion coefficient λ corresponds to minimizing the estimated slope of $\hat{b}$. Note that

$$b = \frac{\mathrm{cov}(P_t - P_{t-\Delta t}, P_{t-\Delta t})}{\mathrm{var}(P_{t-\Delta t})} = \frac{\mathrm{cov}(P_t, P_{t-\Delta t})}{\mathrm{var}(P_{t-\Delta t})} - 1$$

Replacing P_t by $S_t^T x$, we have:

$$b = \frac{\mathrm{cov}(S_t^T x, S_{t-1}^T x)}{\mathrm{var}(S_{t-1}^T x)} - 1$$

We define the direct OU estimator as:

$$\nu_2(x) = \frac{\mathrm{cov}(S_t^T x, S_{t-\Delta t}^T x)}{\mathrm{var}(S_{t-\Delta t}^T x)} \tag{6}$$

If S_t is stationary, then we can rewrite (6) in the following form:

$$\nu_2(x) = \frac{x^T(\text{cov}(S_t, S_{t-\Delta t}) + \text{cov}(S_t, S_{t-\Delta t})^T)x}{2x^T \text{var}(S_{t-\Delta t})x}$$

Minimizing the predictability $\nu_2(x)$ corresponds to maximizing λ.

This proxy is a special case of the crossing statistics in [5] when $p = 1$. The special assumption needed is that the linear combinations of the asset $S_t^T x$ is stationary. This phenomenon is called cointegration [9]. According to our numerical tests, the portfolios constructed under this proxy gives better trading performance.

3 Sparse optimization problem and existing methods

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

$$\frac{x^T A x}{x^T B x}$$

where A and B are $n \times n$ positive definite matrices and we will make this assumption for the rest of the paper. The estimation procedures for A and B for two proxies are discussed in the Appendix B.

If we do not penalize the cardinality of x, then minimizing these proxies is the same as a generalized eigenvalue problem. In order to obtain a sparse solution, d'Aspremont in [6] proposed the following sparse optimization problem:

$$\begin{array}{ll} \min_x & x^T A x / x^T B x \\ s.t. & \|x\|_0 \leq k \\ & \|x\|_2 = 1, \end{array} \tag{7}$$

where the l_0 norm of a vector x is the number of nonzero entries of x. This problem has been proved to be NP-hard [23]. When the dimension of the problem is large, we cannot expect to find the optimal solution. Several methods for solving (7) have been proposed in [6,11,30]. We give a brief summary here:

- 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 yields 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 from $\{1, 2, ..., n\} \setminus I_k$ such that it has smallest objective among all the other choices. Then we add it to I_k and repeat this procedure 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 $|x_{opt}| = (|x_1|, ..., |x_n|)^T$. Next we solve the generalized eigenvalue problem on the set J_k by taking the corresponding part of matrices A and B. It is the fastest among all the listed methods, but it sacrifices performance [11].
- Semidefinite relaxation method: this method reformulates the problem (7) as a semidefinite program. This can be considered as an extension of DSPCA in [34]. We found that this relaxation method can be considered as replacing the $\|x\|_0 \leq k$ by $\|x\|_1/\|x\|_2 \leq \sqrt{k}$. For more details, we refer the readers to [6] and the Appendix A. The major drawback of this algorithm is that it is not scalable to high-dimensional datasets [30].
- DC-PCA: the problem (7) is formulated as difference of convex functions problem and the authors apply the difference of convex functions algorithm(DCA) to efficiently solve the optimization problem. In [30], this method was proven to be efficient for solving various PCA problems, but the authors do not apply it for the generalized eigenvalue problems.

In the next three sections, we will present three approaches for approximating the optimal solution to the problem (7).

4 Optimization problems based with l_1 constraint (l_1 DC-PCA)

In this section, we consider using the l_1 constraint in the sparse generalized eigenvalue problem. This can be considered as an extension of the l_1 constrained sparse principal component analysis problem. The problem is formulated as

$$\begin{aligned} &\max_x \ x^T B x \\ &s.t. \quad \|x\|_1 \leq \sqrt{s} \\ &\qquad x^T A x \leq 1, \end{aligned} \tag{8}$$

One way to tackle this problem is to treat the problem as a difference of convex functions and using the DC (difference of convex functions) algorithm. DC programming is extensively studied in [12, 22, 27, 28]. Here we offer the algorithm in raw form in algorithm 1:

Algorithm 1 DC algorithm

Choose $x_0 \in \mathbb{R}^n$;
repeat
  choose $y_k \in \partial h(x_k)$
  choose $x_{k+1} \in \partial g^*(y_k)$
until convergence

g^* denotes the conjugate function of g and ∂g denotes the subgradient of g. For the detailed definition, we refer to [12]. The method aims to minimize a function $g - h$ where both g and h are convex functions on the whole space.

In most of the literature on DC programming, an algorithm to find a local optimum is used. In practice, the local algorithm often approximates well a global minimum.

For (8), we have $g(x) = \chi_F(x)$ and $h(x) = -x^T Bx$, where $F = \{x : x^T Ax \leq 1, \|x\|_1 < \sqrt{s}\}$.

Therefore, we will update x by solving the following problem by the DC algorithm:

$$\begin{array}{l} x^{(k)} = \arg\max_x \langle 2Bx^{(k-1)}, x\rangle \\ s.t.\ \ \|x\|_1 \leq \sqrt{s} \\ \qquad x^T Ax \leq 1, \end{array} \tag{9}$$

where $x^{(k)}$ is the x in the kth interation.

In fact, we could also obtain the same formulation by the generalized power iteration [18] and alternating maximization [29]. When the A is the identity matrix, then we reduce to a sparse PCA problem. The explicit solution is discussed in [18,29]. For a general A, the maximizer of (9) is then

$$x^* = \frac{A^{-1}(v-z)}{\sqrt{(v-z)^T A^{-1}(v-z)}}$$

where $v = 2Bx^{(k-1)}$ and z is the optimizer of the dual problem of (9):

$$\min_{\lambda \geq 0, \|z\|_\infty \leq \lambda} \lambda\sqrt{s} + \sqrt{(v-z)^T A^{-1}(v-z)}$$

When $A = (A_{ij})$ is a diagonal matrix, then the above reduces to a closed-form expression:

$$x_i^* = \frac{A_{ii}^{-1}\mathbf{sgn}(v_i)(|v_i| - \lambda)_+}{\sqrt{\sum_i A_{ii}^{-1}(|v_i| - \lambda)_+^2}}$$

This method (l_1 DC-PCA)is similar to the DC-PCA in [30] which deals with an l_0 penalized optimization problem. The authors approximate the $\|x\|_0$ by $\sum_i \log(|x_i| + \epsilon)$ and formulate

$$\begin{array}{l} \max_x\ x^T Bx - \rho \sum_i \log(|x_i| + \epsilon) \\ s.t.\quad x^T Ax \leq 1, \end{array} \tag{10}$$

After reformulating the object function with an indicator function and applying the DC algorithm (see details in [30]), we will need to solve the following problem at each iteration:

$$\begin{array}{l} \max_x\ \langle D(x^{(k-1)})Bx^{(k-1)}, x\rangle - \frac{\rho}{2}\|x\|_1 \\ s.t.\quad x^T D(x^{(k-1)})AD(x^{(k-1)})x \leq 1, \end{array} \tag{11}$$

where $D(x)$ is the diagonal matrix whose diagonal entries $D_{ii} = x_i$. The next $x^{(k)}$ follows $x_i^{(k)} = x_i^{(k-1)}x_i$. One thing to note is that if $x_i^{(k-1)} = 0$ then $x_i^{(k)} = 0$. Therefore, the initial $x^{(0)}$ has to be a non-zero vector in the space $\{x : x^T Ax \leq 1\}$.

When $A = (A_{ij})$ is a diagonal matrix, then we also have an explicit solution to the problem (11):

$$x_i^* = \begin{cases} \frac{(c_i)^{-1}\mathbf{sgn}(v_i)(|v_i|-\rho/2)_+}{\sqrt{\sum_i c_i^{-1}(|v_i|-\rho/2)_+^2}} & x_i^{(k-1)} \neq 0 \\ 0 & x_i^{(k-1)} = 0 \end{cases}$$

where $c_i = A_{ii}(x_i^{(k-1)})^2$ and $v = D(x^{(k-1)})Bx^{(k-1)}$.

For non-diagnoal matrix A, the algorithm is a sequence of QCQPs with a multiplicative update. More details can be found in [30].

These two algorithms are very efficient if the matrix A is a diagonal matrix due to the existence of the explicit solution. When A is not a diagonal matrix, we have not found them efficient to handle a large size problem. Since these two DC-PCA algorithms directly apply DC algorithm, we refer to [21] for the convergence analysis.

5 Optimization problems based on the ratio of l_1 and l_2 norms (l_1/l_2 regularization)

5.1 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 (2) is scale invariant. The scale invariant 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 [32].

In the following sections, we will present two formulations of sparse mean reverting problems (7) and discuss the algorithms to solve them. We will also show that the ratio of l_1 and l_2 norms can enforce the sparsity.

5.2 Two formulations

We could reformulate the problem (7) in the following way:

$$\begin{aligned} &\min_x \ x^TAx/x^TBx \\ &s.t. \ \ \tfrac{\|x\|_1}{\|x\|_2} \leq m \\ &\quad\ \ x \neq 0, \end{aligned} \tag{12}$$

This problem can be reformulated as the semidefinite programming problem in [6]. The details can be found in Appendix A.

In addition, we could also modify the problem (7) in the following way:

$$\min_{x\neq 0} \frac{x^TAx}{x^TBx} + \gamma\frac{\|x\|_1}{\|x\|_2} \tag{13}$$

where γ is a regularizer.

Comparing with the problem (7), the problem (13) does not specify the cardinality beforehand. In this way, we can consider the l_1 norm in the numerator as a way to quantify the uniform transaction costs.

5.3 Algorithm to solve (13)

The major difficulty comes from the non-convexity of the problem. Most optimization algorithms will only provide a local minimizer. Since the performance of local minimizers may vary a lot, it is important to develop an algorithm to approximate the global minimizer. The intermittent diffusion(ID) algorithm proposed in [4] 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." For more theoretical analysis of the ID algorithm, we refer to [4].

Algorithm 2 Solve (13)

Input A, B and γ;
Let α be the scale of diffusion strength, κ be the scale for diffusion time and N be the total number of realizations.
Set the initial state x_0 as the minimizer of $\frac{x^T Ax}{x^T Bx}$ 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 (13) 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 (13) and record the final state $x_T := x(T, \omega)$. The sub-gradient of $\|x\|_1$ is used to calculate the gradient of (13).
  Find a local minimizer $\hat{x}_i$ of problem (13) by line search algorithm with starting point x_T.
  $X_{opt} = \hat{x}_i$ if $f(\hat{x}_i) < f(X_{opt})$.
end for

5.4 Analysis of the ratio of l_1 and l_2 penalty

In this section, we want to prove some useful properties of the solutions to the problem (13). Theorem 1 indicates that an almost 1-sparse solution can always be recovered if the value of γ is large enough.

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

$$f(x,\gamma) = \frac{x^T Ax}{x^T Bx} + \gamma \frac{\|x\|_1}{\|x\|_2}$$

$$L(x) = \frac{x^T Ax}{x^T Bx}$$

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

We denote the set of minimizers of the problem (13) by $X(\gamma)$ and we will set their l_2 norm to 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 $x^T Ax/x^T Bx$ 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 and l_2 norms. Therefore, the set of the optimizers, $x(\gamma)$, has a unique value of the function $P(\cdot)$ on this set (and then a unique value of $L(\cdot)$). Mathematically, it can be defined in the following way:

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

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

Under this definition, $P(x(\gamma))$, $L(x(\gamma))$ and $f(x(\gamma),\gamma)$ are well-defined, since $P(\cdot)$, $L(\cdot)$ and $f(\cdot,\gamma)$ are constant on $x(\gamma)$.

In the following proof, we will use the notations:

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

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

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

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

Theorem 1 *Denote $x(\gamma)$ as the set of optimal solutions of problem (13) given γ. Given $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$ both $n \times n$ positive-definite matrices, then for any $\epsilon > 0$, there exists a number $\gamma(\epsilon)$, a vector x and a 1-sparse vector e, such that for any $\gamma \geq \gamma(\epsilon)$, $\|x - e\|_2 < \epsilon$, where $x \in x(\gamma)$, e minimizes $L(x)$ and $\|e\|_2 = 1$.*

Proof First, note that the problem (13) is scale invariant, therefore, we could constrain the problem on the sphere $\|x\|_2 = 1$. All the vectors in our proof have 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 to $S_+ \equiv \{x = (x_1, ..., x_n)^T : x_1 \geq 0 \ \textit{ and } \ \|x\|_2 = 1\}$.

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

Denote $D_i = \{x : \|x - e_i\| < \delta_i\}$ for $i = 2, 3, ..., n$, then for all the points $x \in S_+ \cap (\cup_{i=2}^n D_i)$, $L(x) > L(e_1)$ and $\|x\|_1/\|x\|_2 \geq 1$. Therefore, for all the points $x \in S_+ \cap (\cup_{i=2}^n D_i)$, $f(x, \gamma) > f(e_1, \gamma)$.

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. If we want D_1 to be the only region that can obtain the minimizer of (13), $\gamma(\epsilon)$ must satisfy the following inequality:

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

This is equivalent to

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

And it implies:

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

The previous 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, $\gamma(\epsilon)$ only needs to satisfy:

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

In the case that the minimal ratio is achieved by several indices, the proof is similar by modifying $S_+(\epsilon)$ accordingly.

Assume $\frac{a_{11}}{b_{11}} = ... = \frac{a_{kk}}{b_{kk}} = \min_i\{\frac{a_{ii}}{b_{ii}}\}$. Due to the same reason, $\exists \delta_i$, for any point x in the region $\{x : \|x - e_i\| < \delta_i\} \cap S_+$, $f(x, \gamma) > f(e_1, \gamma)$, for $i = k + 1, ..., n$. For any $\epsilon > 0$, let $D_i = \{x : \|x - e_i\| < \epsilon\}$ for $1 \leq i \leq k$ and $D_i = \{x : \|x - e_i\| < \delta_i\}$ for $k + 1 \leq i \leq n$. We can apply the same logic on $S_+(\epsilon) \equiv S_+ \setminus (\cup_{i=1}^n D_i)$ to get the same conclusion.

6 Optimization problems based on the l_1 norm and prior knowledge (DC-LAR)

6.1 Motivation

In the previous section, we applied a stochastic algorithm in searching for a global optimizer of the problem (13). 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. We also would like to keep the number of tuning parameters as low as possible.

6.2 Formulation

Note that the problem (7) is equivalent to:

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

We use the l_1 norm to enforce the sparsity and consider the following problem:

$$f(r) = \max_{x_i=1, \|x\|_1 \le m} x^T Bx - rx^T Ax \tag{14}$$

where i is a predetermined 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 $x_i = 1$ enables using the l_1 norm to enforce the sparsity and also simplifies the problem to a quadratic program.

Now, we would like to show that

Theorem 2 *Let*

$$f(r) = \max_{x_i=1, \|x\|_1 \le m} x^T Bx - rx^T Ax$$

where A and B are both positive definite matrices, then

a) For any given $R > 0$, (14) is continuous on $0 \le r \le R$;
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:

$$\begin{array}{ll} \max_{x\neq 0} & x^T Bx / x^T Ax \\ s.t. & x_i = 1 \\ & \|x\|_1 \leq m \end{array} \tag{15}$$

Proof In the following proof, we define $g(x, r) = x^T Bx - rx^T Ax$.

a) For any given $R > 0$, since $g(x, r)$ 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 $r_1 > r_2$, $g(x_1, r_1) = f(r_1)$ and $g(x_2, r_2) = f(r_2)$, then we must have

$$\begin{aligned} & f(r_1) - f(r_2) \\ & \leq f(r_1) - g(x_1, r_2) \\ & = (x_1^T Bx_1 - r_1 x_1^T Ax_1) - (x_1^T Bx_1 - r_2 x_1^T Ax_1) \\ & = (r_2 - r_1) x_1^T Ax_1 < 0 \end{aligned}$$

since A is positive definite and $x_1 \neq 0$. 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 u such that

$$r_1 := \frac{u^T Bu}{u^T Au} > r_*$$

Then we must have:

$$\begin{aligned} 0 & = u^T Bu - r_1 u^T Au \\ & = u^T Bu - r_* u^T Au - (r_1 - r_*) u^T Au \\ & \leq x_*^T Bx_* - r_* x_*^T Ax_* - (r_1 - r_*) u^T Au \\ & = -(r_1 - r_*) u^T Au < 0 \end{aligned}$$

Therefore, the result is shown by 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 (15).

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 [24]. Here in Theorem 6.1, we considered the additional l_1 constraint.

6.3 DC-LAR method

First, we present the Algorithm for finding r_*. We used a binary search algorithm.

Algorithm 3 Find r_*

Input the matrices B and A, a threshold $\epsilon > 0$ for stopping, initial $x^{(0)}$, r_{min},r_{max}, i, m and maximum iterations NMAX;
Set $N = 1$;
while $N \leq$ NMAX **do**
  Set $r_0 = \frac{1}{2}(r_{min} + r_{max})$;
  Solve the quadratic program (14) with r_0 and obtain the maximizer $x^{(0)}$;
  if $|x^{(0)})^T Bx^{(0)} - r_0(x^{(0)})^T Ax^{(0)}| < \epsilon$ **OR** $r_{max} - r_{min} < \epsilon$ **then**
    Stop
  end if
  if $(x^{(0)})^T Bx^{(0)} - r_0(x^{(0)})^T Ax^{(0)} > 0$ **then**
    $r_{min} = r_0$
  else
    $r_{max} = r_0$
  end if
end while

The difficult part is solving the quadratic program (14). It is a non-convex optimization problem. Therefore, classical algorithms can not guarantee a global optimizer. In addition, the computational cost is expensive when the dimension of the problem is large(≥ 100). We will again apply the DC algorithm to solve the problem.

In order to apply the DC algorithm, we could first reformulate the problem (14) in the following way by plugging in the constraint $x_i = 1$:

$$\min_{\|x\|_1 \leq m'} rx^T Vx + c^T x - x^T Ux$$

where U and V are the submatrices of B and A without the ith row and column and $x \in \mathbb{R}^{n-1}$.

In addition, by a standard method in DC algorithm, we could change it to an unconstraint problem:

$$\min_x rx^T Vx + c^T x - x^T Ux + \chi_{\|x\|_1 \leq m'}(x) \tag{16}$$

Let

$$g(x) = rx^T Vx + c^T x + \chi_{\|x\|_1 \leq m'}(x) \qquad h(x) = x^T Ux$$

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

Algorithm 4 Solve (16) 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:

$$\inf\{rx^TVx + c^Tx - x^Ty^{(k)} + \chi_{\|x\|_1 \leq m'}(x), x \in \mathbb{R}^{n-1}\} \tag{17}$$

until convergence

The problem (17) can be considered as a quadratic program with l_1 constraint.

$$\begin{array}{l} \min\ rx^TVx + c^Tx + x^Ty^{(k)} \\ s.t.\ \|x\|_1 \leq m' \end{array} \tag{18}$$

It can also be rewritten as a least squares optimization with l_1 constraint. This problem has been well studied in the literature. The major difficulty is handling the l_1 constraint. We apply the least angle regression method to solve this problem.

The idea is that we first reformulate (18) as a LASSO problem [31]. Notice that the objective of a LASSO problem is

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

where $\mathbf{X}$ is a matrix of predictor variables, $\mathbf{Y}$ is an outcome vector and λ is a tuning parameter.

Therefore, by solving the following system for $\mathbf{X}$ and $\mathbf{Y}$, we retrieve a LASSO type problem from (18):

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

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) [8].

7 Numerical tests

Our procedures are implemented in Matlab R2015a. 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 percentage 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 have been in the list since July 2005. After this preselection procedure, we have 458 companies left. We used the stock prices of the first 100 companies according to the alphabetical order of the ticker symbols. The data size in our numerical test is 2000×100.

In the following sections, we performed several tests:

- Scalability and performance tests of DC-PCA and l_1-constraint DC-PCA. Details are in section 7.1.
- In-sample and out-of-sample tests for the performance of the portfolios constructed under two mean reverting proxies: predictability and direct OU estimator on synthetic data. Details are in section 7.2.
- The impact of the size of training set on out-of-sample performance. Details are in section 7.3.
- In-sample and out-of-sample tests for the ratio of l_1 and l_2 norms approach on the U.S. swaps data. Details are in section 7.4.
- In-sample tests for the l_1 norm with prior knowledge approach on the historical daily closing prices of 100 stocks. Details are in section 7.5.1.
- In-sample and out-of-sample tests for the l_1 norm with prior knowledge approach on the U.S. swaps data. Details are in section 7.5.2.
- In-sample and out-of-sample tests for the l_1 norm with prior knowledge approach on synthetic data with 100 assets. Details are in section 7.5.3.
- In-sample and out-of-sample tests using a set of trading rules with risk control measures. Details are in section 7.6.

7.1 Scalability and performance tests of DC-PCA and l_1-constraint DC-PCA

In [30], the numerical tests show that the DC-PCA is preferred over SPCA[7] and DSPCA[34] on sparse PCA problem. The solution of DC-PCA gives larger explained variance and smaller computational costs relatively. Now we want to see how well our proposed method and DC-PCA perform on the sparse generalized eigenvalue problem.

We have performed two tests on synthetic data. The first test is a scalability test. We apply these two method on randomly chosen problems of size n ranging from 200 to 2500 for 6 different values of ρ and s and compute the average time costs. The matrices A and B are both positive definite and A is a diagonal matrix. The second test is to compare the explained variances of two methods. We generate the two positive definite matrices A and B of size 3000×3000 where A is a diagonal matrix. We compute a series of solutions with different sparse parameters ρ and s and compare the explained variances of the solutions with the same numbers of non-zero loadings.

The results are shown in Figure 1. The left plot shows the average CPU time vs. n(the number of rows of the matrix) for these two methods with the empirical complexity growing as $O(n^p)$, where $p = 1.61$ for our proposed l_1 constraint DC-PCA and $p = 2.40$ for DC-PCA. We could see that the l_1 constraint formulation is more efficient than the DC-PCA. The right plot shows the comparative performance (explained variance vs. sparsity) of the

two methods for the first generalized eigenvector. The amount of variance explained by the first generalized eigenvector is 23%. In this case, the two methods are very close and DC-PCA is slightly better than our proposed method.

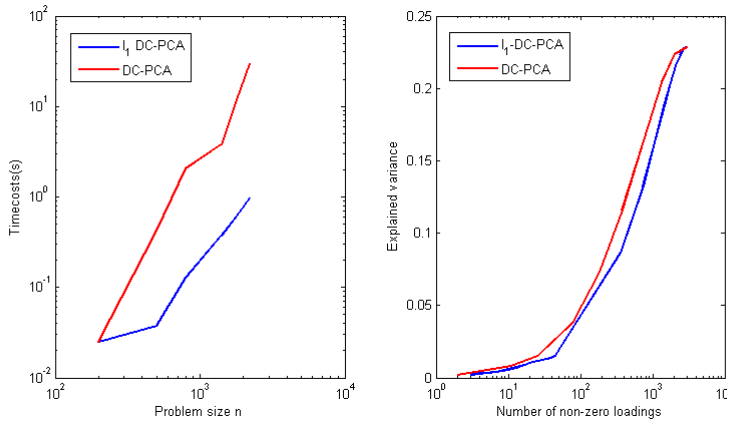

Fig. 1 Comparison between l_1 constraint DC-PCA and DC-PCA in *DC-PCA* in scalability and performance.

From this test, two methods are seen to be very efficient when the matrix A is diagonal. The computation can be done in less than one second even if the problem is of size of 2500.

7.2 Comparison of two mean reverting proxies

Here we will compare the performance of portfolio selection via two proxies that we discussed in section 3.

We set the matrix β and the noise covariance matrix Σ in the VAR(1) model (4) as our estimations based on the U.S. swaps data. We generated a data set of size 350×8, so 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 [11], 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 the ability to short sell. Our objective here is using a simple procedure to evaluate portfolio performance but not building an "optimal" convergence trading strategy. A trading opportunity means the observation that the price converges after out-of-tolerance mispricing. We use P_t to denote the portfolio value at time t.

The trading strategy can be summarized as follows:

- If the observed sample $P_t > \mu + \tau$, close long position if we already hold any. Open a short position if we are not in any position. Otherwise we perform no action.
- If the observed sample $P_t < \mu - \tau$, close short position if we already short any. Open a long position if we are not in any position. Otherwise we perform no action.
- If the observed sample $\mu - \tau \leq P_t \leq \mu + \tau$, close any long or short position. Otherwise we perform no action.

Figure 2 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 $y = \mu - \tau$.

A trading opportunities will occur if we have a high price or low price before returning to the normal range.

The simulation test is similar as before. After we found the optimal solutions to the problem (7) of a given k by the exhaustive search method, we tested the trading strategy on the test set. 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 τ as the sample standard deviation.

The results are shown in Figure 3. The top left plot is the average in-sample mean reversion coefficients versus cardinality. The top right plot is the average out-of-sample mean reversion coefficients versus cardinality. The bottom plot is the average out-of-sample trading opportunities versus cardinality.

We see that the solutions under direct OU estimator usually perform better than that under the predictability. We can also note that the trading opportunities/returns go down somewhat for the portfolios constructed under direct OU estimator when the cardinalities increase from 4 to 8. This could be caused by choice of the parameter τ. A higher τ may increase the profit at each trading opportunity, but decrease the total number of trades. We remark that in [5] the authors use a variance constraint to improve the profits during arbitrage opportunities.

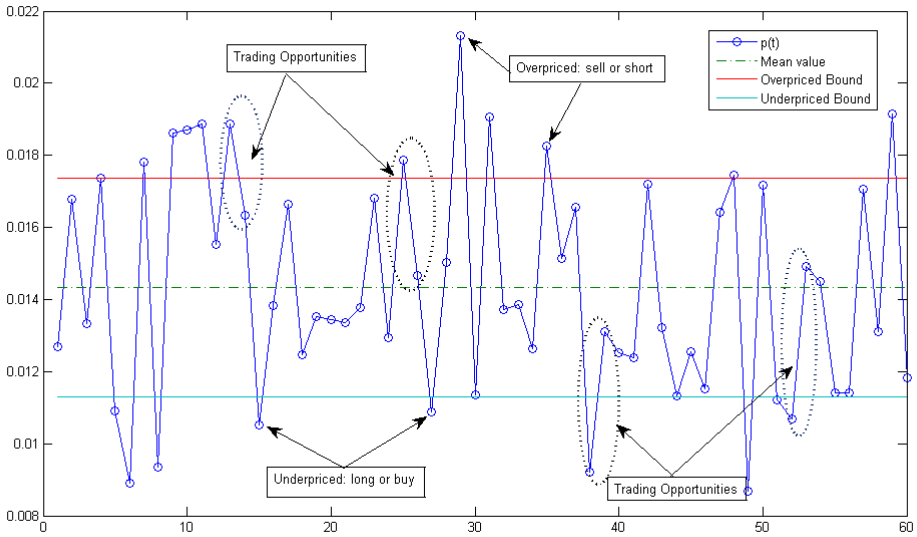

Fig. 2 Trading Opportunities (Not all are marked in the plot.)

7.3 The impact of the size of training set on out-of-sample performance

From our numerical results, we notice that the size of the training set will affect the out-of-sample performance. To illustrate this, we perform the following tests.

After presetting proper β and Σ, we generate a data set of 10 assets for 400 observations based on the VAR(1) model (4). We use the last 50, 100, 200 and 400 observations as the training set and estimate A's and B's in (7) respectively. Next, we construct the portfolios under different cardinality constraints by the exhaustive search method by using those A's and B's.

We then generate the test data sets. Each has 400 observations of 10 assets. They follow the same VAR(1) model. The starting values are the final observations of the training data. In total, we have 100 test data sets and we compare the average out-of-sample mean reversion coefficients of the portfolios constructed based on the training set. If we use the last 50 observations of the training set for estimation, then we will only test those portfolios on the first 50 observations of each test set, and so forth.

Figure 4 shows the results. It illustrates the impact of the size of training set on out-of-sample performance. We compare the in-sample mean reversion coefficients with the average out-of-sample mean reversion coefficients under different cardinality constraints. The portfolios are constructed by solving the problem (7) by the exhaustive search method. The top left uses 50 observations of the training set and 50 observations of the test set; the top right uses 100 observations of the training set and 100 observations of the test set; the

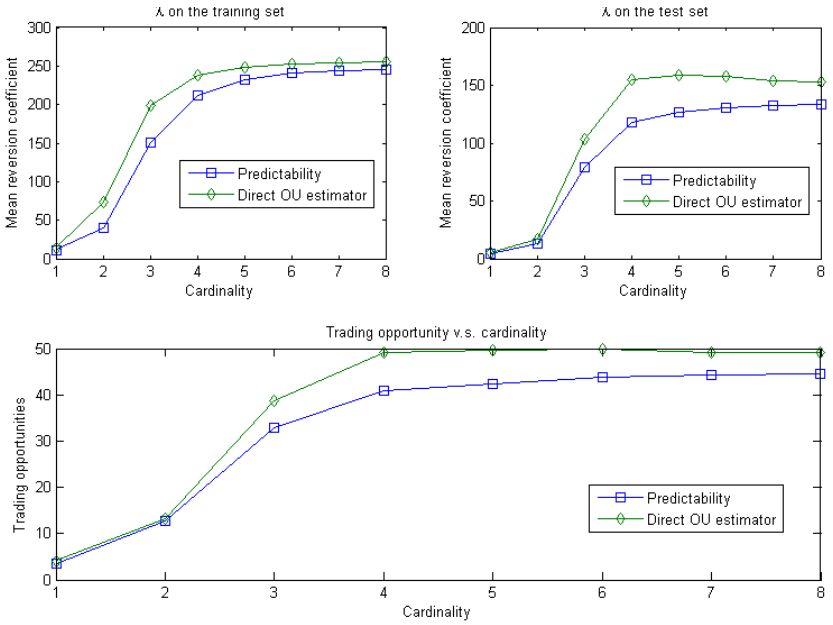

Fig. 3 Comparison of the average performance of sparse mean reverting portfolios solved under different proxies using the exhaustive search method on synthetic data from VAR(1) model (4). Top left: average in-sample mean reversion coefficients versus cardinality; Top right: average out-of-sample mean reversion coefficients versus cardinality; Bottom: average out-of-sample trading opportunities versus cardinality.

bottom left uses 200 observations of the training set and 200 observations of the test set; the bottom right uses 400 observations of the training set and 400 observations of the test set. Notice that when the size of the training set increases, the out-of-sample performance becomes closer to the in-sample performance.

7.4 Tests of the ratio of l_1 and l_2 norms approach

We performed in-sample and out-of-sample tests on the U.S. swaps data and tried to solve the following problem in section 4:

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

by using intermittent diffusion algorithm (ID algorithm) [4]. After each iteration of the line search algorithm [3], we fix the l_2 norm to 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 so that a global minimizer exists in

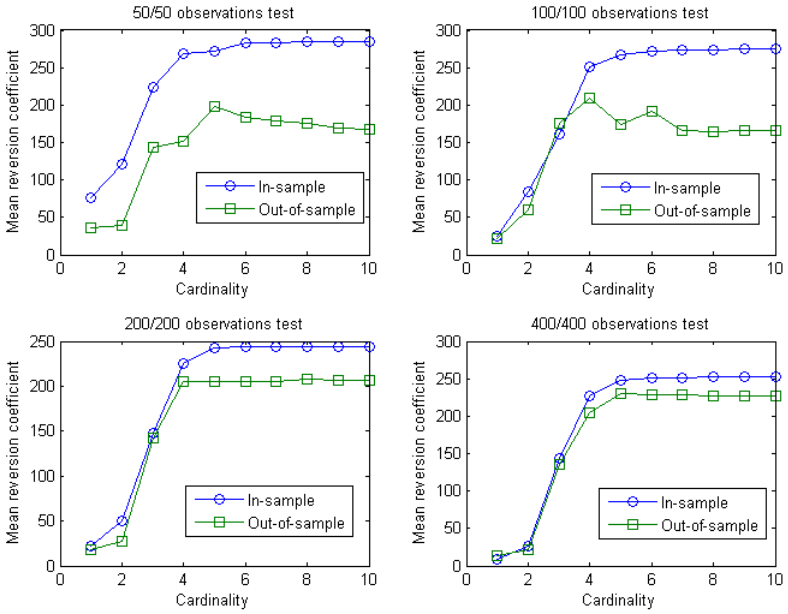

Fig. 4 The tests of the impact of the size of training set on out-of-sample performance based on synthetic data. The portfolios are constructed by the exhaustive search method under different cardinality constraints. Top left: 50 observations of the training set and 50 observations of the test set; Top right: 100 observations of the training set and 100 observations of the test set; Bottom left: 200 observations of the training set and 200 observations of the test set; Bottom right: 400 observations of the training set and 400 observations of the test set;

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

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

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

For in-sample tests, we used the entire U.S. swaps data set to estimate the parameters A and B and calculated the solutions for different γ. We found the minimizers to the problem (13) by the algorithm 2 in section 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 Figure 5. It shows the estimated mean reversion coefficients/trading opportunities versus the values of γ of this in-sample test on the entire swap data set.

For out-of-sample tests, we used every 100 consecutive observations to estimate those parameters and 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 Figure 6. It shows the average estimated mean reversion coefficients/trading opportunities versus the values of γ from in-sample and out-of-sample tests.

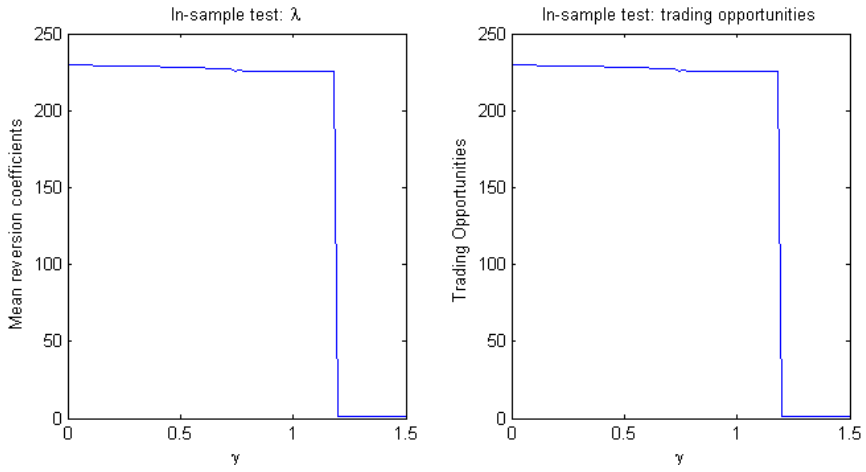

Fig. 5 In-sample tests on the whole data set of the U.S. swaps. Estimated mean reversion coefficients/trading opportunities versus the values of γ using the intermittent diffusion(ID) algorithm with the ratio of l_1 and l_2 norms penalty.

From the numerical results, we 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 5. After checking those portfolios, we found that we only recovered portfolios of cardinality 1, 5, 6, 7 and 8. We failed to recover the portfolios of cardinality 2, 3 and 4. The reason could be that we got trapped in the local minima.

The curves in Figure 6 look smoother, since they are the average performance of portfolios on about 12 data sets. Normally, the in-sample performance is better than the out-of-sample performance. However, if we treat the in-sample performance as a benchmark, 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 advantages of the ratio of l_1 and l_2 norms approach are: 1. It does not require any prior knowledge of the assets; 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

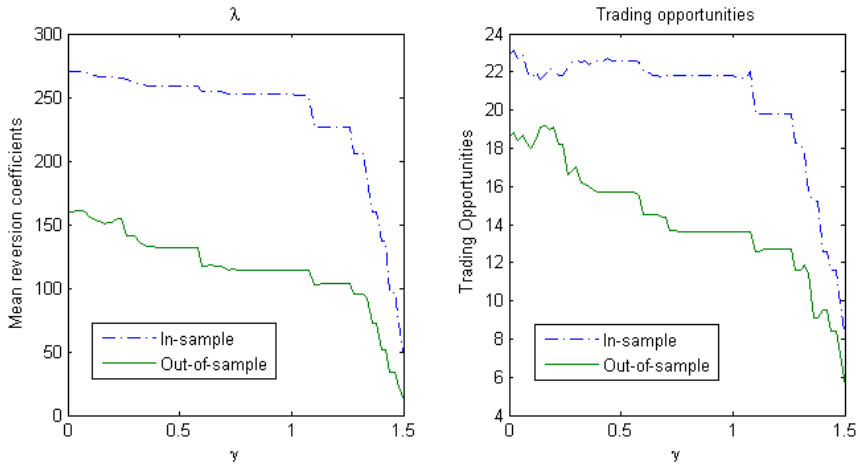

Fig. 6 In-sample and out-of-sample tests on the U.S. swaps. Average estimated mean reversion coefficients/trading opportunities versus the values of γ using the intermittent diffusion(ID) algorithm with the ratio of l_1 and l_2 norms penalty.

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 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.

7.5 Tests of the methods in section 6

7.5.1 In-Sample tests on 100 stocks

In this test, we applied the method in section 6 to a data set of 100 stocks. These stocks are the first 100 S&P stocks in ticker symbols' alphabetical order from our preselected list of S& P 500 stocks. Therefore, the size of the matrices A and B is 100×100.

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 treat the l^1 contraint-free solution as a benchmark. 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 in the densest solution.

Figure 7, 8 and 9 give the numerical results. The l_1 constraints, m, are in the range of 3 to 16 with a step size 1. Figure 7 demonstrates the mean reversion coefficients of portfolios built under different m's. The X-axis shows the level of m and the Y-axis is the mean reversion coefficients. Figure 7 also

demonstrates the cardinality of those portfolios and Figure 8 shows the trading opportunities. Figure 9 displays the weights of 100 stocks.

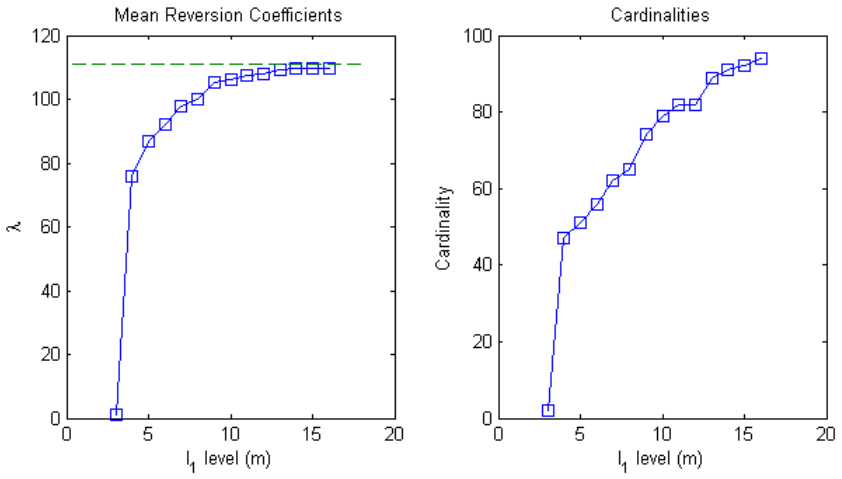

Fig. 7 Mean reversion coefficients/cardinality versus different l_1 constraints using the method of the least angle regression and the difference of convex functions algorithm on 100 stocks. The horizontal line in the left plot shows the largest possible mean reversion coefficient.

The average computational cost of the method of all these tests is 4.928 seconds. This is a remarkable improvement over the other non-greedy-search computational methods to date in terms of performance, efficiency and speed. The following table shows the computational cost of the three methods in the paper in this test:

	l_1 DC-PCA	l_1/l_2 Regularization	DC-LAR
Time costs(s)	100+	100+	4.9

In addition, Figure 10 shows the portfolios built by the DC-LAR method can achieve the similar performance as those built by semidefinite relaxation method in [6] when the exhaustive method and the greedy-search method are used as benchmarks.

7.5.2 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 11 shows the numerical results of the average λ and Figure 12 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,

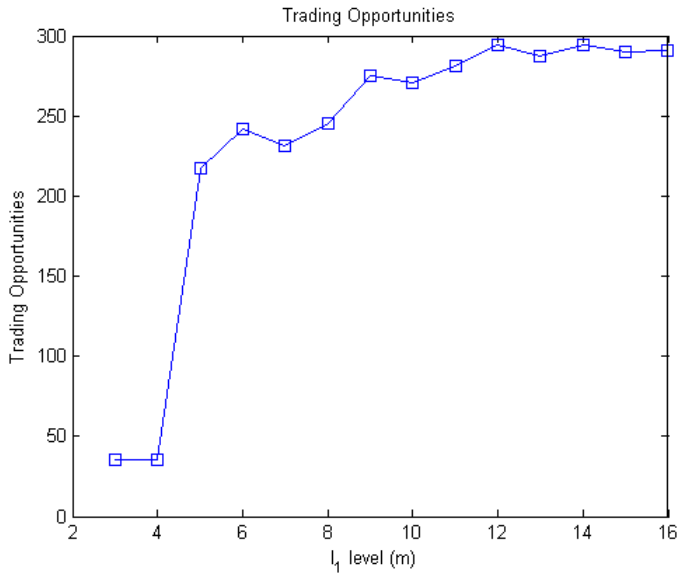

Fig. 8 Trading opportunities versus different l_1 constraints using the method of the least angle regression and DCA on 100 stocks.

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 11, 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.

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 lengths 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.

7.5.3 Out-of-sample tests on high dimensional synthetic data

In this section, we perform in-sample and out-of-sample tests on high dimensional synthetic data.

By presetting the matrix β and the noise covariance matrix Σ in the VAR(1) model (4), we generate a data set of size 1000×100. The eigenvalues of the matrix β lie inside the unit circle. We consider this set as the training set and estimate the matrices A and B based on this set. After picking

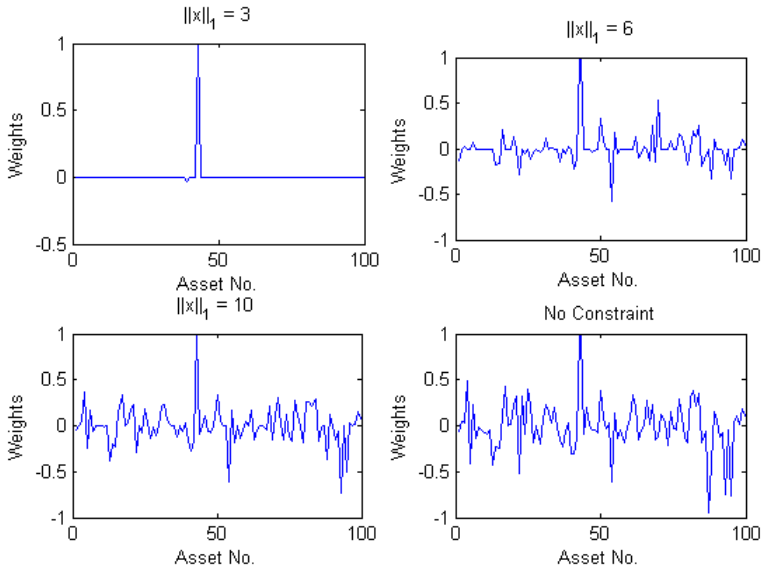

Fig. 9 Solutions under different l_1 constraints using the method of the least angle regression and the difference of convex functions algorithm on 100 stocks.

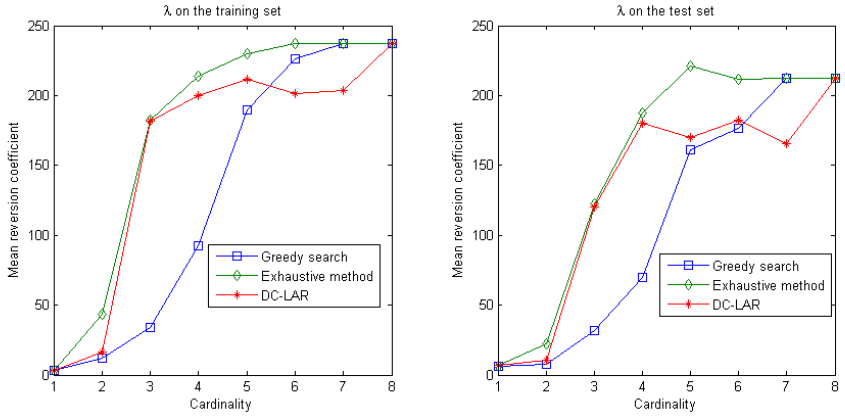

Fig. 10 In-sample and out-of-sample tests on 8-dimensional synthetic data. For the exhaustive method and the greedy search method, we plot the average in-sample and out-of-sample mean reversion coefficients for each cardinality. For DC-LAR, in each data set, we calculated the solutions for 20 different l_1 values and select the best vector for each cardinality. The red line shows the average mean reversion coefficients of the DC-LAR method for each cardinality.

an appropriate i, we solve for the optimal solutions under different m's (from 1.0683 to 17.0933 with a step size 1.0683) by the least angle regression and the difference of convex functions algorithm.

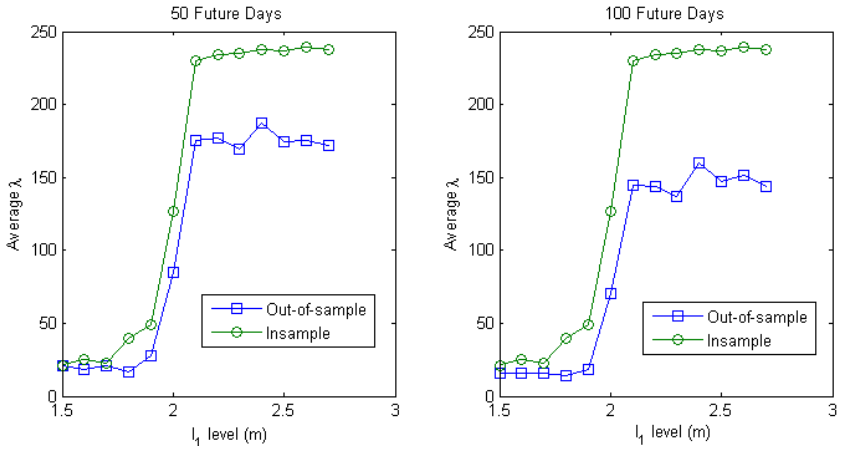

Fig. 11 In-sample and out-of-Sample tests on the U.S. swaps. Average mean reversion coefficients versus different l_1 constraints using the method of the least angle regression and DCA.

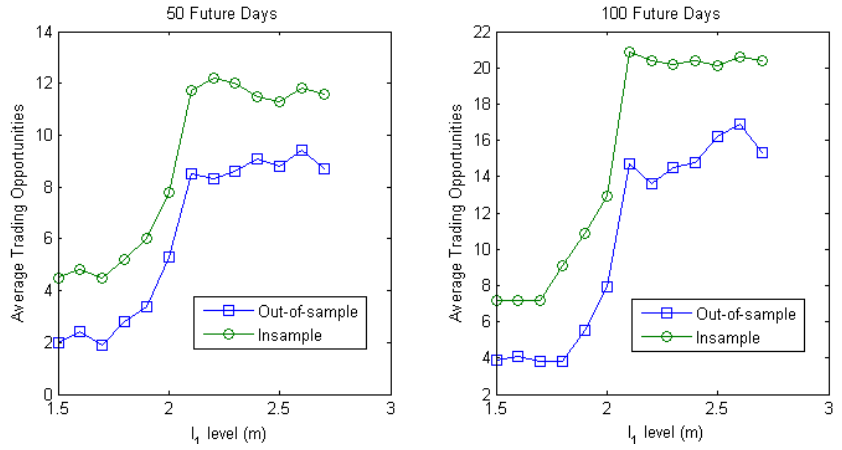

Fig. 12 In-sample and out-of-sample tests on the U.S. swaps. Average trading opportunities versus different l_1 constraints using the method of the least angle regression and DCA.

We also test another two special portfolios. One is a uniform portfolio which has equal weights on 100 assets. The other one is a random portfolio. We set the weight of the same ith asset to 1. For all the other assets, their weights are independently and identically distributed random variables which follows a uniform distribution in the interval $[-1, 1]$. These two portfolios can be considered as two benchmarks. We expect that the portfolios constructed under our algorithms will beat them in out-of-sample performance.

We generate 1000×100 observations based on the same VAR(1) model in each trial. These are test sets on which to evaluate our constructed portfolios and those two benchmark portfolios.

We generate 100 different test sets and then we compare the average of the estimated mean reversion coefficients and trading opportunities of these portfolios on both the training set and the test set. When we count the out-of-sample trading opportunities, we still use 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 13. The blue curves show the in-sample performance and the green curves show the out-of-sample performance. The red horizontal lines show the level of average mean reversion coefficients/trading opportunities of a random portfolio. The light blue horizontal dashed lines show the level of average mean reversion coefficients/trading opportunities of a uniform portfolio. 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 22% of the in-sample levels. We believe that the estimations of A and B will impact this performance. When the number of observations increases, we can expect that the out-of-sample performance will be better. The numbers of trading opportunities do not decrease so dramatically. They are about 55% of the in-sample trading opportunities. In fact, this is more important since the profits of convergence trading strategy come directly from those trading opportunities. The out-of-sample performance of the uniform portfolio and the random portfolio are much lower than our sparse and fast mean reverting portfolios.

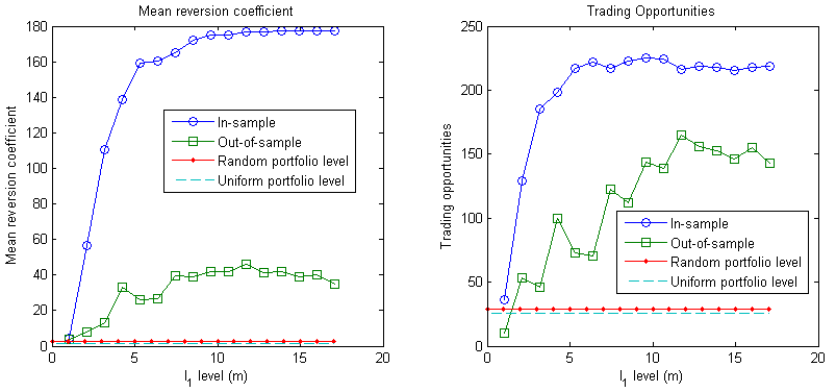

Fig. 13 In-sample and out-of-sample tests on high dimensional synthetic data from VAR(1) model. Average mean reversion coefficients/trading opportunities versus different l_1 constraints using the method of the least angle regression and the difference of convex functions algorithm. The blue curves show the in-sample performance and the green curves show the out-of-sample performance. The red horizontal line shows the level of average mean reversion coefficients/trading opportunities of a random portfolio. The light blue horizontal dashed line shows the level of average mean reversion coefficients/trading opportunities of a uniform portfolio.

7.6 Portfolio annual return performance tests

Trading opportunities used in the previous sections can be considered as a metric for optimistic investors who believe that the mean reverting will definitely happen in the future. However, in reality, investors normally will carry out some risk control measures. In this section, we applied a similar trading strategy used in [11] with an additional risk control rule.

The trading strategy can be summarized as follows:

- If the observed sample $P_t > \mu + \tau$, close long position if we already hold any. Otherwise we perform no action.
- If the observed sample $P_t < \mu - \tau$, open a long position if we are not in any position. Otherwise we perform no action.
- If the observed sample $\mu - \tau \leq P_t \leq \mu + \tau$, close any long position. Otherwise we perform no action.
- Close position if we lose 20% of position value.

We performed two tests using this new set of trading rules. One is an in-sample test on 100 S&P 500 stocks which is similar to section 7.5.1. The other one is an out-of-sample test on high dimensional synthetic data which is similar to section 7.5.3.

Figure 14 shows the number of trading opportunities under the risk control trading strategy and average annual returns for different l_1 constraints using real market data. When the l_1 level is above 10, the average annual returns are above 15%. The return on S&P 500 index was roughly 5% for this period of data.

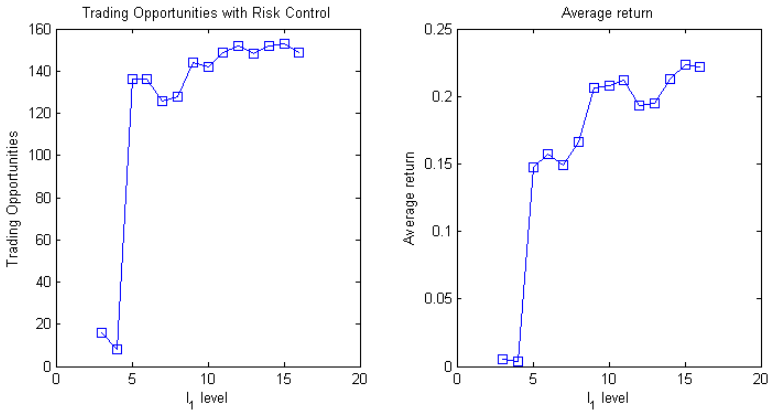

Fig. 14 Trading opportunities under risk control measure and average annual returns versus different l_1 constraints using the method of the least angle regression and DCA on 100 stocks.

Figure 15 shows the number of trading opportunities under the risk control trading strategy and average annual returns versus different l_1 constraints using synthetic data. The random portfolio and uniform portfolio are very

under-performed comparing with our results. The risk control measures also shrink the differences between in-sample and out-of-sample results.

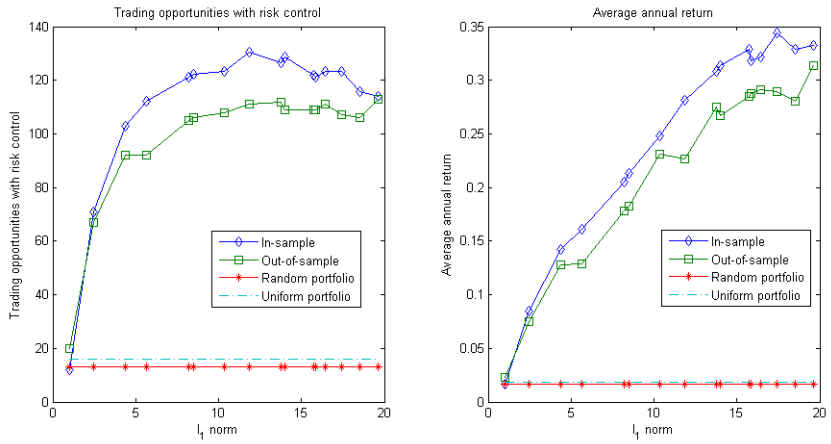

Fig. 15 In-sample and out-of-sample tests on high dimensional synthetic data from VAR(1) model. Trading opportunities under risk control measures and average annual returns versus different l_1 constraints using the method of the least angle regression and the difference of convex functions algorithm. The blue curves show the in-sample performance and the green curves show the out-of-sample performance. The red horizontal line shows the level of trading opportunities under risk control measures/average annual return of a random portfolio. The light blue horizontal dashed line shows the level of trading opportunities under risk control measures/average annual return of a uniform portfolio.

8 Conclusion

In this work, we used a 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.

Without any prior knowledge of the assets, we used two l_1-based constraint/penalty to enforce the sparsity. If we used l_1 norm, the problem could be considered as a simple extension of the sparse PCA. Using the algorithms in [18,29], we could solve the problem efficiently in some special cases. We also studied the properties of the ratio of l_1 and l_2 norms and designed an algorithm in solving the penalized optimization problem.

With prior knowledge of the assets, we adapted the l_1 norm to enforce the sparsity and we simplified the problem to a non-convex quadratic program. We developed various algorithms for approximating the global minimizer.

In our numerical tests, we applied our methods on both historical market data and synthetic data. We presented the in-sample and out-of-sample performance of the portfolios constructed under different algorithms and different problem settings. We also compared the computation costs of different algorithms for non-trivial sparse generalized eigenvalue problem (neither A nor B is a diagonal matrix).

Our numerical tests suggest that the combination of the least angle regression and the difference of convex functions algorithm is the best choice for non-trivial generalized eigenvalue problems. We carried out efficient computation for portfolios with hundreds of assets. The in-sample and out-of-sample performances of mean reversion coefficients, trading opportunities and annual returns have a similar performance trend as functions of cardinality or l_1 penalty level.

Acknowledgements

We would like to thank Dr. Wuan Luo for bringing reference [6] to our attention and for helpful communication.

A Semi-definite relaxation method derived from l_1/l_2

By setting $X = xx^T$, then the problem (12) is equivalent to

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

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

Then after a change of variables:

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

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

$$\begin{array}{ll} \min_Y & \text{trace}(AY) \\ s.t. & 1^T|Y|1 \leq m^2 z \\ & \text{trace}(Y) - z = 0 \\ & \text{trace}(BY) = 1 \\ & Y \succeq 0 \end{array} \tag{19}$$

If we set $\text{Card}(x) = k = m^2$, this is exactly the semidefinite relaxation in [6].

B Estimation of the matrices A and B

For the estimations below, we assume that each column of the data matrix S represents an asset and its mean is 0. Its size is $l \times n$, so we have l observations and n assets.

We define S_c and S_f in the following way:

$$S_c = \begin{pmatrix} S_1^T \\ \vdots \\ S_{l-1}^T \end{pmatrix} = \begin{pmatrix} S_{11} & \dots & S_{1n} \\ \vdots & \ddots & \vdots \\ S_{l-1,1} & \dots & S_{l-1,n} \end{pmatrix}$$

$$S_f = \begin{pmatrix} S_2^T \\ \vdots \\ S_l^T \end{pmatrix} = \begin{pmatrix} S_{21} & \dots & S_{2n} \\ \vdots & \ddots & \vdots \\ S_{l1} & \dots & S_{ln} \end{pmatrix}$$

where S_{ti} is the value at time t of the ith asset.

B.1 Predictability

In [11], the authors discussed several methods in estimating β and Γ in VAR(1) model (4). In most cases, the number of the observations of assets values l is greater than the number of assets n. Under this case and previous assumptions, we could use the following estimates:

$$\hat{\beta} = (S_c^T S_c)^{-1}(S_c^T S_f)) \qquad \hat{\Gamma} = \frac{1}{l-1} S_c^T S_c$$

Therefore, the matrices in problem 7 can be estimated as:

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

B.2 Direct OU estimator

Using a similar method as B.1, we estimate the matrices A and B as follows:

$$\hat{A} = \frac{1}{l-1}(S_c^T S_f + S_f^T S_c) \qquad \hat{B} = \frac{2}{l-1} S_c^T S_c$$

References

1. Barekat, F., Yin, K., Caflisch, R. E., Osher, S. J., Lai, R., and Ozolins, V. Compressed wannier modes found from an l_1 regularized energy functional. *arXiv preprint arXiv:1403.6883* (2014).
2. Box, G. E., and Tiao, G. C. A canonical analysis of multiple time series. *Biometrika 64*, 2 (1977), 355–365.
3. Box, M., Davies, D., Swann, W. H., and Australia, I. *Non-linear optimization techniques*. No. 5. Oliver & Boyd Edinburgh, 1969.
4. Chow, S.-N., Yang, T.-S., and Zhou, H. Global optimizations by intermittent diffusion. *National Science Council Tunghai University Endowment Fund for Academic Advancement Mathematics Research Promotion Center* (2009), 121.
5. Cuturi, M., and D'aspremont, A. Mean reversion with a variance threshold. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)* (2013), pp. 271–279.
6. d'Aspremont, A. Identifying small mean-reverting portfolios. *Quantitative Finance 11*, 3 (2011), 351–364.
7. d'Aspremont, A., El Ghaoui, L., Jordan, M. I., and Lanckriet, G. R. A direct formulation for sparse pca using semidefinite programming. *SIAM review 49*, 3 (2007), 434–448.

8. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. Least angle regression. *The Annals of statistics 32*, 2 (2004), 407–499.
9. Engle, R. F., and Granger, C. W. Co-integration and error correction: representation, estimation, and testing. *Econometrica: journal of the Econometric Society* (1987), 251–276.
10. Esser, E., Lou, Y., and Xin, J. A method for finding structured sparse solutions to nonnegative least squares problems with applications. *SIAM Journal on Imaging Sciences 6*, 4 (2013), 2010–2046.
11. Fogarasi, N., and Levendovszky, J. Improved parameter estimation and simple trading algorithm for sparse, mean reverting portfolios. In *Annales Univ. Sci. Budapest., Sect. Comp* (2012), vol. 37, pp. 121–144.
12. Horst, R., and Thoai, N. V. Dc programming: overview. *Journal of Optimization Theory and Applications 103*, 1 (1999), 1–43.
13. Hotelling, H. Relations between two sets of variates. *Biometrika 28*, 3-4 (1936), 321–377.
14. Hoyer, P. O. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research 5* (2004), 1457–1469.
15. Hu, Y., and Long, H. Least squares estimator for ornstein–uhlenbeck processes driven by α-stable motions. *Stochastic Processes and their applications 119*, 8 (2009), 2465–2480.
16. Ji, H., Li, J., Shen, Z., and Wang, K. Image deconvolution using a characterization of sharp images in wavelet domain. *Applied and Computational Harmonic Analysis 32*, 2 (2012), 295–304.
17. Jolliffe, I. T., Trendafilov, N. T., and Uddin, M. A modified principal component technique based on the lasso. *Journal of computational and Graphical Statistics 12*, 3 (2003), 531–547.
18. Journée, M., Nesterov, Y., Richtárik, P., and Sepulchre, R. Generalized power method for sparse principal component analysis. *The Journal of Machine Learning Research 11* (2010), 517–553.
19. Krishnan, D., Tay, T., and Fergus, R. Blind deconvolution using a normalized sparsity measure. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (2011), IEEE, pp. 233–240.
20. Lai, R., and Osher, S. A splitting method for orthogonality constrained problems. *Journal of Scientific Computing 58*, 2 (2014), 431–449.
21. Le Thi, H., Huynh, V., and Pham, D. T. Convergence analysis of dc algorithm for dc programming with subanalytic data. *Ann. Oper. Res. Technical Report, LMI, INSA-Rouen* (2009).
22. Le Thi, H. A., and Pham, D. T. Solving a class of linearly constrained indefinite quadratic problems by dc algorithms. *Journal of Global Optimization 11*, 3 (1997), 253–285.
23. Natarajan, B. K. Sparse approximate solutions to linear systems. *SIAM journal on computing 24*, 2 (1995), 227–234.
24. Ngo, T. T., Bellalij, M., and Saad, Y. The trace ratio optimization problem. *SIAM Review 54*, 3 (2012), 545–569.
25. Ozoliņš, V., Lai, R., Caflisch, R., and Osher, S. Compressed modes for variational problems in mathematics and physics. *Proceedings of the National Academy of Sciences 110*, 46 (2013), 18368–18373.
26. Ozoliņš, V., Lai, R., Caflisch, R., and Osher, S. Compressed plane waves yield a compactly supported multiresolution basis for the laplace operator. *Proceedings of the National Academy of Sciences 111*, 5 (2014), 1691–1696.
27. Pham, D. T., and Le Thi, H. A. Convex analysis approach to dc programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica 22*, 1 (1997), 289–355.
28. Pham, D. T., and Le Thi, H. A. A dc optimization algorithm for solving the trust-region subproblem. *SIAM Journal on Optimization 8*, 2 (1998), 476–505.
29. Richtárik, P., Takáč, M., and Ahipaşaoğlu, S. D. Alternating maximization: unifying framework for 8 sparse pca formulations and efficient parallel codes. *arXiv preprint arXiv:1212.4137* (2012).

30. Sriperumbudur, B. K., Torres, D. A., and Lanckriet, G. R. Sparse eigen methods by dc programming. In *Proceedings of the 24th international conference on Machine learning* (2007), ACM, pp. 831–838.
31. Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* (1996), 267–288.
32. Yin, P., Esser, E., and Xin, J. Ratio and difference of l1 and l2 norms and sparse representation with coherent dictionaries. *Commun. Inform. Systems 14*, 2 (2014), 87–109.
33. Yu, J. Bias in the estimation of the mean reversion parameter in continuous time models. *Journal of Econometrics 169*, 1 (2012), 114–122.
34. Zou, H., Hastie, T., and Tibshirani, R. Sparse principal component analysis. *Journal of computational and graphical statistics 15*, 2 (2006), 265–286.