Department of Industrial Engineering and Management

Technical Report

No. 2013-9

Multi-Period Portfolio Selection Using Kernel-Based Control Policy with Dimensionality Reduction

Yuichi TAKANO and Jun-ya GOTOH

July, 2013

Tokyo Institute of Technology

2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, JAPAN
http://www.me.titech.ac.jp/index-e.html

Multi-Period Portfolio Selection Using Kernel-Based Control Policy with Dimensionality Reduction

Yuichi Takano takano.y.ad@m.titech.ac.jp
Department of Industrial Engineering and Management
Graduate School of Decision Science and Technology, Tokyo Institute of Technology
2-12-1-W9-77 Ookayama, Meguro-ku, Tokyo 152-8552, JAPAN

Jun-ya Gotoh jgoto@indsys.chuo-u.ac.jp
Department of Industrial and Systems Engineering, Chuo University
1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, JAPAN

Abstract

This paper studies a nonlinear control policy for multi-period investment. The nonlinear strategy we develop is categorized as a kernel method, but solving large-scale instances of the resulting optimization problem in a direct manner is computationally intractable in the literature. In order to overcome this difficulty, we employ a dimensionality reduction technique which is often used in principal component analysis. Numerical experiments show that our strategy works not only to reduce the computation time, but also to improve out-of-sample investment performance.

Keywords: Multi-period portfolio selection, Kernel method, Control policy, Dimensionality reduction

1 Introduction

Risk management based on diversified investment makes it possible to mitigate the risk of suffering a large loss while securing a certain level of profitability, and portfolio selection accordingly plays an important role in financial decision making (see, e.g., Cornuejols and Tütüncü [5]). Portfolio selection is usually conducted in a single-period framework, as initially formulated by Markowitz [14]. It is, however, advantageous for investors to periodically adjust their portfolios by following an effective rebalancing strategy. In this respect, the traditional single-period model is not sufficient. Indeed, Mulvey et al. [17] state that multi-period models can enhance risk-adjusted performance and help investors evaluate the probability of reaching a certain target by linking asset and liability policies.

Among the various rebalancing strategies, constant rebalancing reverts the investment proportion to the original proportion at the beginning of every period. It is known that a constant rebalancing strategy achieves the optimal growth rate of wealth on the assumption that asset returns in each period are independent and identically distributed (see, e.g., [1]). Due to this fact, a number of studies (see, e.g., [7, 15, 25, 27]) have dealt with multi-period portfolio optimization with the constant rebalancing strategy.

However, it has been demonstrated, e.g., in [12, 13], that stock returns are serially dependent; therefore, it is probably effective to dynamically rebalance the portfolio in view of the observed asset returns. For instance, DeMiguel et al. [6] improve the out-of-sample investment performance of single-period models by predicting future stock returns through the use of a vector autoregressive (VAR) model. More importantly, Fleten et al. [7] have shown by means of an out-of-sample simulation test that the stochastic dynamic approach dominates the constant rebalancing strategy. These observations motivated us to develop a rebalancing strategy for exploiting the time-series dependence of stock returns.

Multi-period portfolio selection was first framed as a stochastic control problem (see Infanger [10] for detailed references). In general, however, it is very difficult to handle a stochastic control problem of a practical size because it requires one to solve a large-scale dynamic programming problem or partial differential equations. Consequently, a number of studies have focused on mathematical optimization approaches with an appropriate uncertainty modeling. Among them the simulated path model (see, e.g., Hibiki [9]) describes multi-period scenarios of asset returns using a number of simulated paths. The actual market behavior can be simulated in detail by this model, but there is no room for conditional investment decisions in this model due to what is called the "non-anticipativity condition," which requires one to prevent investment decisions from depending on future observations on each simulated path. By contrast, the scenario tree model (see Steinbach [24] for detailed references) enables one to make conditional investment decisions in each future state; however, this model is disadvantageous in that the size of the resultant optimization problem grows exponentially as the number of time periods increases. The hybrid model devised by Hibiki [8] integrates the simulated path model and the scenario tree model; nevertheless, it is still computationally burdensome to make conditional investment decisions in the hybrid model as well as in the scenario tree model.

In view of these facts, we shall utilize a control policy, which maps past outcomes to the investment amount to be rebalanced. While the control policy enables one to make conditional investment decisions, determining the best control policy generally leads to a computationally intractable optimization. Accordingly, most studies (e.g., [2, 3, 4, 16, 19, 22]) have dealt with a restricted class of control policies, e.g., affine functions of the past outcomes.

On the other hand, the authors of this paper build in [26] a computational framework based on the kernel method for finding the best nonlinear control policy in the simulated path model. Such a kernel method is often employed in estimating nonlinear statistical models in machine learning (see, e.g., Schölkopf and Smola [23]), and it allows one to treat a highly nonlinear transformation in the feature space efficiently. Numerical experiments in [26] show that a model with this kernel-based control policy performs better than other models.

However, we are yet confronted by two difficulties in using the kernel-based control policy: long computation time and overfitting. Indeed, the experiments in [26] show that substantial time is required even for small problems, despite the fact that the problem is formulated as a convex quadratic programming problem. In addition, since the kernel approach admits a highly nonlinear mapping, the resulting control policy may overfit the scenarios used in the optimization problem and consequently weaken its out-of-sample performance.

The purpose of this paper is to devise an approach for efficiently solving the multi-period portfolio selection problem with a kernel-based control policy [26] and for further improving its investment performance. To this end, a method of problem reduction is posed based on a dimensionality reduction technique which is often used in principal component analysis (PCA). More precisely, our application is directly related to what is called kernel principal component analysis (kernel PCA), which is an extension of PCA into a feature space of (possibly, infinitely) high dimension (see, e.g., Schölkopf and Smola [23]). Yajima et al. [28] use a dimensionality reduction technique to reduce the problem size of a nonlinear support vector machine. Their results encouraged us to apply a similar reduction method to our multi-period portfolio selection problem. In addition, it has been demonstrated in the context of regression analysis that kernel PCA has an effect of de-noising (see, e.g., [11, 20]). This means that kernel PCA has the potential of not just achieving a high degree of computation efficiency, but also improving investment performance.

The rest of the paper is organized as follows: In Section 2, we present a multi-period portfolio selection model equipped with a kernel-based control policy. In Section 3, we develop a method for reducing the problem size by means of eigenvalue decomposition and formulate an optimization problem in a reduced form. In Section 4, numerical experiments show that our optimization model sharply lessened the computation time without worsening investment performance. Furthermore, our optimization-based approach avoided overfitting, and accordingly, it enhanced out-of-sample investment performance in certain situations. Finally, conclusions are given in Section 5.

2 Control Policy for Multi-Period Portfolio Selection

In this section, after giving a mathematical description of portfolio dynamics, we formulate the multi-period portfolio selection problem with a kernel-based nonlinear control policy.

2.1 Preliminaries and portfolio dynamics

The terminology and notation used in this subsection are as follows:

Index Sets

$\mathcal{I} := \{1, 2, ..., I\}$	:	index set of investable financial assets (where asset 1 is cash)
$\mathcal{S} := \{1, 2, ..., S\}$	:	index set of given scenarios (or simulated paths)
$\mathcal{T} := \{1, 2, ..., T\}$	:	index set of planning time periods

Decision Variables

$x_{i,s}(t)$: investment amount in asset i at the end of period t in scenario s $(i \in \mathcal{I},\ s \in \mathcal{S},\ t \in \mathcal{T})$

$u_i(t)$: adjustment of asset i at the beginning of period t $(i \in \mathcal{I},\ t \in \mathcal{T})$

$u_{i,s}(t)$: adjustment of asset i at the beginning of period t in scenario s $(i \in \mathcal{I},\ s \in \mathcal{S},\ t \in \mathcal{T} \setminus \{1\})$

$v_s(t)$: portfolio value at the end of period t in scenario s $(s \in \mathcal{S},\ t \in \mathcal{T})$

$a(t)$: the value-at-risk (VaR) in period t $(t \in \mathcal{T})$

$z_s(t)$: auxiliary decision variable for calculating the conditional value-at-risk (CVaR) in period t $(s \in \mathcal{S},\ t \in \mathcal{T})$

Given Constants

$\bar{x}_i(0)$: the initial holdings of asset i $(i \in \mathcal{I})$

$C(t)$: net cash flow at the beginning of period t $(t \in \mathcal{T})$

$R_{i,s}(t)$: total return of asset i in period t in scenario s $(i \in \mathcal{I},\ s \in \mathcal{S},\ t \in \mathcal{T})$

P_s : occurrence probability of scenario s $(s \in \mathcal{S})$

L_i, U_i : lower and upper limits of the investment proportion in asset i $(i \in \mathcal{I})$

User Defined Parameters

α : the trade-off parameter between profitability and risk (where $\alpha \in (0,1)$)

β : the confidence level of the CVaR (where $\beta \in (0,1)$)

$\eta(t)$: weight of the expected portfolio value at the end of period t (where $\eta(t) \geq 0$) $(t \in \mathcal{T})$

$\theta(t)$: weight of the CVaR in period t (where $\theta(t) \geq 0$) $(t \in \mathcal{T})$

Functions

γ_i : transaction cost function of asset i (where $\gamma_i : \mathbb{R} \to \mathbb{R}_+$) $(i \in \mathcal{I})$

Figure 1 illustrates the portfolio dynamics in a scenario s. We assume that one has an initial portfolio $\bar{x}_i(0)$, $i \in \mathcal{I}$. If the investor has no initial holdings, $\bar{x}_i(0)$ can be set to 0 for all $i \in \mathcal{I}$.

One adjusts the portfolio at the beginning of each period as follows:

$$\begin{aligned} &x_i^+(0) := \bar{x}_i(0) + u_i(1), \\ &x_{i,s}^+(t-1) := x_{i,s}(t-1) + u_{i,s}(t),\ t \in \mathcal{T} \setminus \{1\}. \end{aligned} \tag{1}$$

The investment amount changes over the period due to the changing price of each asset. Specifically, by multiplying the investment amount by the total return, we derive the following

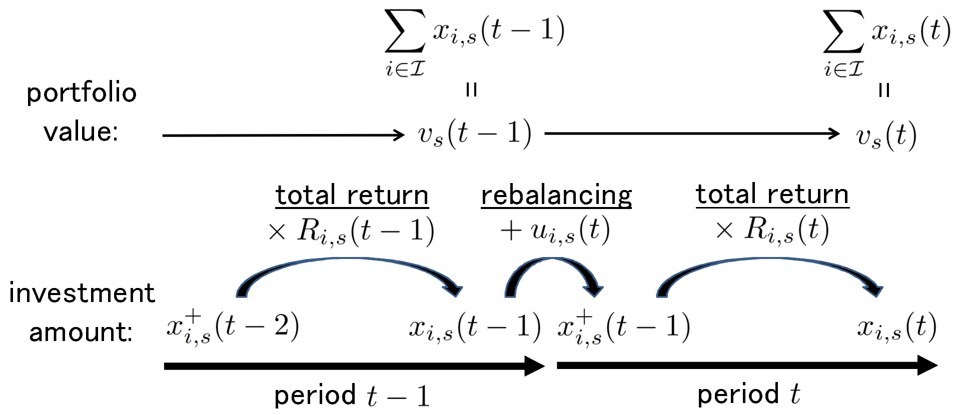


Figure 1: Portfolio dynamics in scenario s

portfolio dynamics equations:

$$\begin{aligned} x_{i,s}(1) &= R_{i,s}(1)\, x_i^+(0) = R_{i,s}(1)\left(\bar{x}_i(0) + u_i(1)\right), \\ x_{i,s}(t) &= R_{i,s}(t)\, x_{i,s}^+(t-1) = R_{i,s}(t)\left(x_{i,s}(t-1) + u_{i,s}(t)\right),\ t \in \mathcal{T}\setminus\{1\}. \end{aligned} \tag{2}$$

The portfolio value at the end of period $t \in \mathcal{T}$ in scenario s is the sum of investments:

$$v_s(t) = \sum_{i\in\mathcal{I}} x_{i,s}(t),$$

and therefore, the expected portfolio value at the end of period $t \in \mathcal{T}$ is $\sum_{s\in\mathcal{S}} P_s v_s(t)$.

In addition, the adjustments must satisfy the following cash flow balance equations in each period:

$$\begin{aligned} &\sum_{i\in\mathcal{I}} u_i(1) = C(1) - \sum_{i\in\mathcal{I}} \gamma_i(u_i(1)), \\ &\sum_{i\in\mathcal{I}} u_{i,s}(t) = C(t) - \sum_{i\in\mathcal{I}} \gamma_i(u_{i,s}(t)),\ t \in \mathcal{T}\setminus\{1\}. \end{aligned} \tag{3}$$

When a self-financing strategy is employed, the net cash flow $C(t)$ is set to 0 for all $t \in \mathcal{T}$, and the equations (3) imply that the sum of sales is equal to the sum of purchases and transaction costs. In this paper, we consider only a linear transaction cost. Specifically, it is assumed that τ_i^{buy} and τ_i^{sell} are transaction costs per unit for buying and selling asset i, respectively. Then by introducing auxiliary decision variables, u^{buy} and u^{sell}, the transaction cost function of an adjustment u can be represented with the following linear constraints:

$$\gamma_i(u) = \tau_i^{\text{buy}} u^{\text{buy}} + \tau_i^{\text{sell}} u^{\text{sell}},\ u = u^{\text{buy}} - u^{\text{sell}},\ u^{\text{buy}} \geq 0,\ u^{\text{sell}} \geq 0.$$

Moreover, we impose constraints on the investment proportion (1) right after rebalancing:

$$\begin{aligned} &L_i \sum_{j\in\mathcal{I}} \left(\bar{x}_j(0) + u_j(1)\right) \leq \bar{x}_i(0) + u_i(1) \leq U_i \sum_{j\in\mathcal{I}} \left(\bar{x}_j(0) + u_j(1)\right), \\ &L_i \sum_{j\in\mathcal{I}} \left(x_{j,s}(t-1) + u_{j,s}(t)\right) \leq x_{i,s}(t-1) + u_{i,s}(t) \leq U_i \sum_{j\in\mathcal{I}} \left(x_{j,s}(t-1) + u_{j,s}(t)\right),\ t \in \mathcal{T}\setminus\{1\}. \end{aligned}$$

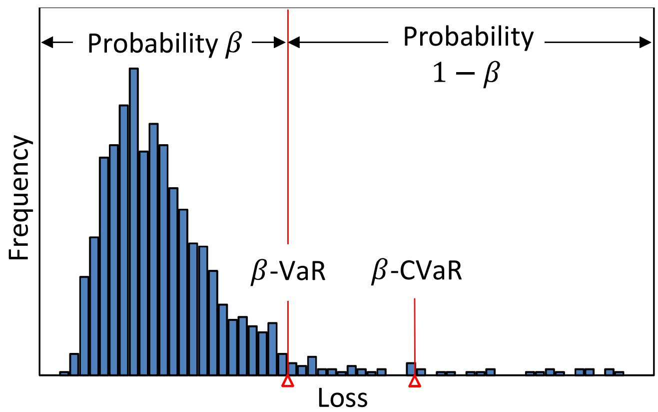


Figure 2: Value-at-risk and conditional value-at-risk

(4)

In this paper, we shall use the expected portfolio value as a measure of profitability and the conditional value-at-risk (CVaR) as a measure of risk. It is well known that the CVaR has desirable computational and theoretical properties (see, e.g., [18, 21] for the details). β-CVaR can approximately be regarded as the conditional expectation of a random loss exceeding the β-value-at-risk (β-VaR), which is the β-quantile of the random loss (see Figure 2). Now the random loss is defined as the negative of the portfolio value at the end of period t, i.e., $-v_s(t)$, and the corresponding CVaR in each period is the optimal value of the following linear optimization problem (see [21]):

$$\min_{a(t),\, z_s(t)} \left\{ a(t) + \frac{1}{1-\beta} \sum_{s \in \mathcal{S}} P_s z_s(t) \;\middle|\; z_s(t) \geq -v_s(t) - a(t),\ z_s(t) \geq 0,\ s \in \mathcal{S} \right\}.$$

To take into account the investment performance in all periods, we minimize the following weighted sum of the measures of profitability and risk:

$$\underbrace{(1-\alpha) \sum_{t \in \mathcal{T}} \theta(t) \left(a(t) + \frac{1}{1-\beta} \sum_{s \in \mathcal{S}} P_s z_s(t) \right)}_{(5.\,\mathrm{a})} \underbrace{- \alpha \sum_{t \in \mathcal{T}} \eta(t) \sum_{s \in \mathcal{S}} P_s v_s(t)}_{(5.\,\mathrm{b})}. \tag{5}$$

2.2 Optimization of nonlinear control policy

The next thing we need to do is establish an effective rebalancing strategy for adjusting the portfolio. For this purpose, we shall use the control policy for making conditional investment decisions.

By following the previous studies [2, 3, 4], we define the control policy $\mathcal{F}_{i,t}$ as a function of the past investment amount and the past total return. Specifically, the adjustments $u_{i,s}(t)$ are determined as follows:

$$u_{i,s}(t) = \mathcal{F}_{i,t}\left(\boldsymbol{x}_s(t-1), \boldsymbol{R}_s(t-1)\right),\ t \in \mathcal{T} \setminus \{1\}, \tag{6}$$

where $\boldsymbol{x}_s(t) := (x_{i,s}(k);\ i \in \mathcal{I},\ 1 \leq k \leq t)$ and $\boldsymbol{R}_s(t) := (R_{i,s}(k);\ i \in \mathcal{I},\ 1 \leq k \leq t)$. Note that the function $\mathcal{F}_{i,t}$ itself is independent of the scenario s, whereas the adjustments $u_{i,s}(t)$ are dynamically determined depending on the past outcomes $\boldsymbol{x}_s(t-1)$ and $\boldsymbol{R}_s(t-1)$. As a result, we can make conditional investment decisions corresponding to each scenario.

We can, however, omit the past investment amounts, $\boldsymbol{x}_s(t-1)$, from the control policies (6) and accordingly use the following control policies:

$$u_{i,s}(t) = \mathcal{F}_{i,t}\left(\boldsymbol{R}_s(t-1)\right),\ t = \mathcal{T} \setminus \{1\} \tag{7}$$

because it has been shown in [4, 26] that the above control policies (6) and (7) have the same capability to create investment strategies.

More specifically, we shall consider a control policy of the form,

$$u_{i,s}(t) = u_i(t) + \boldsymbol{w}_i(t)^\top \boldsymbol{\phi}_{i,t}\left(\boldsymbol{R}_s(t-1)\right),\ t = \mathcal{T} \setminus \{1\}, \tag{8}$$

where $\boldsymbol{\phi}_{i,t}$ are nonlinear mappings from the original space $\mathbb{R}^{I \times (t-1)}$ to a high-dimensional feature space $\mathbb{R}^{N_{i,t}}$, and $\boldsymbol{w}_i(t)$ are decision variables representing the weight of the associated feature. After the fashion of machine learning, we call the image of the mapping a feature vector. A simple example of features would be the polynomials of the total return in the period $t-1$:

$$\boldsymbol{\phi}_{i,t}\left(\boldsymbol{R}_s(t-1)\right) = (R_{i,s}(t-1), R_{i,s}(t-1)^2, \ldots, R_{i,s}(t-1)^{N_{i,t}})^\top. \tag{9}$$

Note that the feature vector may be excessively nonlinear and, accordingly, so may the policy (9). In order for the policy not to overfit the total returns, $R_{i,s}(t)$, we add regularization terms, $\|\boldsymbol{w}_i(t)\|^2$, to be minimized.

We can now formulate the multi-period portfolio selection problem with nonlinear control policies (8):

$$
\begin{array}{lll}
\underset{\substack{a(t),\ u_i(t) \\ u_{i,s}(t),\ v_s(t) \\ \boldsymbol{w}_i(t),\ x_{i,s}(t) \\ z_s(t)}}{\text{minimize}} & (1-\alpha)\sum_{t\in\mathcal{T}}\theta(t)\left(a(t)+\frac{1}{1-\beta}\sum_{s\in\mathcal{S}}P_s z_s(t)\right)-\alpha\sum_{t\in\mathcal{T}}\eta(t)\sum_{s\in\mathcal{S}}P_s v_s(t) & \\
& \qquad +\lambda\sum_{i\in\mathcal{I}\setminus\{1\}}\sum_{t\in\mathcal{T}\setminus\{1\}}\|\boldsymbol{w}_i(t)\|^2 & \cdots(10.\mathrm{a}) \\
\text{subject to} & z_s(t)\ge -v_s(t)-a(t),\ z_s(t)\ge 0,\ s\in\mathcal{S},\ t\in\mathcal{T} & \cdots(10.\mathrm{b}) \\
& x_{i,s}(1)=R_{i,s}(1)\left(\bar{x}_i(0)+u_i(1)\right),\ i\in\mathcal{I},\ s\in\mathcal{S} & \cdots(10.\mathrm{c}) \\
& x_{i,s}(t)=R_{i,s}(t)\left(x_{i,s}(t-1)+u_{i,s}(t)\right),\ i\in\mathcal{I},\ s\in\mathcal{S},\ t\in\mathcal{T}\setminus\{1\} & \cdots(10.\mathrm{d}) \\
& v_s(t)=\sum_{i\in\mathcal{I}}x_{i,s}(t),\ s\in\mathcal{S},\ t\in\mathcal{T} & \cdots(10.\mathrm{e}) \\
& \sum_{i\in\mathcal{I}}u_i(1)=C(1)-\sum_{i\in\mathcal{I}}\gamma_i(u_i(1)) & \cdots(10.\mathrm{f}) \\
& \sum_{i\in\mathcal{I}}u_{i,s}(t)=C(t)-\sum_{i\in\mathcal{I}}\gamma_i(u_{i,s}(t)),\ s\in\mathcal{S},\ t\in\mathcal{T}\setminus\{1\} & \cdots(10.\mathrm{g}) \\
& L_i\sum_{j\in\mathcal{I}}\left(\bar{x}_j(0)+u_j(1)\right)\le \bar{x}_i(0)+u_i(1)\le U_i\sum_{j\in\mathcal{I}}\left(\bar{x}_j(0)+u_j(1)\right),\ i\in\mathcal{I} & \cdots(10.\mathrm{h}) \\
& L_i\sum_{j\in\mathcal{I}}\left(x_{j,s}(t-1)+u_{j,s}(t)\right)\le x_{i,s}(t-1)+u_{i,s}(t)\le U_i\sum_{j\in\mathcal{I}}\left(x_{j,s}(t-1)+u_{j,s}(t)\right), & \\
& \qquad i\in\mathcal{I},\ s\in\mathcal{S},\ t\in\mathcal{T}\setminus\{1\} & \cdots(10.\mathrm{i}) \\
& u_{i,s}(t)=u_i(t)+\boldsymbol{w}_i(t)^\top\boldsymbol{\phi}_{i,t}\left(\boldsymbol{R}_s(t-1)\right),\ i\in\mathcal{I}\setminus\{1\},\ s\in\mathcal{S},\ t\in\mathcal{T}\setminus\{1\}, & \\
& & \cdots(10.\mathrm{j})
\end{array}
\tag{10}
$$

where $\lambda > 0$ is a trade-off parameter that controls the balance between the regularization term and the investment performance, which consists of the sum of the CVaR and the expected portfolio value. Note that the adjustments of cash, $u_{1,s}(t)$, are uniquely determined from the adjustments of other assets through the cash flow balance equations (10. g). Therefore, as seen from (10. j), the control policy is not used to rebalance the cash (asset 1).

The above optimization problem is a convex quadratic optimization problem. Nevertheless, the main difficulty is that the feature vectors, $\boldsymbol{\phi}_{i,t}\left(\boldsymbol{R}_s(t-1)\right)$, need to be fixed before solving problem (10). It is not clear, in advance, what nonlinear term of $\boldsymbol{R}_s(t-1)$ will improve investment performance. Thus, to guarantee a high standard of investment performance, a variety of nonlinear terms need to be included in the feature vectors. As is clear from (9), infinite-dimensional feature vectors (i.e., $N_{i,t} = \infty$) are necessary to create all nonlinear functions by using $\boldsymbol{\phi}_{i,t}$. Unfortunately, however, problem (10) is extremely difficult to solve when $N_{i,t}$ is very high.

To overcome this difficulty, the authors in [26] employed the kernel method, which is a class of algorithm for analyzing nonlinear data in machine learning (see, e.g., [23]). The kernel method enables us to determine an optimal control policy (8) without explicitly computing in a high-dimensional feature space.

Let $\mathcal{K}_{i,\ell,s}(t)$ be the kernel functions:

$$\mathcal{K}_{i,\ell,s}(t) := \boldsymbol{\phi}_{i,t}\left(\boldsymbol{R}_\ell(t-1)\right)^\top \boldsymbol{\phi}_{i,t}\left(\boldsymbol{R}_s(t-1)\right). \tag{11}$$

We then use the following theorem:

Theorem 2.1 (Representer theorem [23]) Let $\boldsymbol{w}_i^*(t)$ be optimal solutions to the problem (10). Then there exist $e_{i,s}(t)$, $i \in \mathcal{I} \setminus \{1\}$, $s \in \mathcal{S}$, $t \in \mathcal{T} \setminus \{1\}$ such that

$$\boldsymbol{w}_i^*(t)^\top \boldsymbol{\phi}_{i,t}\left(\boldsymbol{R}_s(t-1)\right) = \sum_{\ell \in \mathcal{S}} e_{i,\ell}(t)\, \mathcal{K}_{i,\ell,s}(t),\ i \in \mathcal{I} \setminus \{1\},\ s \in \mathcal{S},\ t \in \mathcal{T} \setminus \{1\}.$$

Proof. See Theorem 3.1 in Takano and Gotoh [26]. ■

Theorem 2.1 states that the optimal adjustments, $u_{i,s}^*(t)$, can be computed without any concern about the nature of the feature vectors.

Noting that the regularization terms in the problem (10) can be expressed as

$$\lambda \sum_{i \in \mathcal{I}\setminus\{1\}} \sum_{t \in \mathcal{T}\setminus\{1\}} \|\boldsymbol{w}_i(t)\|^2 = \lambda \sum_{i \in \mathcal{I}\setminus\{1\}} \sum_{t \in \mathcal{T}\setminus\{1\}} \sum_{\ell \in \mathcal{S}} \sum_{s \in \mathcal{S}} e_{i,\ell}(t)\, e_{i,s}(t)\, \mathcal{K}_{i,\ell,s}(t) \tag{12}$$

(see [26] for the details), we can rewrite problem (10) as follows:

$$\left|\begin{array}{ll} \underset{\substack{a(t),\ e_{i,s}(t)\\ u_i(t),\ u_{i,s}(t)\\ v_s(t),\ x_{i,s}(t)\\ z_s(t)}}{\text{minimize}} & (1-\alpha)\sum_{t \in \mathcal{T}} \theta(t) \left(a(t) + \frac{1}{1-\beta} \sum_{s \in \mathcal{S}} P_s z_s(t) \right) - \alpha \sum_{t \in \mathcal{T}} \eta(t) \sum_{s \in \mathcal{S}} P_s v_s(t) \\ & \qquad + \lambda \sum_{i \in \mathcal{I}\setminus\{1\}} \sum_{t \in \mathcal{T}\setminus\{1\}} \sum_{\ell \in \mathcal{S}} \sum_{s \in \mathcal{S}} e_{i,\ell}(t)\, e_{i,s}(t)\, \mathcal{K}_{i,\ell,s}(t) \ \cdots (13.\mathrm{a}) \\ \text{subject to} & (10.\mathrm{b}), \ldots, (10.\mathrm{i}) \\ & u_{i,s}(t) = u_i(t) + \sum_{\ell \in \mathcal{S}} e_{i,\ell}(t)\, \mathcal{K}_{i,\ell,s}(t),\ i \in \mathcal{I} \setminus \{1\},\ s \in \mathcal{S},\ t \in \mathcal{T} \setminus \{1\}. \ \cdots (13.\mathrm{b}) \end{array}\right. \tag{13}$$

Note that the kernel function (11) constructs a Gram matrix, which is a positive semidefinite symmetric matrix. Therefore, problem (13) is a convex quadratic optimization.

Among the various kernel functions proposed in the literature (see, e.g., [23]), the most popular is the Gaussian kernel:

$$\mathcal{K}_{i,\ell,s}(t) = \exp\left(- \frac{\sum_{j \in \mathcal{I}} \sum_{k=1}^{t-1} \left(R_{j,\ell}(k) - R_{j,s}(k)\right)^2}{\sigma_{i,t}^2} \right), \tag{14}$$

where $\sigma_{i,t}$ are user-defined parameters. It is known that the Gaussian kernel corresponds to an inner product in an infinite-dimensional feature space (see, e.g., [23]). Therefore, solving problem (13) with the Gaussian kernel (14) is equivalent to solving problem (10) with infinite-dimensional feature vectors.

3 Dimensionality Reduction Technique

As indicated in the previous section, the use of an adequate kernel function leads to a convex optimization. Nevertheless, the kernel function (11) makes the problem structure worse, or more specifically, the associated coefficient matrix of the linear constraints becomes dense. As a result, problem (13) requires a long computation time in many cases. In this section, we present a dimensionality reduction technique for solving our multi-period portfolio optimization problem efficiently.

To begin with, let us rewrite the control policies (13. b) as follows:

$$\boldsymbol{u}^i(t) = u_i(t)\,\mathbf{1} + \boldsymbol{K}^i(t)\,\boldsymbol{e}^i(t),\ i \in \mathcal{I} \setminus \{1\},\ t \in \mathcal{T} \setminus \{1\}, \tag{15}$$

where

$$\boldsymbol{u}^i(t) := (u_{i,1}(t), u_{i,2}(t), \ldots, u_{i,S}(t))^\top \in \mathbb{R}^S,\ \mathbf{1} := (1, 1, \ldots, 1)^\top \in \mathbb{R}^S,$$

$$\boldsymbol{K}^i(t) := \begin{pmatrix} \mathcal{K}_{i,1,1}(t) & \mathcal{K}_{i,2,1}(t) & \cdots & \mathcal{K}_{i,S,1}(t) \\ \mathcal{K}_{i,1,2}(t) & \mathcal{K}_{i,2,2}(t) & \cdots & \mathcal{K}_{i,S,2}(t) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{K}_{i,1,S}(t) & \mathcal{K}_{i,2,S}(t) & \cdots & \mathcal{K}_{i,S,S}(t) \end{pmatrix} \in \mathbb{R}^{S\times S}, \tag{16}$$

$$\boldsymbol{e}^i(t) := (e_{i,1}(t), e_{i,2}(t), \ldots, e_{i,S}(t))^\top \in \mathbb{R}^S.$$

Recall that $\boldsymbol{K}^i(t)$ is a positive semidefinite symmetric matrix, which has only nonnegative eigenvalues.

Suppose that the matrix $\boldsymbol{K}^i(t)$ has $\bar{S}$ positive eigenvalues. Furthermore, let $\lambda_{i,1}(t) \ge \lambda_{i,2}(t) \ge \cdots \ge \lambda_{i,\bar{S}}(t) > 0$ be the positive eigenvalues of the matrix $\boldsymbol{K}^i(t)$ and $\boldsymbol{d}^i_1(t), \boldsymbol{d}^i_2(t), \ldots, \boldsymbol{d}^i_{\bar{S}}(t) \in \mathbb{R}^S$ be the associated normalized eigenvectors. Then, the matrix $\boldsymbol{D}^i(M;t)$ for a positive integer M $(\le \bar{S})$ is defined as

$$\boldsymbol{D}^i(M;t) := \left(\sqrt{\lambda_{i,1}(t)}\,\boldsymbol{d}^i_1(t), \sqrt{\lambda_{i,2}(t)}\,\boldsymbol{d}^i_2(t), \ldots, \sqrt{\lambda_{i,M}(t)}\,\boldsymbol{d}^i_M(t)\right) \in \mathbb{R}^{S\times M}, \tag{17}$$

so that $\boldsymbol{K}^i(t)$ can be decomposed as $\boldsymbol{K}^i(t) = \boldsymbol{D}^i(\bar{S};t)\,\boldsymbol{D}^i(\bar{S};t)^\top$.

When $M < \bar{S}$, the kernel-based control policy (15) can be approximated as follows:

$$\begin{aligned} \boldsymbol{u}^i(t) &= u_i(t)\,\mathbf{1} + \boldsymbol{D}^i(\bar{S};t)\,\boldsymbol{D}^i(\bar{S};t)^\top \boldsymbol{e}^i(t) \\ &\approx u_i(t)\,\mathbf{1} + \boldsymbol{D}^i(M;t)\,\boldsymbol{D}^i(M;t)^\top \boldsymbol{e}^i(t) \\ &= u_i(t)\,\mathbf{1} + \boldsymbol{D}^i(M;t)\,\boldsymbol{y}^i(t), \end{aligned}$$

where $\boldsymbol{y}^i(t) := \boldsymbol{D}^i(M;t)^\top \boldsymbol{e}^i(t) \in \mathbb{R}^M$.

Additionally, the regularization term in problem (13) is transformed as follows:

$$\begin{aligned}\sum_{\ell\in\mathcal{S}}\sum_{s\in\mathcal{S}} e_{i,\ell}(t)\, e_{i,s}(t)\, \mathcal{K}_{i,\ell,s}(t) &= \boldsymbol{e}^i(t)^\top \boldsymbol{K}^i(t)\, \boldsymbol{e}^i(t)\\ &= \boldsymbol{e}^i(t)^\top \boldsymbol{D}^i(\bar{S};t)\, \boldsymbol{D}^i(\bar{S};t)^\top \boldsymbol{e}^i(t)\\ &\approx \boldsymbol{e}^i(t)^\top \boldsymbol{D}^i(M;t)\, \boldsymbol{D}^i(M;t)^\top \boldsymbol{e}^i(t)\\ &= \boldsymbol{y}^i(t)^\top \boldsymbol{y}^i(t)\end{aligned}$$

Finally, by treating $\boldsymbol{y}^i(t) = (y_{i,1}(t), y_{i,2}(t), \ldots, y_{i,M}(t))^\top$ as new decision variables, problem (13) can be reduced to an approximation problem:

$$\left|\begin{array}{ll}\underset{\substack{a(t),\ u_i(t)\\ u_{i,s}(t),\ v_s(t)\\ x_{i,s}(t),\ y_{i,m}(t)\\ z_s(t)}}{\text{minimize}} & (1-\alpha)\sum_{t\in\mathcal{T}}\theta(t)\left(a(t)+\frac{1}{1-\beta}\sum_{s\in\mathcal{S}}P_s z_s(t)\right)-\alpha\sum_{t\in\mathcal{T}}\eta(t)\sum_{s\in\mathcal{S}}P_s v_s(t)\\ & \qquad +\lambda\sum_{i\in\mathcal{I}\setminus\{1\}}\sum_{t\in\mathcal{T}\setminus\{1\}}\sum_{m=1}^{M} y_{i,m}(t)^2 \quad\cdots(18.\mathrm{a})\\ \text{subject to} & (10.\mathrm{b}), ..., (10.\mathrm{i})\\ & u_{i,s}(t) = u_i(t) + \sum_{m=1}^{M} D_{i,s,m}(M;t)\, y_{i,m}(t),\\ & \qquad i\in\mathcal{I}\setminus\{1\},\ s\in\mathcal{S},\ t\in\mathcal{T}\setminus\{1\}, \quad\cdots(18.\mathrm{b})\end{array}\right. \tag{18}$$

where the (s,m)-th entry of $\boldsymbol{D}^i(M;t)$ is denoted by $D_{i,s,m}(M;t)$. We should also notice that reasonable performance can be attained by setting M to $0.02\times S$ in the context of the kernel-based support vector machine (see Yajima et al. [28]).

In what follows, we show another representation of $D_{i,s,m}(M;t)$, which is closely related to kernel PCA (see, e.g., Schölkopf and Smola [23]). It follows from the definition that

$$\boldsymbol{K}^i(t)\, \boldsymbol{d}^i_m(t) = \lambda_{i,m}(t)\, \boldsymbol{d}^i_m(t),$$

or equivalently,

$$\sum_{\ell\in\mathcal{S}} \mathcal{K}_{i,\ell,s}(t)\, d_{i,\ell,m}(t) = \lambda_{i,m}(t)\, d_{i,s,m}(t),\ \ s\in\mathcal{S}. \tag{19}$$

Accordingly, we have

$$\begin{aligned}D_{i,s,m}(M;t) &\overset{(17)}{=} \sqrt{\lambda_{i,m}(t)}\, d_{i,s,m}(t)\\ &\overset{(19)}{=} \sqrt{\lambda_{i,m}(t)}\cdot\frac{\sum_{\ell\in\mathcal{S}}\mathcal{K}_{i,\ell,s}(t)\, d_{i,\ell,m}(t)}{\lambda_{i,m}(t)}\\ &= \sum_{\ell\in\mathcal{S}}\frac{d_{i,\ell,m}(t)}{\sqrt{\lambda_{i,m}(t)}}\cdot\mathcal{K}_{i,\ell,s}(t).\end{aligned} \tag{20}$$

4 Numerical Experiments

The numerical results reported in this section demonstrate how our approach works. We used two datasets, 4FF and 3IND (see Appendix A for the details), and considered a planning horizon of five periods (i.e., $T = 5$). The initial holdings were set as $\bar{x}_1(0) := 100$ and $\bar{x}_i(0) := 0$ for $i \in \mathcal{I} \setminus \{1\}$. The lower limit, L_i, of the investment proportion was set to 0 for all $i \in \mathcal{I}$. The upper limit, U_i, of the investment proportion was set to 0.5 for $i \in \mathcal{I} \setminus \{1\}$, and U_1 was set to 0.2 to avoid over-investing in cash. The net cash flow, $C(t)$, was 0 for all $t \in \mathcal{T}$. The occurrence probability, P_s, was $1/S$ for all $s \in \mathcal{S}$ and the confidence level, β, was 0.9. The weights, $\theta(t)$ and $\eta(t)$, of the CVaR (5. a) and the expected portfolio value (5. b) were set as $\theta(T) = \eta(T) = 1$ and $\theta(t) = \eta(t) = 0$ for $t \in \mathcal{T} \setminus \{T\}$. The transaction costs, τ_i^{buy} and τ_i^{sell}, were equally set as $\tau_1^{\text{buy}} = \tau_1^{\text{sell}} = 0$ for cash and $\tau_i^{\text{buy}} = \tau_i^{\text{sell}} = \tau \in \{0, 0.005, 0.01\}$ for $i \in \mathcal{I} \setminus \{1\}$. We employed the Gaussian kernel (14) and set the associated parameter value as

$$\sigma_{i,t} := \sqrt{2(t-1)\frac{1}{S}\sum_{\ell \in \mathcal{S}} \left(r_\ell(t) - \frac{1}{S}\sum_{s \in \mathcal{S}} r_s(t) \right)^2}$$

on the basis of the standard deviation of $r_s(t) := \sqrt{\sum_{i \in \mathcal{I}} (R_{i,s}(t))^2}$.

4.1 Scenario generation

Along the lines of Cornuejols and Tütüncü (Section 16.5.1, [5]), to capture the serial dependence of stock returns, we used the vector autoregressive (VAR) model for scenario generation. Specifically, we randomly generated scenarios of the total returns, $R_{i,s}(t)$, by using the following model:

$$\boldsymbol{r}(t) = \boldsymbol{\delta} + \boldsymbol{\Psi}\, \boldsymbol{r}(t-1) + \boldsymbol{\varepsilon}(t), \tag{21}$$

where $\boldsymbol{r}(t) \in \mathbb{R}^{I-1}$ is the vector of the rate of returns, i.e., $R_{i,s}(t) - 1$, $i \in \mathcal{I} \setminus \{1\}$; the vector of intercepts $\boldsymbol{\delta} \in \mathbb{R}^{I-1}$ and the matrix of coefficients $\boldsymbol{\Psi} \in \mathbb{R}^{(I-1)\times(I-1)}$ are parameters to be estimated; and $\boldsymbol{\varepsilon}(t)$ is the vector of random errors, which is independently and identically distributed with respect to $t \in \mathcal{T}$. It is assumed that $\boldsymbol{\varepsilon}(t)$ follows a multivariate normal distribution with zero mean and variance-covariance matrix $\boldsymbol{\Sigma}$. Note that asset 1 was cash and $R_{1,s}(t)$ were accordingly set to 1 for all $s \in \mathcal{S}$ and $t \in \mathcal{T}$. The estimated parameter values of the VAR model (21) are shown in Appendix A

4.2 Models for comparison

The numerical experiments assessed the efficiency of the following models:

- Basic does not use control policies. Specifically, it is an optimization model (10. a),...,(10. i) with the non-anticipativity condition (i.e., all $u_{i,s}(t)$ are replaced by $u_i(t)$) and $\lambda = 0$.

- Linear uses linear control policies. Specifically, it is an optimization model (10. a),...,(10. i) with the following constraints:

$$u_{i,s}(t) = u_i(t) + \sum_{k=1}^{t-1}\sum_{j \in \mathcal{I}} w_{i,j}(k,t) \left(R_{j,s}(k) - \sum_{\ell \in \mathcal{S}} P_\ell R_{j,\ell}(k) \right), \ i \in \mathcal{I}\setminus\{1\},\ s \in \mathcal{S},\ t \in \mathcal{T}\setminus\{1\},$$

(22)

where $\lambda = 0$ and $w_{i,j}(k,t)$ are decision variables representing linear feedback from past total returns.

- KerDR(M) is our model (18) that uses kernel-based control policies with the dimensionality reduction technique. M represents the number of eigenvectors used in the model.
- Kernel is an optimization model (13) that uses kernel-based control policies, but not the dimensionality reduction technique.

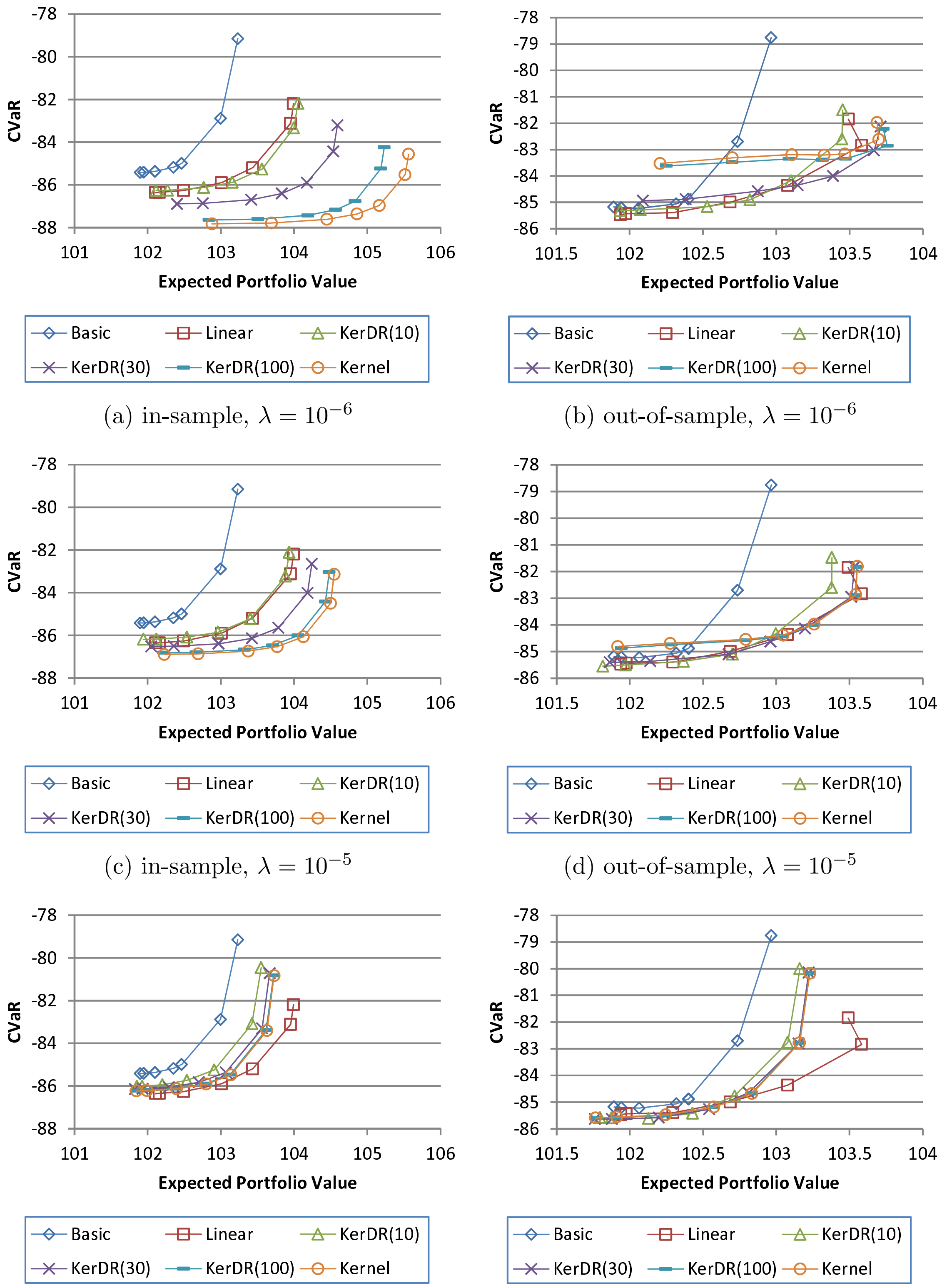
4.3 Investment performance

This subsection compares the in-sample and out-of-sample investment performance of each model (see Appendix B for a description of the performance evaluation methodology). Figures 3, 4 and 5 show the efficient frontiers of the solutions to the optimization models in Section 4.2. In these figures, the horizontal axis and vertical axis are the expected portfolio value (5. b) and CVaR (5. a), respectively. Each plot on the frontier corresponds to a different value of the trade-off parameter, $\alpha \in \{0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 0.99\}$.

Figures 3 and 4 show the investment performance in 300 scenarios (i.e., $S = 300$) when there are no transaction costs (i.e., $\tau = 0$). Figures 3 and 4 are associated with the 4FF dataset and 3IND dataset, respectively. The in-sample performance ((a), (c), (e)) of KerDR(M) became better as the number of used eigenvectors, M, got larger. In particular, the in-sample performance of KerDR(10) was similar to that of Linear, and the in-sample performance of KerDR(100) was similar to that of Kernel. In addition, we can see from these figures that the in-sample performances of KerDR(M) and Kernel improve as the value of the regularization parameter, λ, decreases. We can conclude that as far as the in-sample performance is concerned, Kernel with $\lambda = 10^{-6}$ performed the best (see Figures 3(a) and 4(a)).

As for the out-of-sample performance ((b), (d), (f)), Kernel with $\lambda = 10^{-6}$ got drastically worse (see Figures 3(b) and 4(b)). This poor result is because of overfitting; that is, the kernel-based control policy overfitted the scenarios used in the optimization problem and therefore was not effective in other scenarios. By contrast, the figures indicate that KerDR(10) and KerDR(30) generally had good out-of-sample performance regardless of the value of λ. Although the number of eigenvectors, M, can be up to S=300, these observations confirmed that only 10 or 30 eigenvectors were necessary to enable KerDR(M) to have an investment performance comparable to Kernel. Meanwhile, whereas Linear performed well in Figure 3, it performed poorly in Figure 4, and this deterioration was also caused by overfitting.

Now let us move on to Figure 5, which shows the out-of-sample investment performance in 1,000 scenarios (i.e., $S = 1{,}000$). Here, the regularization parameter, λ, was set to 10^{-5}, and the transaction cost was set to 0% (i.e., $\tau = 0$), 0.5% (i.e., $\tau = 0.005$) and 1% (i.e., $\tau = 0.01$). We can see that large transaction costs made the expected portfolio value significantly worse. For instance, although most solutions had the expected portfolio value of over 103 in



(a) in-sample, $\lambda = 10^{-6}$

(b) out-of-sample, $\lambda = 10^{-6}$

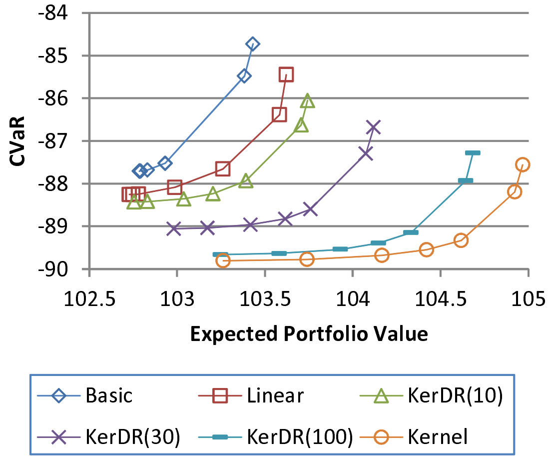
(c) in-sample, $\lambda = 10^{-5}$

(d) out-of-sample, $\lambda = 10^{-5}$

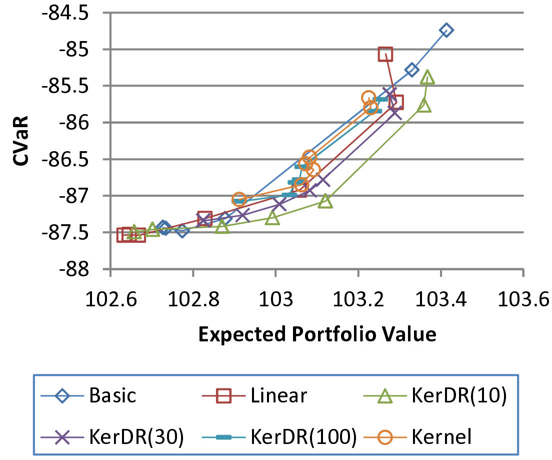
(e) in-sample, $\lambda = 10^{-4}$

(f) out-of-sample, $\lambda = 10^{-4}$

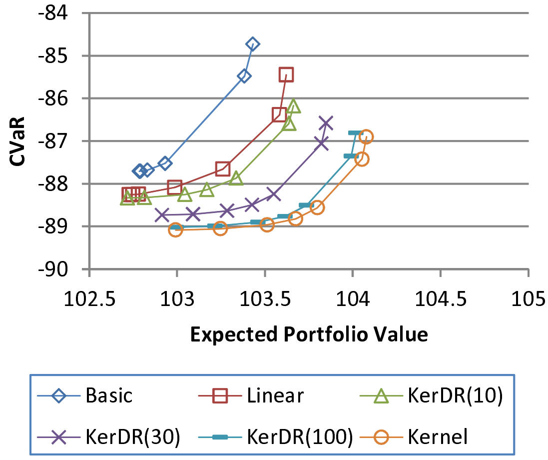
Figure 3: Efficient frontier (4FF dataset, $S = 300$, $\tau = 0$; see also Section 4.2)



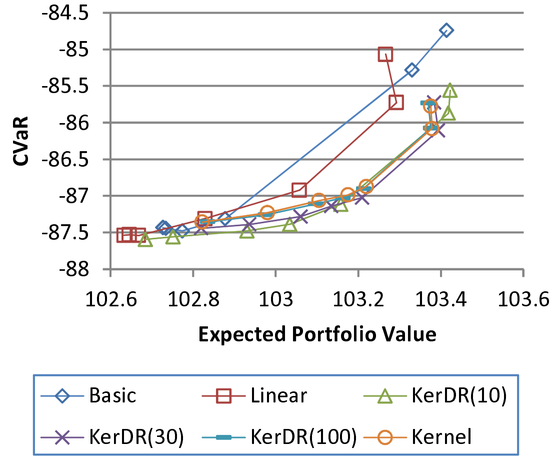
(a) in-sample, $\lambda = 10^{-6}$



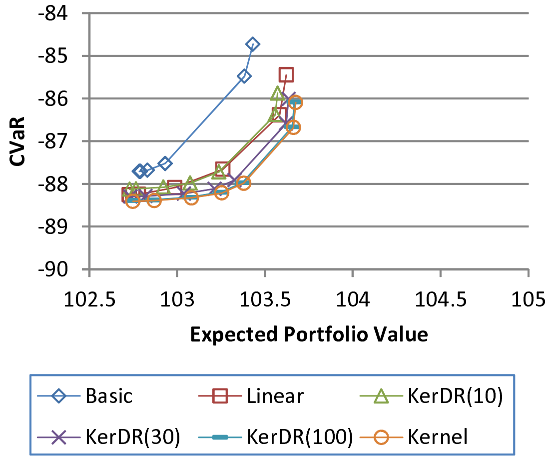
(b) out-of-sample, $\lambda = 10^{-6}$



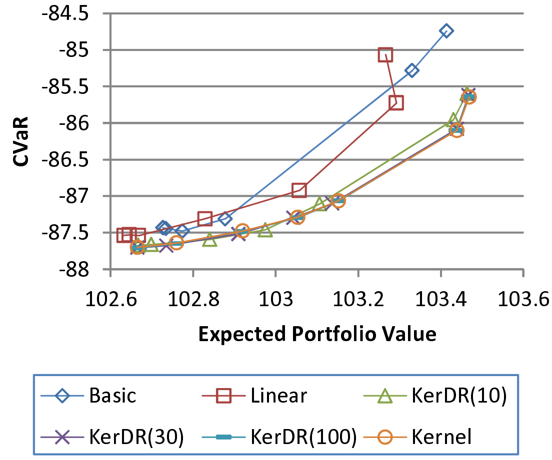
(c) in-sample, $\lambda = 10^{-5}$



(d) out-of-sample, $\lambda = 10^{-5}$

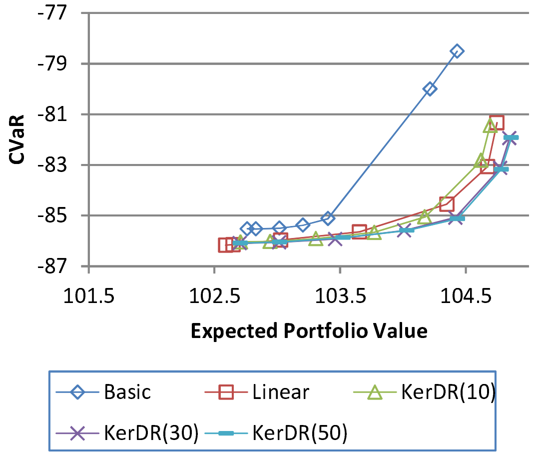


(e) in-sample, $\lambda = 10^{-4}$

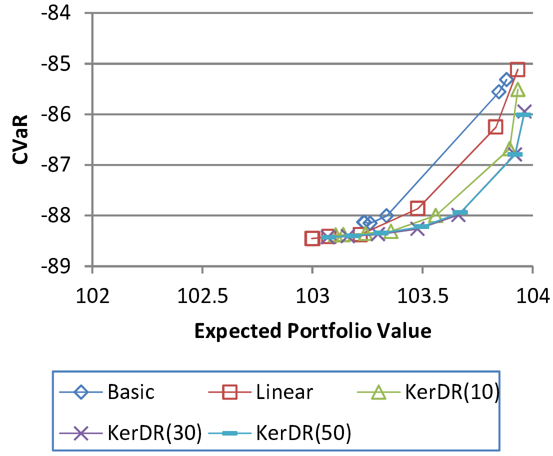


(f) out-of-sample, $\lambda = 10^{-4}$

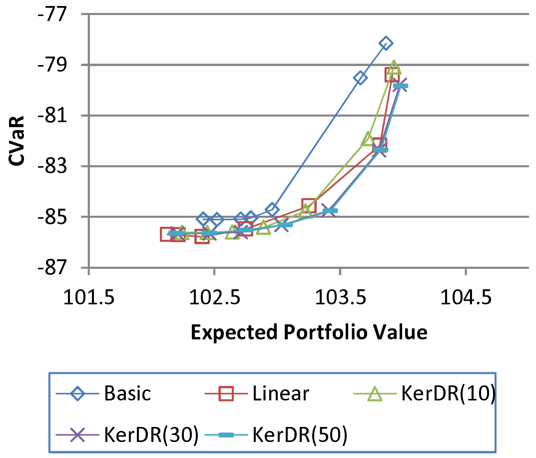
Figure 4: Efficient frontier (3IND dataset, $S = 300$, $\tau = 0$; see also Section 4.2)



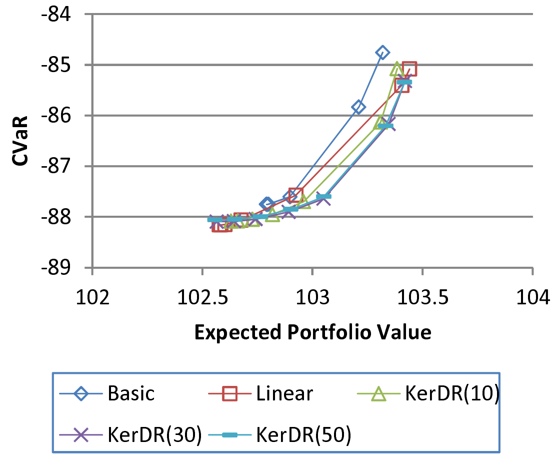
(a) 4FF dataset, $\tau = 0$



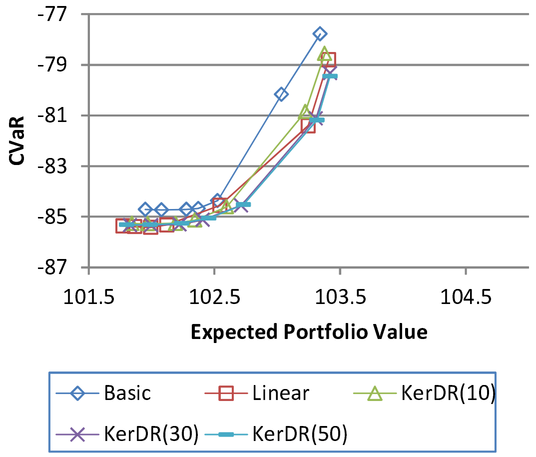
(b) 3IND dataset, $\tau = 0$



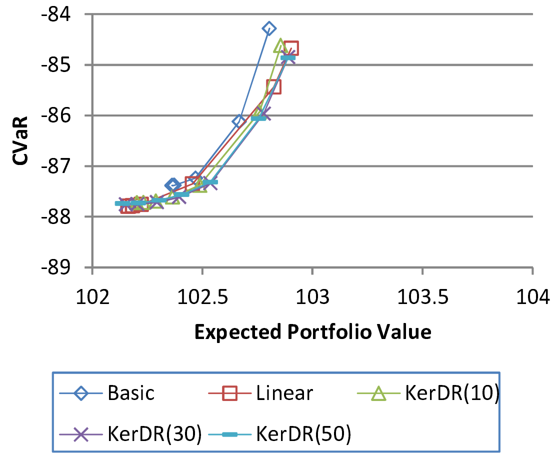
(c) 4FF dataset, $\tau = 0.005$



(d) 3IND dataset, $\tau = 0.005$



(e) 4FF dataset, $\tau = 0.01$



(f) 3IND dataset, $\tau = 0.01$

Figure 5: Efficient frontier (out-of-sample, $S = 1{,}000$, $\lambda = 10^{-5}$; see also Section 4.2)

Figure 5(b), no model obtained a value of 103 in Figure 5(f). Consequently, the difference in investment performance among these models became smaller as the transaction costs increased. Nevertheless, we found, even in the presence of a transaction cost, that Linear performed better than Basic, and that KerDR(30) and KerDR(50) performed better than Linear.

4.4 Computation time

Tables 1 and 2 report the average computation times of solving seven optimization problems for each value of the trade-off parameter, $\alpha \in \{0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 0.99\}$. All computations were conducted on a Windows 7 personal computer with a Core i5 Processor (2.40GHz) and 8GB memory. Also, FICO Xpress[1] was used to solve the optimization problems. In the case of 300 scenarios, the out-of-sample performances of KerDR(10) and KerDR(30) were better than or equal to Kernel, as shown in Figures 3 and 4. Moreover, KerDR(10) and KerDR(30) sharply reduced the computation time compared with Kernel (see Table 1).

In the case of 1,000 scenarios (see Table 2), on the other hand, the computation time of solving Kernel was over one hour; accordingly, there was no efficient frontier of solutions to Kernel in Figure 5. By contrast, it took less than one minute to solve KerDR(10), KerDR(30) and KerDR(50). We should also recall that these models had a relatively good investment performance.

Table 1: Average computation time [sec.] ($S = 300$, $\tau = 0$; see also Section 4.2)

	4FF dataset			3IND dataset		
Basic	0.5			0.4		
Linear	1.6			0.8		
	$\lambda = 10^{-6}$	$\lambda = 10^{-5}$	$\lambda = 10^{-4}$	$\lambda = 10^{-6}$	$\lambda = 10^{-5}$	$\lambda = 10^{-4}$
KerDR(10)	2.3	2.3	2.3	1.3	1.3	1.3
KerDR(30)	4.3	4.2	4.5	2.4	2.3	2.4
KerDR(100)	29.5	29.3	33.2	11.3	11.6	12.9
Kernel	31.1	29.0	32.8	13.8	15.2	15.3

4.5 Investment amounts

Figures 6 and 7 show the box plots of the investment amounts, $x_{i,s}(t)$, of each model. The plots display the distribution of $\{x_{i,s}(t) \mid s \in \mathcal{S}\}$ for each asset $i \in \mathcal{I}$ and each period $t \in \mathcal{T}$. These results show that the investment amounts of Linear and KerDR(30) had a lower dispersion in the presence of transaction costs ((d), (f)) than with no transaction costs ((c), (e)). In the presence of transaction costs, it costs a lot to actively rebalance a portfolio, and the investment

[1] `http://www.fico.com/en`

Table 2: Average computation time [sec.] ($S = 1{,}000$, $\lambda = 10^{-5}$; see also Section 4.2)

	4FF dataset			3IND dataset		
	$\tau = 0$	$\tau = 0.005$	$\tau = 0.01$	$\tau = 0$	$\tau = 0.005$	$\tau = 0.01$
Basic	2.7	3.1	2.7	2.0	2.4	2.7
Linear	9.1	23.2	20.7	5.6	11.4	12.5
KerDR(10)	9.4	14.7	15.8	5.1	11.8	12.0
KerDR(30)	20.2	28.8	29.9	9.9	21.5	21.6
KerDR(50)	43.4	60.0	61.1	17.8	43.6	41.8
Kernel	>3,600	>3,600	>3,600	>3,600	>3,600	>3,600

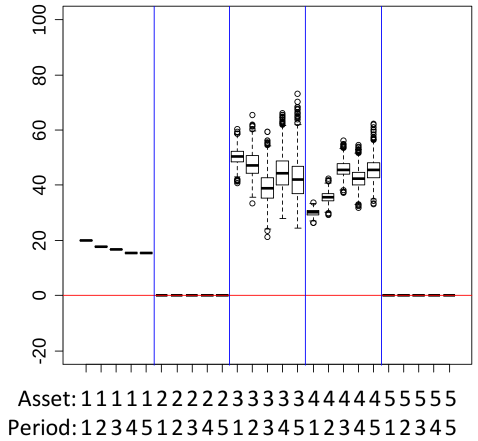
amounts accordingly fluctuate within a small range. When there are no transaction costs ((a), (c), (e)), the investment amounts of Linear and KerDR(30) are spread wider than those of Basic. Furthermore, KerDR(30) invested in assets that were not invested in by the other models (see assets 2 and 4 in Figure 6(b), (d), (f), and asset 4 in Figure 7). This is probably because the kernel-based control policy exploits to the full the time-series dependence of asset returns, and accordingly, can make a profit by investing in seemingly unprofitable assets (see also the estimated parameter values in Appendix A).

5 Conclusions

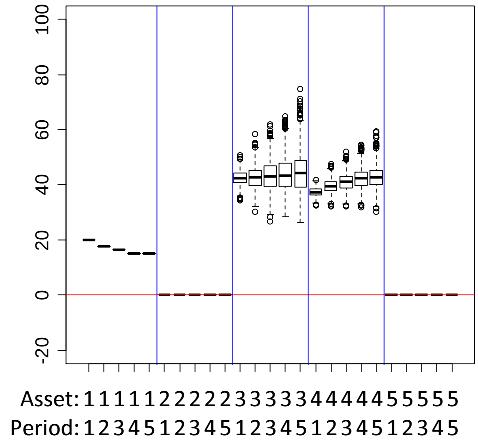
The present paper studied the kernel-based nonlinear control policies in a multi-period portfolio selection model. The dimensionality reduction technique was used therein to reduce the computational burden.

Numerical experiments were carried out to assess the computational advantages of our reduced optimization model and to compare the investment performance of our model with those of other models. The results show that our reduced optimization model could prevent the kernel-based control policy from overfitting. As a result, our model had high out-of-sample investment performance in comparison with the other models. Moreover, the results demonstrated that our reduced optimization model could decrease the computation time required for optimizing the kernel-based control policy. Additionally, the investment amounts of each model illustrated some features of investment strategies based on control policies.

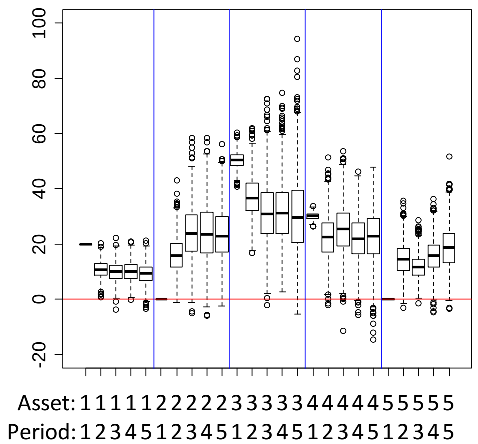
There is, however, much work left to be done. For practical purposes, we need to solve portfolio selection problems involving hundreds or thousands of investable assets. A large number of assets requires a larger number of scenarios for preventing overfitting; thus, a future direction of study will be to devise a method for solving such large-scale problems.



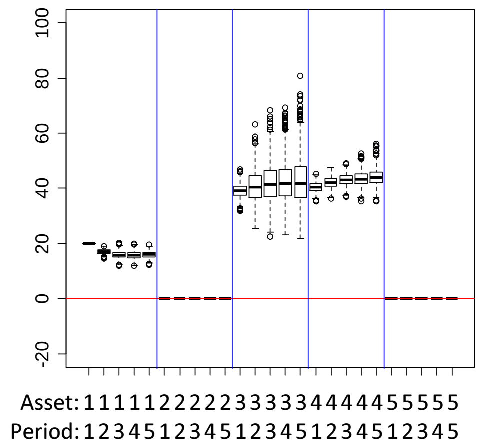
(a) Basic, $\tau = 0$



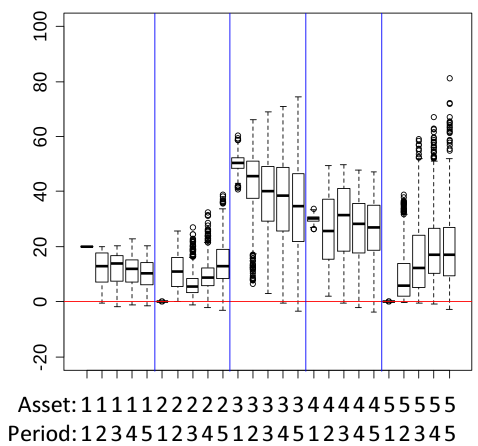
(b) Basic, $\tau = 0.01$



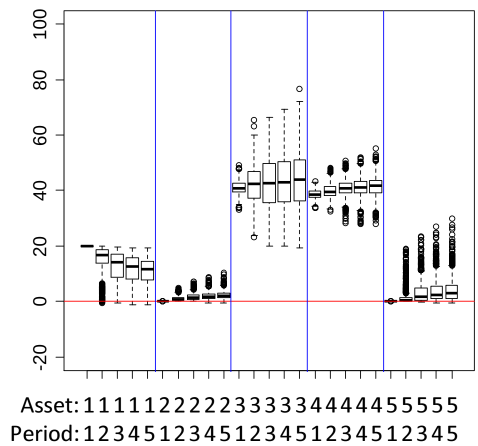
(c) Linear, $\tau = 0$



(d) Linear, $\tau = 0.01$

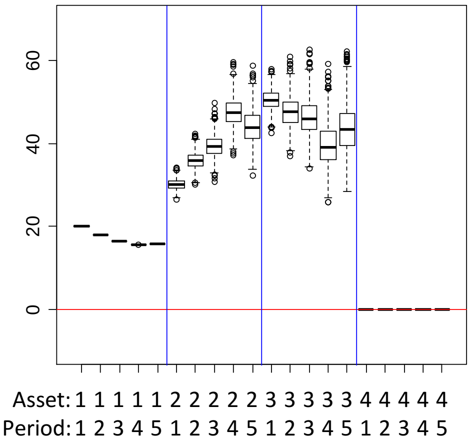


(e) KerDR(30), $\tau = 0$

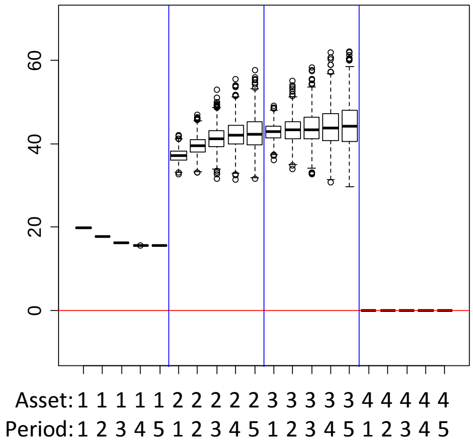


(f) KerDR(30), $\tau = 0.01$

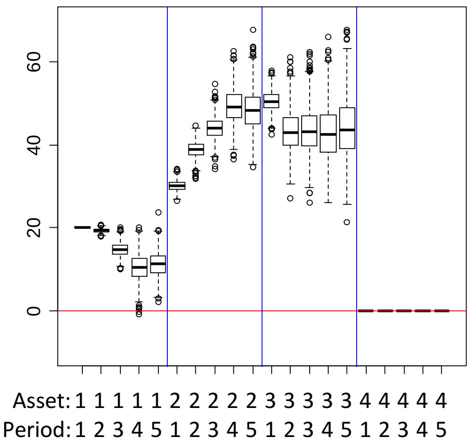
Figure 6: Investment amounts (4FF dataset, $S = 1{,}000$, $\lambda = 10^{-5}$, $\alpha = 0.7$; see also Section 4.2)



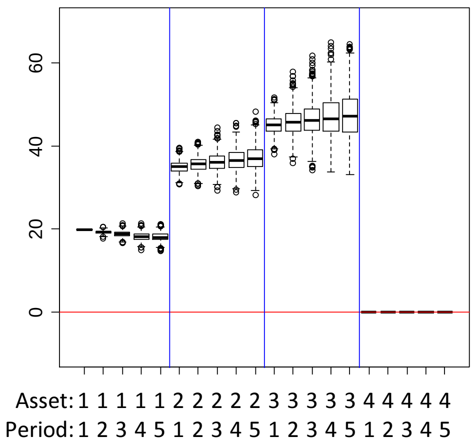
(a) Basic, $\tau = 0$



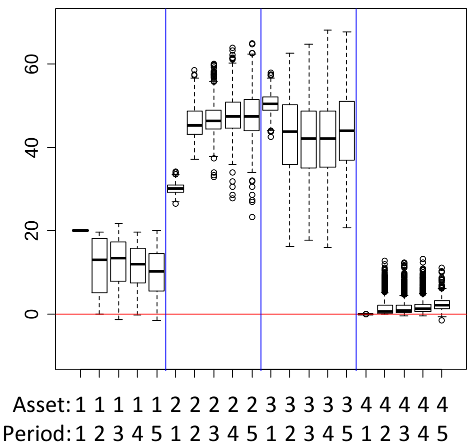
(b) Basic, $\tau = 0.01$



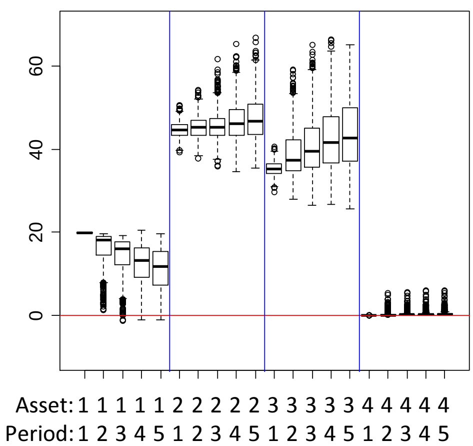
(c) Linear, $\tau = 0$



(d) Linear, $\tau = 0.01$



(e) KerDR(30), $\tau = 0$



(f) KerDR(30), $\tau = 0.01$

Figure 7: Investment amounts (3IND dataset, $S = 1{,}000$, $\lambda = 10^{-5}$, $\alpha = 0.7$; see also Section 4.2)

Appendix

A Datasets and Estimated Parameter Values

Tables A.1 and A.2 show the details of the datasets used in numerical experiments of Section 4. The estimated parameter values of the VAR model (21) were

$$\boldsymbol{\delta} = \begin{pmatrix} 0.0057 \\ 0.0076 \\ 0.0040 \\ 0.0026 \end{pmatrix}, \ \boldsymbol{\Psi} = \begin{pmatrix} -0.11 & -0.34 & 0.86 & 0.08 \\ -0.37 & 0.02 & 0.83 & 0.09 \\ -0.06 & -0.14 & 0.46 & -0.06 \\ -0.28 & -0.03 & 0.69 & 0.08 \end{pmatrix}, \ \boldsymbol{\Sigma} = \begin{pmatrix} 0.0037 & 0.0035 & 0.0023 & 0.0029 \\ 0.0035 & 0.0039 & 0.0021 & 0.0031 \\ 0.0023 & 0.0021 & 0.0018 & 0.0020 \\ 0.0029 & 0.0031 & 0.0020 & 0.0031 \end{pmatrix}$$

Table A.1 : Details of the 4FF dataset

Source	6 Portfolios Formed on Size and Book-to-Market (2 × 3) from K.R. French's website[a]
Time Period	monthly data from 2002 to 2011
Asset 1	Cash
Asset 2	Small Value: a portfolio composed of small-sized and value stocks
Asset 3	Small Growth: a portfolio composed of small-sized and growth stocks
Asset 4	Large Value: a portfolio composed of large-sized and value stocks
Asset 5	Large Growth: a portfolio composed of large-sized and growth stocks

Table A.2 : Details of the 3IND dataset

Source	5 Industry Portfolios from K.R. French's website[a]
Time Period	monthly data from 2002 to 2011
Asset 1	Cash
Asset 2	Cnsmr: a portfolio composed of the following industries: Consumer Durables, NonDurables, Wholesale, Retail, and Some Services (Laundries, Repair Shops)
Asset 3	Manuf: a portfolio composed of the following industries: Manufacturing, Energy, and Utilities
Asset 4	HiTec: a portfolio composed of the following industries: Business Equipment, Telephone and Television Transmission

[a] `http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/index.html`

in the 4FF dataset, and

$$\boldsymbol{\delta} = \begin{pmatrix} 0.0056 \\ 0.0080 \\ 0.0043 \end{pmatrix}, \ \boldsymbol{\Psi} = \begin{pmatrix} 0.09 & -0.08 & 0.11 \\ 0.28 & -0.13 & 0.09 \\ 0.04 & -0.14 & 0.17 \end{pmatrix}, \ \boldsymbol{\Sigma} = \begin{pmatrix} 0.0016 & 0.0015 & 0.0020 \\ 0.0015 & 0.0024 & 0.0023 \\ 0.0020 & 0.0023 & 0.0038 \end{pmatrix}$$

in the 3IND dataset.

B Performance Evaluation Methodology

We generated two different sets of scenarios, i.e., scenario set A: $\{R_{i,s}(t) \mid i \in \mathcal{I}, \ s \in \mathcal{S}^{\mathrm{A}}, \ t \in \mathcal{T}\}$ and scenario set B: $\{R_{i,s}(t) \mid i \in \mathcal{I}, \ s \in \mathcal{S}^{\mathrm{B}}, \ t \in \mathcal{T}\}$, where $|\mathcal{S}^{\mathrm{A}}| = |\mathcal{S}^{\mathrm{B}}| = S$ and $\mathcal{S}^{\mathrm{A}} \cap \mathcal{S}^{\mathrm{B}} = \emptyset$ The optimal solutions to the problems in Section 4.2 are denoted by $u_i^{\mathrm{A}}(t)$, $w_{i,j}^{\mathrm{A}}(k,t)$, $e_{i,s}^{\mathrm{A}}(t)$ and $y_{i,m}^{\mathrm{A}}(t)$ for scenario set A; and by $u_i^{\mathrm{B}}(t)$, $w_{i,j}^{\mathrm{B}}(k,t)$, $e_{i,s}^{\mathrm{B}}(t)$ and $y_{i,m}^{\mathrm{B}}(t)$ for scenario set B.

To assess in-sample performance, the solutions were evaluated on the basis of the scenario set that was used to compute them. Specifically, we first calculated the CVaR (5. a) and the expected portfolio value (5. b) by using $u_i^{\mathrm{A}}(t)$, $w_{i,j}^{\mathrm{A}}(k,t)$, $e_{i,s}^{\mathrm{A}}(t)$, $y_{i,m}^{\mathrm{A}}(t)$ and scenario set A. Likewise, we then calculated them by using $u_i^{\mathrm{B}}(t)$, $w_{i,j}^{\mathrm{B}}(k,t)$, $e_{i,s}^{\mathrm{B}}(t)$, $y_{i,m}^{\mathrm{B}}(t)$ and scenario set B. We refer to these average value as the in-sample performance.

To assess out-of-sample performance, the solutions were evaluated on the basis of the scenario set that was not used to compute them. To describe a way to assess out-of-sample performance, we shall first use $u_i^{\mathrm{A}}(t)$, $w_{i,j}^{\mathrm{A}}(k,t)$, $e_{i,s}^{\mathrm{A}}(t)$, $y_{i,m}^{\mathrm{A}}(t)$ and scenario set B. In the case of Basic, the investment performance can be measured by means of $u_i^{\mathrm{A}}(t)$ and scenario set B. In other cases, we first calculate the adjustments at the beginning of the first period as

$$\hat{u}_i^{\mathrm{B}}(1) := u_i^{\mathrm{A}}(1), \ i \in \mathcal{I}.$$

Next, we calculate the adjustments of assets $i \in \mathcal{I} \setminus \{1\}$ in subsequent periods. Specifically, we use (22) as

$$\hat{u}_{i,s}^{\mathrm{B}}(t) := u_i^{\mathrm{A}}(t) + \sum_{k=1}^{t-1} \sum_{j \in \mathcal{I}} w_{i,j}^{\mathrm{A}}(k,t) \left(R_{j,s}(k) - \sum_{\ell \in \mathcal{S}^{\mathrm{A}}} P_\ell R_{j,\ell}(k) \right), \ i \in \mathcal{I} \setminus \{1\}, \ s \in \mathcal{S}^{\mathrm{B}}, \ t \in \mathcal{T} \setminus \{1\}$$

in the case of Linear, use (18. b) and (20) as

$$\hat{u}_{i,s}^{\mathrm{B}}(t) := u_i^{\mathrm{A}}(t) + \sum_{m=1}^{M} D_{i,s,m}^{\mathrm{A}}(M;t)\, y_{i,m}^{\mathrm{A}}(t), \ i \in \mathcal{I} \setminus \{1\}, \ s \in \mathcal{S}^{\mathrm{B}}, \ t \in \mathcal{T} \setminus \{1\},$$

where $D_{i,s,m}^{\mathrm{A}}(M;t) := \sum_{\ell \in \mathcal{S}^{\mathrm{A}}} \frac{d_{i,\ell,m}(t)}{\sqrt{\lambda_{i,m}(t)}} \mathcal{K}_{i,\ell,s}(t)$ in the case of KerDR(M), and use (13. b) as

$$\hat{u}_{i,s}^{\mathrm{B}}(t) := u_i^{\mathrm{A}}(t) + \sum_{\ell \in \mathcal{S}^{\mathrm{A}}} e_{i,\ell}^{\mathrm{A}}(t)\, \mathcal{K}_{i,\ell,s}(t), \ i \in \mathcal{I} \setminus \{1\}, \ s \in \mathcal{S}^{\mathrm{B}}, \ t \in \mathcal{T} \setminus \{1\}$$

in the case of Kernel. After that, we calculate the remaining adjustments of cash from (10. g) as follows:

$$\hat{u}^{\mathrm{B}}_{1,s}(t) = C(t) - \sum_{i \in \mathcal{I} \setminus \{1\}} \gamma_i(\hat{u}^{\mathrm{B}}_{i,s}(t)) - \sum_{i \in \mathcal{I} \setminus \{1\}} \hat{u}^{\mathrm{B}}_{i,s}(t),\ s \in \mathcal{S}^{\mathrm{B}},\ t \in \mathcal{T} \setminus \{1\}.$$

Finally, investment performance can be measured by means of $\hat{u}^{\mathrm{B}}_{i,s}(t)$ and scenario set B. We performed the same calculation for $u^{\mathrm{B}}_i(t)$, $w^{\mathrm{B}}_{i,j}(k,t)$, $e^{\mathrm{B}}_{i,s}(t)$, $y^{\mathrm{B}}_{i,m}(t)$ and scenario set A, and referred to these average values as the out-of-sample performance.

Acknowledgments

The first author was supported by JSPS KAKENHI Grant Number 23810007. The second author was supported by JSPS KAKENHI Grant Number 23710176.

References

[1] P.H. Algoet and T.M. Cover, "Asymptotic Optimality and Asymptotic Equipartition Properties of Log-Optimum Investment," *The Annals of Probability*, Vol.16, No.2, pp.876-898 (1988).

[2] G.C. Calafiore, "Multi-Period Portfolio Optimization with Linear Control Policies," *Automatica*, Vol.44, No.10, pp.2463–2473 (2008).

[3] G.C. Calafiore, "An Affine Control Method for Optimal Dynamic Asset Allocation with Transaction Costs," *SIAM Journal on Control and Optimization*, Vol.48, No.4, pp.2254–2274 (2009).

[4] G.C. Calafiore and M.C. Campi, "On Two-Stage Portfolio Allocation Problems with Affine Recourse," *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference* (2005).

[5] G. Cornuejols and R. Tütüncü, *Optimization Methods in Finance* (Cambridge University Press, 2007).

[6] V. DeMiguel, F.J. Nogales, and R. Uppal, "Stock Return Serial Dependence and Out-of-Sample Portfolio Performance," AFA 2011 Denver Meetings Paper (2010).

[7] S.-E. Fleten, K. Høyland and S.W. Wallace, "The Performance of Stochastic Dynamic and Fixed Mix Portfolio Models," *European Journal of Operational Research*, Vol.140, No.1, pp.37–49 (2002).

[8] N. Hibiki, "A Hybrid Simulation/Tree Stochastic Optimization Model for Dynamic Asset Allocation," In B. Scherer (ed.), *Asset and Liability Management Tools: A Handbook for Best Practice*, pp.269–294 (Risk Books, 2003).

[9] N. Hibiki, "Multi-Period Stochastic Optimization Models for Dynamic Asset Allocation," *Journal of Banking and Finance*, Vol.30, No.2, pp.365–390 (2006).

[10] G. Infanger, "Dynamic Asset Allocation Strategies Using a Stochastic Dynamic Programming Approach," In S.A. Zenios & W.T. Ziemba (eds.), *Handbook of Asset and Liability Management*, pp.199–251 (Elsevier, Amsterdam, 2006).

[11] A.M. Jade, B. Srikanth, V.K. Jayaraman, B.D. Kulkarni, J.P. Jog, and L. Priya, "Feature Extraction and Denoising Using Kernel PCA," *Chemical Engineering Science*, Vol.58, No.19, pp.4441–4448 (2003).

[12] N. Jegadeesh and S. Titman, "Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency," *Journal of Finance*, Vol.48, No.1, pp.65–91 (1993).

[13] A. Lo and A.C. MacKinlay: "When are Contrarian Profits due to Stock Market Overreaction?," *The Review of Financial Studies*, Vol.3, No.2, pp.175–205 (1990).

[14] H. Markowitz, "Portfolio Selection," *Journal of Finance*, Vol.7, No.1, pp.77–91 (1952).

[15] C.D. Maranas, I.P. Androulakis, C.A. Floudas, A.J. Berger and J.M. Mulvey, "Solving Long-Term Financial Planning Problems via Global Optimization," *Journal of Economic Dynamics & Control*, Vol.21, No.8-9, pp.1405-1425 (1997).

[16] C.C. Moallemi and M. Sağlam, "Dynamic Portfolio Choice with Linear Rebalancing Rules," Available at SSRN 2011605 (2012).

[17] J.M. Mulvey, W.R. Pauling, and R.E. Madey, "Advantages of Multiperiod Portfolio Models," *The Journal of Portfolio Management*, Vol.29, No.2, pp.35–45 (2003).

[18] G.Ch. Pflug, "Some Remarks on the Value-at-Risk and the Conditional Value-at-Risk," In S. Uryasev (ed.), *Probabilistic Constrained Optimization: Methodology and applications*, pp.272–281 (Kluwer Academic Publishers, Dordrecht, 2000).

[19] J.A. Primbs and C.H. Sung, "A Stochastic Receding Horizon Control Approach to Constrained Index Tracking," *Asia-Pacific Financial Markets*, Vol.15, No.1, pp.3–24 (2008).

[20] R. Rosipal, M. Girolami, L.J. Trejo, and A. Cichocki, "Kernel PCA for Feature Extraction and De-Noising in Nonlinear Regression," *Neural Computing & Applications*, Vol.10, No.3, pp.231–243 (2001).

[21] R.T. Rockafellar and S. Uryasev, "Conditional Value-at-Risk for General Loss Distributions," *Journal of Banking & Finance*, Vol.26, No.7, pp.1443–1471 (2002).

[22] J. Skaf and S. Boyd, "Multi-Period Portfolio Optimization with Constraints and Transaction Costs," Working Paper, Stanford University (2008).

[23] B. Schölkopf and A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press (2001).

[24] M.C. Steinbach, "Markowitz Revisited: Mean-Variance Models in Financial Portfolio Analysis," *SIAM Review*, Vol.43, No.1, pp.31–85 (2001).

[25] Y. Takano and J. Gotoh, "Constant Rebalanced Portfolio Optimization under Nonlinear Transaction Costs," *Asia-Pacific Financial Markets*, Vol.18, No.2, pp.191–211 (2011).

[26] Y. Takano and J. Gotoh, "A Nonlinear Control Policy Using Kernel Method for Dynamic Asset Allocation," *Journal of the Operations Research Society of Japan*, Vol.54, No.4, pp.201–218 (2011).

[27] Y. Takano and R. Sotirov, "A Polynomial Optimization Approach to Constant Rebalanced Portfolio Selection," *Computational Optimization and Applications*, Vol.52, No.3, pp.645–666 (2012). .

[28] Y. Yajima, H. Ohi, and M. Mori, "Extracting Feature Subspace for Kernel Based Linear Programming Support Vector Machines," *Journal of the Operations Research Society of Japan*, Vol.46, No.4, pp.395–408 (2003).