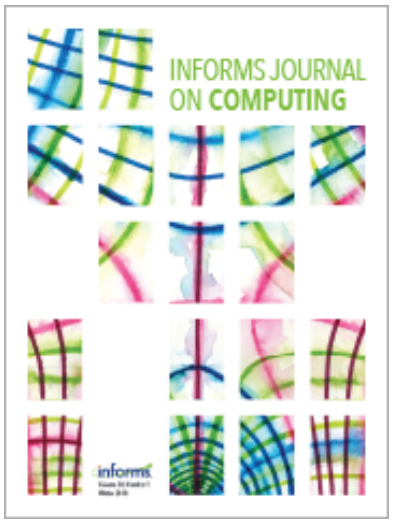


INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Portfolio Optimization with Nonparametric Value at Risk: A Block Coordinate Descent Method

Xueting Cui, Xiaoling Sun, Shushang Zhu, Rujun Jiang, Duan Li

To cite this article:
Xueting Cui, Xiaoling Sun, Shushang Zhu, Rujun Jiang, Duan Li (2018) Portfolio Optimization with Nonparametric Value at Risk: A Block Coordinate Descent Method. INFORMS Journal on Computing 30(3):454-471. https://doi.org/10.1287/ijoc.2017.0793

http://pubsonline.informs.org/journal/ijoc/

INFORMS JOURNAL ON COMPUTING
Vol. 30, No. 3, Summer 2018, pp. 454–471
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

Portfolio Optimization with Nonparametric Value at Risk: A Block Coordinate Descent Method

Xueting Cui,[a] Xiaoling Sun,[b,*] Shushang Zhu,[c] Rujun Jiang,[d] Duan Li[e]

[a] School of Mathematics, Shanghai University of Finance and Economics, Shanghai 200433, P. R. China; [b] Department of Management Science, School of Management, Fudan University, Shanghai 200433, P. R. China; [c] Department of Finance and Investment, Sun Yat-Sen Business School, Sun Yat-Sen University, Guangzhou 510275, P. R. China; [d] School of Data Science, Fudan University, Shanghai 200433, P. R. China; [e] Department of Management Sciences, College of Business, City University of Hong Kong, Kowloon, Hong Kong

[*] Deceased

Contact: cui.xueting@shufe.edu.cn (XC); zhuss@mail.sysu.edu.cn (SZ); rjjiang@fudan.edu.cn (RJ); dli226@cityu.edu.hk (DL)

Received: October 1, 2014
Revised: January 11, 2016; November 29, 2016; February 1, 2017; March 20, 2017; April 4, 2017; April 13, 2017; May 24, 2017; June 30, 2017
Accepted: September 15, 2017
Published Online: September 11, 2018

https://doi.org/10.1287/ijoc.2017.0793

Abstract. In this paper, we investigate a portfolio optimization methodology using nonparametric value at risk (VaR). In particular, we adopt kernel VaR and quadratic VaR as risk measures. As the resulting models are nonconvex and nonsmooth optimization problems, albeit with some special structures, we propose some specially devised block coordinate descent (BCD) methods for finding approximate or local optimal solutions. Computational results show that the BCD methods are efficient for finding local solutions with good quality and they compare favorably with the branch-and-bound method-based global optimal solution procedures. From the simulation test and empirical analysis that we carry out, we are able to conclude that the mean-VaR models using kernel VaR and quadratic VaR are more robust compared to those using historical VaR or parametric VaR under the normal distribution assumption, especially when the information of the return distribution is limited.

History: Accepted by Karen Aardal, Area Editor for Design and Analysis of Algorithms.
Funding: This research is supported partially by the National Natural Science Foundation of China [Grants 71471180, 11371103, and 71501122]; by the National Natural Science Foundation of China and the Research Grants Council of Hong Kong Joint Research Scheme [Grant 71061160506]; and by the University Grants Council of Hong Kong [Grants 414513 and 14204514].
Supplemental Material: The online supplement is available at https://doi.org/10.1287/ijoc.2017.0793.

Keywords: portfolio selection • nonparametric VaR • kernel • BCD method

1. Introduction

Value at risk (VaR) has been a popular and powerful analytical tool in financial risk management and has attracted much attention in the literature of portfolio selection as a viable risk measure (see, e.g., J. P. Morgan 1996, Duffie and Pan 1997, Linsmeier and Pearson 2000, Jorion 2007). By definition, VaR refers to the maximum potential loss in portfolio value under a specific confidence level, that is, VaR is a quantile of the loss distribution. Compared with the standard deviation and the mean absolute deviation, VaR is a downside risk measure that is more preferable when the underlying return distribution is asymmetric or heavy tailed. Moreover, the adoption of VaR as a risk measure has been a regulatory obligation in Basel Accord II.

While VaR is intuitive in risk management practice and simple in its definition, it has several notorious limitations and drawbacks such as its insensitivity to the magnitude of losses beyond VaR, nonqualification as a coherent risk measure, and its nonconvexity with respect to the portfolio weights (Artzner et al. 1999, Mausser and Rosen 1999), which result in computational difficulties. On the other hand, conditional value at risk (CVaR) or expected shortfall (ES) measures the conditional expectation of losses beyond VaR. Being a coherent risk measure, CVaR is theoretically attractive and may partly resolve the shortcomings of VaR (see Rockafellar and Uryasev 2000, 2002). Lim et al. (2011), however, showed that the portfolios obtained from data-driven mean-CVaR models are unreliable because of estimation errors of CVaR. Heyde and Kou (2004) and Kou et al. (2013) also pointed out that CVaR, as a risk measure, is not robust with respect to the underlying models and data; they further argued that VaR is a more suitable risk measure for trading book capital requirements. Therefore, VaR is still considered to be a very useful risk measure in financial portfolio optimization.

A crucial issue in VaR-based portfolio selection models is how to estimate VaR values of portfolio as a function of portfolio weights. The estimation of VaR is closely related to the quantile estimation or tail estimation. The estimation methods of VaR in the literature can be classified into two categories: parametric methods and nonparametric methods. Parametric estimation methods are based on certain distribution assumptions

of the asset return, which are popular among the practitioners in risk management (J. P. Morgan 1996). For instance, if the asset return follows a normal distribution, then the VaR-based portfolio selection model can be formulated as a second-order cone program, which is polynomially solvable using interior-point methods. Alexander and Baptista (2004) studied the influence of VaR and CVaR constraints to the mean-variance efficient frontiers under the assumption of normal distribution of asset returns. Bonami and Lejeune (2009) presented an efficient branch-and-bound method to solve the portfolio selection problem with discrete variables and VaR constraint under a normal distribution assumption. Cui et al. (2013) investigated nonlinear portfolio selection using approximate parametric VaR based on the first- and second-order approximations of VaR, where the underlying factors of returns are assumed to follow a normal distribution, and showed that the portfolio selection models using these parametric VaR approximations can be reformulated as second-order cone programs.

Among various nonparametric estimations of VaR, *historical VaR*, which is estimated by the empirical quantile of the portfolio return, has been widely used as a simple nonparametric estimator of VaR. The portfolio selection problem based on historical VaR can be transformed into a mixed-integer binary programming problem by introducing logical variables and a group of constraints with "big-M" coefficients (see Benati and Rizzi 2007). Qiu et al. (2014) introduced a big-M coefficient strengthening scheme to improve the lower bounds generated from the continuous relaxation of the mixed-integer binary programming formulation. To avoid the big-M constraints, which are more likely to generate weak lower bounds, Luedtke (2014) proposed a branch-and-cut decomposition algorithm for the chance-constrained programming problems. Using a smoothing technique to approximate the historical VaR, Gaivoronski and Pflug (2005) proposed a solution method to generate a suboptimal portfolio. Wen et al. (2013) proposed an alternative direction method to solve a portfolio selection problem where historical VaR and CVaR are combined as the risk measure. While historical VaR is simple to calculate, it suffers from the lack of tail information of asset returns and thus is very sensitive to the confidence level and the portfolio allocation; the tail information is usually hard to obtain as the extreme samples in the tail part are rare. As a remedy of these deficiencies, Parzen (1979) proposed a *kernel VaR* estimator, which is a nonparametric quantile estimator by averaging over the values of empirical quantiles in a neighborhood of the considered confidence level. More generally, kernel quantile estimators are also discussed in Sheather and Marron (1990). Butler and Schachter (1998) investigated nonparametric estimation of VaR by combining kernel estimation with historical simulation. Cheng and Peng (2002) gave a local quadratic estimator by minimizing the average square error of a quadratic approximation to the empirical quantile estimator. The resulting VaR estimator is called *quadratic VaR* and can be expressed as a weighted sum of empirical quantiles. As alternative nonparametric VaR estimators, kernel VaR and quadratic VaR have been shown to be more robust than historical VaR (see Butler and Schachter 1998, Cheng and Peng 2002, Chang et al. 2003, Chen and Tang 2005). Yao et al. (2013) proposed a nonparametric CVaR-based portfolio selection model, which is shown to be an easily solvable convex program.

In this paper, we focus on portfolio selection models where nonparametric VaR is adopted as the risk measure. In particular, we discuss mean-nonparametric VaR portfolio selection problems using kernel VaR or quadratic VaR. To the best of our knowledge, this is the first attempt to integrate these nonparametric VaR estimators in the portfolio optimization models. Because of the nonconvex nature of these nonparametric VaR estimators, these models are nonconvex optimization problems and are in general NP-hard. The contribution of this paper is twofold. First, we propose a block coordinate descent (BCD) method for solving the mean-nonparametric VaR portfolio selection problems. By exploiting the special structure of the problems, the method alternatively solves two tractable subproblems at each iteration and generates a sequence of approximate solutions. We prove that the proposed BCD method converges to a first-order stationary point of the problem. Computational results suggest that the BCD method is capable of finding good-quality solutions and compares favorably with the branch-and-bound method-based global optimization procedure that solves a mixed-integer program (MIP) reformulation of the problem. Second, we carry out both simulation analysis and an empirical study to compare the performance of portfolios generated by the mean-nonparametric VaR portfolio models using different nonparametric VaR estimators. The results show that the portfolio selection models based on kernel VaR and quadratic VaR are promising in generating robust portfolios.

The rest of this paper is organized as follows. In Section 2, we first introduce the definitions of nonparametric VaR estimators including the kernel VaR estimator and the quadratic VaR estimator. We then establish the portfolio selection models based on the nonparametric VaR estimators. In Section 3, we present block coordinate descent methods for finding a local optimal solution of the nonconvex optimization problems resulting from the nonparametric VaR-based portfolio selection models and analyze the convergence of these methods. We present computational results in Section 4 to demonstrate the effectiveness of the proposed block

coordinate descent methods. We then carry out simulation analysis in Section 5 to check the robustness of nonparametric VaR-based models. We also conduct simulation analysis and empirical studies on the performance of the nonparametric VaR-based portfolio selection models in Section 6. Finally, we give some concluding remarks in Section 7.

All the data used in Sections 4–6 can be downloaded from http://www.se.cuhk.edu.hk/~dli under the item of "Data set for 'Portfolio optimization with nonparametric Value-at-Risk: A block coordinate descent method.'" We also provide a document entitled "Data instruction for 'Portfolio Selection with Nonparametric Value-at-Risk: A Block Coordinate Descent Method'" in the online supplement of this paper to indicate where readers can obtain the actual data values that were used in our paper, and we explain how the data used in the computational experiments of our paper were created.

2. Problem Formulation and Preliminary Properties

Suppose that there are n assets in the market, where the vector of random returns is denoted by $\mathbf{r} = (r_1, \ldots, r_n)^T$. Let $\mathbf{x} = (x_1, \ldots, x_n)^T$ denote a given portfolio with x_i being the weight of asset i. The random return of portfolio $\mathbf{x}$ can be then expressed as $\mathbf{r}^T\mathbf{x}$. The value at risk of portfolio $\mathbf{x}$ at confidence level α $(0.5 < \alpha < 1)$ is defined as the smallest number u such that the probability that loss $-\mathbf{r}^T\mathbf{x}$ exceeds u is not greater than $1-\alpha$, that is,

$$\mathrm{VaR}_\alpha(\mathbf{x}) = \inf\{u \mid \mathbb{P}(u < -\mathbf{r}^T\mathbf{x}) \le 1-\alpha\}. \tag{1}$$

See J. P. Morgan (1996) for more details of VaR.

In the following, we first introduce two nonparametric VaR estimators, kernel VaR and quadratic VaR. We then establish the mean-nonparametric VaR portfolio selection models using kernel VaR and quadratic VaR as risk measures, respectively, and we discuss their properties.

2.1. Nonparametric VaR

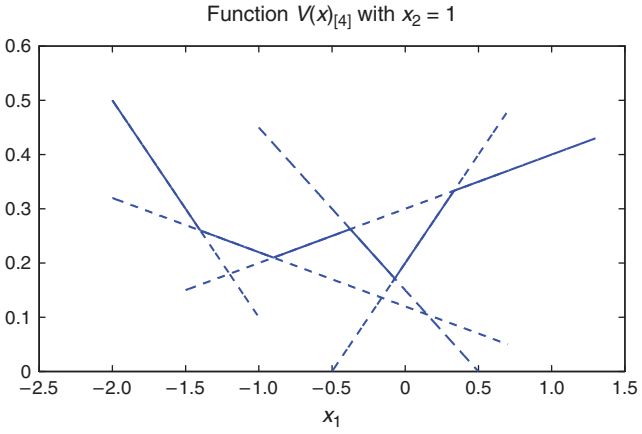
In practice, we usually have to calculate VaR with limited samples of asset returns since we only have some historical asset returns without knowing the exact distribution. Even if we know the full information of the distribution of asset returns, because of some difficulty in computation, we may have to calculate the VaR of portfolio with limited samples generated by some kind of sampling methods, such as Monte Carlo simulation.

Let $(\mathbf{r}^1, \mathbf{r}^2, \ldots, \mathbf{r}^N)$ denote N available historical data or independent and identically distributed (i.i.d) samples of the asset return vector $\mathbf{r}$. The corresponding vector of historical losses or sample losses of portfolio $\mathbf{x}$ is

$$V(\mathbf{x}) = [(-\mathbf{r}^1)^T\mathbf{x}, -(\mathbf{r}^2)^T\mathbf{x}, \ldots, -(\mathbf{r}^N)^T\mathbf{x})].$$

Let $p = \lceil \alpha N \rceil$, where $\lceil t \rceil$ denotes the smallest integer greater than or equal to t. Then the *historical VaR* of portfolio $\mathbf{x}$ is defined as $V(\mathbf{x})_{[p]}$, where we denote by $\mathbf{a}_{[p]}$ the pth smallest element of $\mathbf{a} \in \Re^N$. In the sequel, we also call $V(\mathbf{x})_{[p]}$ a *historical* or *empirical* quantile. It is easy to see that $V(\mathbf{x})_{[p]}$ is a nonconvex function of $\mathbf{x}$ (see, e.g., Gaivoronski and Pflug 2005). Figure 1 describes a two-dimensional function $V(\mathbf{x})_{[4]}$ when variable x_2 is fixed to 1, where $V(x) = [0.1x_1 + 0.3x_2, 0.4x_1 + 0.2x_2, -0.1x_1 + 0.12x_2, -0.3x_1 + 0.15x_2, -0.4x_1 - 0.3x_2]$. From the figure, we can see that $V(\mathbf{x})_{[4]}$ is nonconvex and has many local minimizers.

Figure 1. (Color online) Illustration of Two-Dimensional Function $V(x)_{[4]}$ with $x_2 = 1$, Where $V(x) = [0.1x_1 + 0.3x_2, 0.4x_1 + 0.2x_2, -0.1x_1 + 0.12x_2, -0.3x_1 + 0.15x_2, -0.4x_1 - 0.3x_2]$



The *kernel VaR* estimator of portfolio x can be defined as follows (see Parzen 1979):

$$\mathrm{VaR}^k_\alpha(\mathbf{x}) = \int_0^1 \frac{1}{h} K\left(\frac{s-\alpha}{h}\right) Q(s, \mathbf{x})\, ds, \tag{2}$$

where $h > 0$ is a constant termed as bandwidth, and $Q(s, \mathbf{x})$ is the historical or empirical quantile function defined by $Q(s, \mathbf{x}) = V(\mathbf{x})_{[i]}$ with $i = \max\{1, \lceil sN \rceil\}$, and $K(t)$ is a kernel function satisfying $\int_{-\infty}^{\infty} K(t)\, dt = 1$, $K(t) \ge 0$ and $K(-t) = K(t)$. For instance, the uniform kernel is defined by $K(t) = \frac{1}{2}1_{\{|t| \le 1\}}$ with $1_{\{\cdot\}}$ denoting the indicator function, and the Gaussian kernel is defined by $K(t) = (1/\sqrt{2\pi})e^{-t^2/2}$. The readers can refer to Li and Racine (2007) for more details on kernel estimation of VaR and the related topics about nonparametric econometrics.

The kernel VaR estimator can be viewed as a smoothing version of the empirical quantile function. Indeed, from (2), we can decompose $\mathrm{VaR}^k_\alpha(\mathbf{x})$ as

$$\mathrm{VaR}^k_\alpha(\mathbf{x}) = \sum_{i=1}^{N} \int_{(i-1)/N}^{i/N} \frac{1}{h} K\left(\frac{s-\alpha}{h}\right) Q(s, \mathbf{x})\, ds,$$

$$= \sum_{i=1}^{N} \left(\int_{(i-1)/N}^{i/N} \frac{1}{h} K\left(\frac{s-\alpha}{h} \right) ds \right) V(\mathbf{x})_{[i]}$$
$$= \sum_{i=1}^{N} w_i V(\mathbf{x})_{[i]} \tag{3}$$

with

$$w_i = \int_{(i-1)/N}^{i/N} \frac{1}{h} K\left(\frac{s-\alpha}{h} \right) ds \geq 0, \quad i = 1, \ldots, N. \tag{4}$$

We see from (4) that the bandwidth constant h controls the density of w_i in the neighborhood of the confidence level α. When h varies, the weight w_i will change, especially for i close to $p = \lceil \alpha N \rceil$.

An alternative nonparametric VaR estimator was proposed by Cheng and Peng (2002) by minimizing the average square error of a quadratic approximation to the empirical quantile estimator. The resultant *quadratic VaR* estimator is given by

$$\mathrm{VaR}_\alpha^q(\mathbf{x}) = \sum_{i=0}^{2} \frac{d_i}{v} \int_0^1 (\alpha - s)^i K\left(\frac{s-\alpha}{h} \right) Q(s, \mathbf{x})\, ds \tag{5}$$

with

$$v = \sum_{j=0}^{2} a_j d_j, \quad d_0 = a_2 a_4 - a_3^2, \quad d_1 = a_2 a_3 - a_1 a_4,$$
$$d_2 = a_1 a_3 - a_2^2, \quad a_i = \int_0^1 (\alpha - s)^i K\left(\frac{s-\alpha}{h} \right) ds, \quad i = 0, 1, 2, 3, 4.$$

Similar to the decomposition (3) for kernel VaR, we can decompose the quadratic VaR in (5) as a weighted sum of the empirical quantiles:

$$\mathrm{VaR}_\alpha^q(\mathbf{x}) = \sum_{i=1}^{N} u_i V(\mathbf{x})_{[i]} \tag{6}$$

with

$$u_i = \frac{b_i}{v}, \quad b_i = \sum_{j=0}^{2} a_j^i d_j,$$
$$a_j^i = \int_{(i-1)/N}^{i/N} (\alpha - s)^j K\left(\frac{s-\alpha}{h} \right) ds, \quad j = 0, 1, 2.$$

2.2. Nonparametric VaR-Based Portfolio Selection Models

We now turn to discuss portfolio selection models based on nonparametric VaR. We notice from (3) and (6) that both kernel VaR and quadratic VaR can be expressed as a weighted sum of empirical quantiles over different confidence levels, although the weights can be different for the two nonparametric VaRs. A general form of the mean-nonparametric VaR portfolio selection model can then be expressed as

$$\text{(P)} \quad \min \left\{ \varrho(\mathbf{x}) := \sum_{i=1}^{N} c_i V(\mathbf{x})_{[i]} \,\middle|\, \boldsymbol{\mu}^T \mathbf{x} \geq \rho, \mathbf{x} \in \mathscr{X} \right\},$$

where $c_i \geq 0$ ($i = 1, \ldots, N$) are the weights defined in (3) or (6); $\boldsymbol{\mu}$ is the mean vector of random returns; ρ is a prescribed return level; and $\mathscr{X}$ is a polyhedral set representing the budget constraint, position bounds, and other side constraints of the portfolio. We see that the objective function $\varrho(\mathbf{x})$ in (P) includes historical VaR, kernel VaR, and quadratic VaR as its special cases.

It is evident that problem (P) is only an approximation (with finite samples of asset returns) to the portfolio selection model with exact VaR, which is calculated under full information on the distribution of asset returns. Then, before getting deep into the methodologies for solving (P), the following crucial question should be asked first. What is the quality of the solution generated by the approximate problem (P) when compared with the real optimal one? Will the approximate solutions converge to the real one as the sample size increases to infinity?

Consider problem (P) with kernel VaR as its objective function and denote $\mathbf{x}_N^*$ as the optimal solution to (P) under a group of samples with size N. Denote $\mathbf{x}^*$ as the real optimal solution (although it is unknown in most cases) to the problem with full information on the return distribution. Denote the feasible portfolio set as

$$\Omega = \{ \mathbf{x} \in \Re^n \mid \boldsymbol{\mu}^T \mathbf{x} \geq \rho, \mathbf{x} \in \mathscr{X} \}.$$

Recall that $\mathrm{VaR}_\alpha(\mathbf{x})$ denotes the true VaR value under portfolio $\mathbf{x}$. We will give a convergence property of the approximate optimal VaR, $\mathrm{VaR}_\alpha^k(\mathbf{x}_N^*)$, from (P) to the real optimal VaR, $\mathrm{VaR}_\alpha(\mathbf{x}^*)$, when the sample size goes large. We first list some assumptions and a lemma, which are necessary in the proof of the convergence property.

Suppose that $X_1, X_2, \ldots, X_N$ are i.i.d. random variables with absolutely continuous cumulative distribution function F and a corresponding probability density function f. Furthermore, the corresponding quantile function is defined as $Q(\lambda) = F^{-1}(\lambda) = \inf\{u \mid \mathbb{P}(X \leq u) \geq \lambda\}$. Let $X_{1,N} \leq X_{2,N} \leq \cdots \leq X_{N,N}$ be the order statistics of $X_1, X_2, \ldots, X_N$. Define

$$T_N(\lambda) = \sum_{i=1}^{N} X_{i,N} \int_{(i-1)/N}^{i/N} \frac{1}{h(N)} K\left(\frac{s-\lambda}{h(N)} \right) ds$$

as a kernel estimator of the quantile $Q(\lambda)$ for $0 < \lambda < 1$. We will use the result in Yang (1985) to show in Lemma 1 that the kernel estimator converges to the quantile $Q(\lambda)$ uniformly (distribution free) as $N \to \infty$ and $h(N) \to 0$. For simplicity, we write $h = h(N)$ in the sequel. We first introduce the following seven assumptions:

1. The probability density function $f(\cdot)$ is continuous and strictly positive on $\{x \mid 0 < F(x) < 1\}$.
2. There exists a natural number j and a constant $M > 0$ such that $|Q(\lambda)| \leq M[\lambda(1-\lambda)]^{-j}$, $\forall\, \lambda \in (0, 1)$.
3. $\lim_{x \to \infty} x^\varepsilon [F(-x) + 1 - F(x)] = 0$ for some $\varepsilon > 0$.

4. $Q(\lambda)$ is sufficiently smooth and satisfies that $|Q'(\lambda)| \le M_1$ and $|Q''(\lambda)| \le M_2$, $\forall\, \lambda \in (\delta, 1-\delta)$ for some sufficiently small $\delta > 0$, where M_1 and $M_2 > 0$ are two finite constants.

5. The kernel function $K(\cdot)$ is a probability density function with finite support set $S = [-c, c]$, where c is a finite positive real number (i.e., $K(x) = 0$ for $x < -c$ or $x > c$).

6. $K(\cdot)$ is bounded.

7. $\int_{-\infty}^{\infty} xK(x)\,dx = 0$.

All these assumptions, except for item 4, have been made in the previous related literature (see, e.g., Bickel 1967 and Yang 1985). Assumption 4 states that the quantile function is not too steep in a large portion of the interior of its domain, that is, the interval $(\delta, 1-\delta)$ for some sufficiently small $\delta > 0$. The above assumptions can be satisfied for most distributions under some mild conditions. For example, let us discuss conditions to satisfy Assumption 2. Assume $f(x)$ has a bounded support, that is, there exists a sufficiently large number $M_0 > 0$ such that $f(x) = 0$ for all $x \le -M_0$ or $x \ge M_0$ (this assumption holds naturally because a return of an arbitrary portfolio section in the real world is always finite). Then by setting $M = M_0/4$ and $j = 1$, we have $|Q(\lambda)| \le M_0 \le M[\lambda(1-\lambda)]^{-1}$ for all $0 < \lambda < 1$.

Under Assumptions 1–7, we have the following convergence result.

Lemma 1. *$E[T_N(\lambda)] - Q(\lambda) = o(1/\sqrt{N}) + O(h^2)$ holds true uniformly (independent of the distribution F) for sufficiently large N and small h.*

We have placed the proof of this lemma in the online supplement. Actually, the above lemma was proved in Theorem 2 of Yang (1985) under conditions similar to Assumptions 1–7. We provide in this paper a new proof of Lemma 1 to emphasize that the convergence is independent of the distribution F, that is, a uniform convergence.

Applying Lemma 1, we have the following proposition.

Proposition 1. *Assume that* (i) *for any portfolio $\mathbf{x} \in \Omega$, the cumulative distribution function of portfolio return $\mathbf{r}^T\mathbf{x}$ is absolutely continuous and, furthermore, the distribution satisfies Assumptions* 1–4, *and* (ii) *the Kernel function $K(\cdot)$ satisfies Assuptions* 5–7 *in Lemma* 1. *Denote $\mathbf{x}_N^* := \arg\min_{\mathbf{x}\in\Omega} \mathrm{VaR}_\alpha^k(\mathbf{x})$. Then we have*

$$|E[\mathrm{VaR}_\alpha^k(\mathbf{x}_N^*)] - \mathrm{VaR}_\alpha(\mathbf{x}^*)| = o(1/\sqrt{N}) + O(h^2).$$

We have placed the proof of Proposition 1 in the online supplement. The previous result indicates that the quality of the approximate solution depends on both the sample size and the bandwidth. Furthermore, from Theorem 2 of Yang (1985), it holds that $\mathrm{var}(\mathrm{VaR}_\alpha^k(\mathbf{x}_N^*)) = o(1/(Nh^2))$ under some mild conditions. Thus we can see that the quality of the solution can be guaranteed by enlarging the sample size and properly controlling the bandwidth at the same time (e.g., by setting $h = O(1/\sqrt[4]{N})$).

According to Cheng and Peng (2002), when the bandwidth h is small enough, the quadratic VaR, $\mathrm{VaR}_\alpha^q(\mathbf{x})$, can be reduced to a kernel VaR under a newly defined kernel function $\bar{K}(t) = (h(a_4 - a_2 t^2 h^2)/(a_0 a_4 - a_2^2))K(t)$. Under this circumstance, the quadratic VaR should share a similar convergence property with respect to the sample size and the bandwidth as the kernel VaR.

Now the remaining question is how to solve model (P). A straightforward way is to reformulate(P) as a tractable model that can be solved by some standard software packages. Indeed, by introducing a continuous variable γ_i for each empirical quantile $V(\mathbf{x})_{[i]}$ and a 0-1 variable z_t^i for each sample or scenario, we can rewrite the constraint $V(\mathbf{x})_{[i]} \le \gamma_i$ as

$$-(\mathbf{r}^t)^T\mathbf{x} \le \gamma_i + M_t z_t^i, \quad z_t^i \in \{0,1\}, \quad t = 1,\dots,N,$$
$$\sum_{t=1}^{N} z_t^i \le N - i,$$

where M_t is a sufficiently large number, for instance, $M_t \ge \max_{\mathbf{x}\in\mathscr{X}}[(\mathbf{r}^t)^T\mathbf{x}] - \min_{\mathbf{x}\in\mathscr{X}}[(\mathbf{r}^t)^T\mathbf{x}]$ for each t. Thus, model (P) is equivalent to the following mixed-integer 0-1 linear programming problem:

$$\begin{aligned}
(\mathrm{MIP}) \quad & \min \ \sum_{i=1}^{N} c_i\gamma_i, \\
\text{s.t.} \ & \boldsymbol{\mu}^T\mathbf{x} \ge \rho, \quad \mathbf{x} \in \mathscr{X}, \\
& -(\mathbf{r}^t)^T\mathbf{x} \le \gamma_i + M_t z_t^i, \quad t = 1,\dots,N,\ i = 1,\dots,N, \\
& \sum_{t=1}^{N} z_t^i \le N - i, \quad i = 1,\dots,N, \\
& z_t^i \in \{0,1\}, \quad t = 1,\dots,N,\ i = 1,\dots,N.
\end{aligned}$$

We see that problem (MIP) has $n + N$ continuous variables, N^2 binary variables, and $N^2 + N$ additional constraints. It is clear that problem (MIP) is a large-scale mixed-integer 0-1 linear programming problem even for problems with a relatively small sample size. For instance, $N = 100$ gives rise to 10000 binary variables and 10100 additional linear constraints in model (MIP). This dimensionality challenge makes model (MIP) difficult to solve for problems with realistic sample size even by the most advanced mixed-integer programming solvers such as CPLEX. Thus, some other solution methodologies should be further developed.

Now, let us turn to the discussion of some preliminary properties of the model (P) that are useful in developing the solution methodologies. Notice that the objective function $\varrho(\mathbf{x})$ in (P) is both nonconvex and nonsmooth since each empirical quantile $V(\mathbf{x})_{[i]}$ is a nonconvex and nonsmooth function of $\mathbf{x}$.

Lemma 2. *The function $\varrho(\mathbf{x})$ is a locally Lipschitz function of $\mathbf{x}$ on $\Re^n$.*

Proof. Denote $R = (\mathbf{r}^1, \mathbf{r}^2, \ldots, \mathbf{r}^N)^T$ and $\mathbf{y} = -R\mathbf{x}$. Let

$$\phi(\mathbf{y}) = \sum_{i=1}^{N} c_i \mathbf{y}_{[i]}.$$

It suffices to show that $\phi(\mathbf{y})$ is a locally Lipschitz function of $\mathbf{y}$. We first note that $\mathbf{y}_{[i]}$ can be expressed as a DC function (difference of two convex functions):

$$\mathbf{y}_{[i]} = \sum_{j=i}^{N} \mathbf{y}_{[j]} - \sum_{j=i+1}^{N} \mathbf{y}_{[j]}. \tag{7}$$

Let $\psi_i(\mathbf{y}) = \sum_{j=i}^{N} \mathbf{y}_{[j]}$ and $\psi_{i+1}(\mathbf{y}) = \sum_{j=i+1}^{N} \mathbf{y}_{[j]}$. Then, $\psi_i(\mathbf{y})$ (or $\psi_{i+1}(\mathbf{y})$) is the sum of $N-i+1$ (or $N-i$) largest components of $\mathbf{y}$. Since the sum of k largest components is a convex function (see, e.g., Example 3.6 in Boyd and Vandenberghe 2004), $\psi_i(\mathbf{y})$ and $\psi_{i+1}(\mathbf{y})$ are convex functions on $\Re^N$. Thus, $\mathbf{y}_{[i]}$ is a DC function and hence is locally Lipschitz function since convex function and concave functions on $\Re^N$ are locally Lipschitz and the sum of two locally Lipschitz functions is still locally Lipschitz. Therefore, $\phi(\mathbf{y})$, as a linear combination of locally Lipschitz functions, is also a locally Lipschitz function. Finally, since $\varrho(\mathbf{x})$ is exactly $\phi(\mathbf{y})$ with $\mathbf{y} = -R\mathbf{x}$, $\varrho(\mathbf{x})$ is also a locally Lipschitz function. □

By Lemma 2, the Clarke generalized gradient of $\phi(\mathbf{y})$, denoted by $\partial\phi(\mathbf{y})$, exists for any $\mathbf{y} \in \Re^N$ (see Clarke 1983). Moreover, by the chain rule, we have $\partial\varrho(\mathbf{x}) = -R^T\partial\phi(\mathbf{y})$, where $\mathbf{y} = -R\mathbf{x}$. Notice that problem (P) can be rewritten as

$$(\mathrm{P}_m) \quad \min\{\phi(\mathbf{y}) \mid \mathbf{y} = -R\mathbf{x},\ \mathbf{x} \in \Omega\}.$$

We have the following first-order stationary condition for (P_m) (see Clarke 1983).

Proposition 2. *Let $(\mathbf{x}^*, \mathbf{y}^*)$ be a local optimal solution of (P_m). Then the following first-order stationary condition holds:*

$$0 \in -R^T\partial\phi(\mathbf{y}^*) + N_\Omega(\mathbf{x}^*), \tag{8}$$

where $\mathbf{y}^* = -R\mathbf{x}^*$, and $N_\Omega(\mathbf{x}^*)$ is the normal cone of Ω at $\mathbf{x}^*$ defined by $N_\Omega(\mathbf{x}^*) = \{\mathbf{y} \in \Re^n \mid \forall\, \mathbf{x} \in \Omega, \langle \mathbf{y}, \mathbf{x} - \mathbf{x}^* \rangle \le 0\}$.

An alternative mean-nonparametric VaR portfolio selection model is to maximize the expected return under a constraint on the nonparametric VaR. The resultant problem can be formulated as

$$(\mathrm{P}_c) \quad \max\left\{\boldsymbol{\mu}^T\mathbf{x} \,\middle|\, \varrho(\mathbf{x}) := \sum_{i=1}^{N} c_i V(\mathbf{x})_{[i]} \le \varsigma_0,\ \mathbf{x} \in \mathscr{X}\right\},$$

where ς_0 is a given risk level. Similar to formulation (MIP), we can reformulate (P_c) as an equivalent mixed-integer 0-1 linear programming problem with $n + N$ continuous variables, N^2 binary variables, and $N^2 + N$ additional constraints.

The mixed-integer programming reformulations for (P_m) and (P_c) suggest that these mean-nonparametric VaR models are nonconvex optimization problems, which are in general NP-hard. For problems with a realistic sample size and a medium-to-large number of assets, it is reasonable to consider *local methods* that can efficiently generate some local solutions of these nonconvex models. This will be the task in the next section.

3. Block Coordinate Descent Methods

In this section, we propose two block coordinate descent methods for problems (P_m) and (P_c), respectively. Block coordinate descent methods, also known as alternative direction methods, have been successfully applied to convex programming and some nonconvex optimization problems arising from image processing and matrix optimization (see, e.g., Goldstein 2009, Yin et al. 2008, He et al. 2012, Xu et al. 2012, Shen et al. 2014). The idea of the block coordinate descent method is to alternatively fix some variables in the augmented Lagrangian formulation of (P_m) or (P_c) and solve the resulting more tractable subproblems at each iteration of the algorithm. It can also be viewed as a solution method that iterates along alternating directions and eventually achieves global or local optimality.

We first consider applying a block coordinate descent method to (P_m). By introducing coupling constraints $y_i = -(\mathbf{r}^i)^T\mathbf{x}$ $(i = 1, \ldots, N)$, the problem (P_m) can then be rewritten as

$$\min\left\{\sum_{i=1}^{N} c_i \mathbf{y}_{[i]} \,\middle|\, \mathbf{y} = -R\mathbf{x},\ \boldsymbol{\mu}^T\mathbf{x} \ge \rho,\ \mathbf{x} \in \mathscr{X}\right\}, \tag{9}$$

where we recall that $R = (\mathbf{r}^1, \mathbf{r}^2, \ldots, \mathbf{r}^N)^T$ and $\mathbf{y}_{[i]}$ is the ith smallest element of $\mathbf{y}$. The augmented Lagrangian function for (9) is defined as

$$\mathscr{L}_\sigma(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) = \sum_{i=1}^{N} c_i \mathbf{y}_{[i]} + \boldsymbol{\lambda}^T(\mathbf{y} + R\mathbf{x}) + \frac{\sigma}{2}\|\mathbf{y} + R\mathbf{x}\|^2, \tag{10}$$

where $\sigma > 0$ is the penalty parameter and $\boldsymbol{\lambda} \in \Re^m$ is the Lagrangian multiplier vector for $\mathbf{y} = -R\mathbf{x}$. For given σ and $\boldsymbol{\lambda}$, the augmented Lagrangian relaxation of (9) is

$$\min\{\mathscr{L}_\sigma(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) \mid \boldsymbol{\mu}^T\mathbf{x} \ge \rho,\ \mathbf{x} \in \mathscr{X}\}. \tag{11}$$

A key observation is that if either $\mathbf{x}$ or $\mathbf{y}$ is fixed in (11), then (11) can be reduced to a tractable subproblem with decision variable $\mathbf{y}$ or $\mathbf{x}$. We can therefore apply the algorithmic framework of a block coordinate descent method to (9).

Let $\sigma_k > 0$ be given. Suppose at the kth iteration, we have a tuple $(\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}^k)$, where $\boldsymbol{\mu}^T\mathbf{x}^k \ge \rho$ and $\mathbf{x}^k \in \mathscr{X}$.

Consider that we solve the following two subproblems alternatively:

$$\mathbf{y}^{k+1} = \arg\min\{\mathcal{L}_{\sigma_k}(\mathbf{x}^k, \mathbf{y}, \boldsymbol{\lambda}^k) \mid \mathbf{y} \in \Re^N\}, \quad (12)$$

$$\mathbf{x}^{k+1} = \arg\min\{\mathcal{L}_{\sigma_k}(\mathbf{x}, \mathbf{y}^{k+1}, \boldsymbol{\lambda}^k) \mid \mathbf{x} \in \Omega\}. \quad (13)$$

Since $\mathcal{L}_{\sigma_k}(\mathbf{x}, \mathbf{y}^{k+1}, \boldsymbol{\lambda}^k)$ is a convex quadratic function of $\mathbf{x}$, the subproblem (13) is a convex quadratic program when the set $\mathcal{X}$ is polyhedral, which can be solved efficiently.

In the following, we show that the subproblem (12) can also be solved efficiently. We first note that $\mathcal{L}_{\sigma_k}(\mathbf{x}^k, \mathbf{y}, \boldsymbol{\lambda}^k)$ can be written as

$$\mathcal{L}_{\sigma_k}(\mathbf{x}^k, \mathbf{y}, \boldsymbol{\lambda}^k) = \sum_{i=1}^{N} c_i \mathbf{y}_{[i]} + \frac{\sigma_k}{2}\|\mathbf{y} - \boldsymbol{\omega}^k\|^2 - \frac{1}{2\sigma_k}\|\boldsymbol{\lambda}^k\|^2 \quad (14)$$

with $\boldsymbol{\omega}^k = -R\mathbf{x}^k - \boldsymbol{\lambda}^k/\sigma_k$. We have the following lemma for subproblem (12).

Lemma 3. *Let $\boldsymbol{\omega}^k \in \Re^N$ be ranked in an ascending order*:

$$\omega^k_{i_1} \le \omega^k_{i_2} \le \cdots \le \omega^k_{i_N}, \quad (15)$$

where $\{i_1, i_2, \ldots, i_N\}$ is a permutation of $\{1, 2, \ldots, N\}$. Consider the following convex quadratic program:

$$\begin{aligned} \min \quad & \sum_{j=1}^{N} c_j y_{i_j} + \frac{\sigma_k}{2}\|\mathbf{y} - \boldsymbol{\omega}^k\|^2 - \frac{1}{2\sigma_k}\|\boldsymbol{\lambda}^k\|^2 \quad (16)\\ \text{s.t.} \quad & y_{i_j} \le y_{i_{j+1}}, \quad j = 1, \ldots, N-1. \end{aligned}$$

Then, any optimal solution to problem (16) is also an optimal solution to problem (12).

Proof. We suppose, without loss of generality, that $\omega^k_1 \le \omega^k_2 \le \cdots \le \omega^k_N$. It is evident that there exists an optimal solution to problem (16). Assume that $\mathbf{y}^*$ is an optimal solution to problem (16). Now we show that $\mathbf{y}^*$ is also an optimal solution to problem (12). We prove this by contradiction. Suppose that $\mathbf{y}^*$ is not an optimal solution to problem (12). Then there must exist an optimal solution $\tilde{\mathbf{y}}$ to problem (12) such that

$$\mathcal{L}_{\sigma_k}(\mathbf{x}^k, \mathbf{y}^*, \boldsymbol{\lambda}^k) - \mathcal{L}_{\sigma_k}(\mathbf{x}^k, \tilde{\mathbf{y}}, \boldsymbol{\lambda}^k) > 0. \quad (17)$$

For the optimal solution $\tilde{\mathbf{y}}$ to problem (12), we assume that there is an s such that $1 \le s < t \le N$ and $\tilde{y}_s > \tilde{y}_t$. Swapping the positions of $\tilde{y}_s$ and $\tilde{y}_t$, we obtain a new point $\bar{\mathbf{y}} = (\tilde{y}_1, \ldots, \tilde{y}_t, \ldots, \tilde{y}_s, \ldots, \tilde{y}_N)^T$. Since $\tilde{\mathbf{y}}$ and $\bar{\mathbf{y}}$ have the same set of components, we have $\tilde{\mathbf{y}}_{[i]} = \bar{\mathbf{y}}_{[i]}$ for $i = 1, \ldots, N$. Thus, by (14) and the fact that $\omega^k_t \ge \omega^k_s$, we have

$$\begin{aligned} \mathcal{L}_{\sigma_k}(\mathbf{x}^k, \tilde{\mathbf{y}}, \boldsymbol{\lambda}^k) - \mathcal{L}_{\sigma_k}(\mathbf{x}^k, \bar{\mathbf{y}}, \boldsymbol{\lambda}^k) &= \frac{\sigma_k}{2}(\|\tilde{\mathbf{y}} - \boldsymbol{\omega}^k\|^2 - \|\bar{\mathbf{y}} - \boldsymbol{\omega}^k\|^2)\\ &= \sigma_k(\tilde{y}_s - \tilde{y}_t)(\omega^k_t - \omega^k_s)\\ &\ge 0. \end{aligned}$$

Repeating the swapping procedure if necessary, we can eventually get a vector $\hat{y}$ such that

$$\hat{y}_j \le \hat{y}_{j+1}, \quad j = 1, \ldots, N-1; \quad (18)$$

and

$$\begin{aligned} 0 \le\ & \mathcal{L}_{\sigma_k}(\mathbf{x}^k, \tilde{\mathbf{y}}, \boldsymbol{\lambda}^k) - \mathcal{L}_{\sigma_k}(\mathbf{x}^k, \hat{\mathbf{y}}, \boldsymbol{\lambda}^k)\\ <\ & \mathcal{L}_{\sigma_k}(\mathbf{x}^k, \mathbf{y}^*, \boldsymbol{\lambda}^k) - \mathcal{L}_{\sigma_k}(\mathbf{x}^k, \hat{\mathbf{y}}, \boldsymbol{\lambda}^k), \end{aligned}$$

where the second strict inequality is from (17). By these facts, $\mathbf{y}^*$ cannot be an optimal solution to problem (16), which contradicts the original assumption. Thus $\mathbf{y}^*$ must be an optimal solution to problem (12). □

In the following, we develop a BCD algorithm for solving (P_m). First, we have the following lemma when considering problem (12).

Lemma 4. *For any pregiven $\boldsymbol{\lambda}^k$ and $\mathbf{x}^k$ and $\epsilon > 0$, there exists a bounded number $\bar{\sigma}$ such that, for any $\sigma_k > \bar{\sigma}$, the optimal solution $\mathbf{y}^{k+1}$ to subproblem (12) satisfies $\|\mathbf{y}^{k+1} + R\mathbf{x}^k\| \le \epsilon$.*

Proof. Notice that $y = -R\mathbf{x}^k$ is a feasible solution to problem (12) and the corresponding objective value of problem (12) is $\sum_{i=1}^{N} c_i(-R\mathbf{x}^k)_{[i]}$, which gives an upper bound to its optimal objective value. Assume that the claim of the lemma is not true. Then for any given $\bar{\sigma}$, there exists at least one number $\sigma_k > \bar{\sigma}$ such that the optimal solution $\mathbf{y}^{k+1}$ to problem (12) satisfying $\|\mathbf{y}^{k+1} + R\mathbf{x}^k\| > \epsilon$. Thus, the corresponding optimal objective value is greater than $\sum_{i=1}^{N} c_i \mathbf{y}^{k+1}_{[i]} + (\boldsymbol{\lambda}^k)^T \cdot (\mathbf{y}^{k+1} + R\mathbf{x}^k) + (\sigma_k\epsilon^2)/2$. Notice that we have

$$\sum_{i=1}^{N} c_i \mathbf{y}^{k+1}_{[i]} + (\boldsymbol{\lambda}^k)^T(\mathbf{y}^{k+1} + R\mathbf{x}^k) + (\sigma_k\epsilon^2)/2 \le \sum_{i=1}^{N} c_i(-R\mathbf{x}^k)_{[i]},$$

which indicates that if $\sigma_k \to +\infty$, then $\sum_{i=1}^{N} c_i \mathbf{y}^{k+1}_{[i]} + (\boldsymbol{\lambda}^k)^T(\mathbf{y}^{k+1} + R\mathbf{x}^k) \to -\infty$, which further implies that $\|\mathbf{y}^{k+1}\| \to +\infty$ as $\sigma_k \to +\infty$. However, from a result for penalty methods (see Proposition 4.2.1 of Bertsekas 1999), we have that $\|\mathbf{y}^{k+1}\| \to \|-R\mathbf{x}^k\| < +\infty$ as $\sigma_k \to +\infty$. This contradiction indicates that the claim of Lemma 4 is true. □

The BCD method for solving (P_m) is actually a procedure to solve subproblems (12) and (13) iteratively. In the following, we first show the convergent property of the BCD method without giving a rule for updating $\boldsymbol{\lambda}^k$, which will be discussed later. For the moment, we only suppose that $\{\boldsymbol{\lambda}^k\}$ is a constant.

Algorithm 1 (BCD algorithm 1 for (P_m))

Step 0. Choose two accuracy tolerance parameters $\epsilon_1 > 0$ and $\epsilon_2 > 0$. Choose penalty parameters $\sigma_0 \ge 0$ and $c > 1$. Choose an initial $\mathbf{x}^0 \in \Omega$. Set $k := 0$.

Step 1. Set $\boldsymbol{\lambda}^k = \boldsymbol{\lambda}$ (a constant). Solve problem (16) to obtain an optimal solution $\mathbf{y}^{k+1}$ and then solve (13) to obtain an optimal solution $\mathbf{x}^{k+1}$.

Step 2. If $\|\mathbf{y}^{k+1}+R\mathbf{x}^{k+1}\|\le\epsilon_1$ and $\|\mathbf{x}^{k+1}-\mathbf{x}^k\|\le\epsilon_2$, stop.

Step 3. Set $\sigma_{k+1} = \begin{cases}\sigma_k, & \text{if } \|\mathbf{y}^{k+1}+R\mathbf{x}^k\|\le\epsilon_1\\ c\sigma_k, & \text{else}\end{cases}$

and $k := k+1$. Go back to Step 1.

For any $(\mathbf{x},\mathbf{y})$ that satisfies $\|\mathbf{y}+R\mathbf{x}\|\le\epsilon$ and $\mathbf{x}\in\Omega$, we call it an *ϵ-feasible solution* to (P_m). If $(\mathbf{x},\mathbf{y})$ satisfies $\|\mathbf{y}+R\mathbf{x}\|\le\epsilon$, $\mathbf{x}\in\Omega$ and $0\in R^T\partial\phi(\mathbf{y})+N_\Omega(\mathbf{x})+\boldsymbol{\eta}$, we call it an *$\epsilon$-feasible solution* to (P_m) satisfying the *$\boldsymbol{\eta}$-near first-order stationary condition*.

The following theorem establishes the convergence property of Algorithm 1 to a first-order stationary point of (P_m).

Theorem 1. *Assume that $R^TR>0$ and the feasible set Ω is bounded. Let $\{(\mathbf{x}^k,\mathbf{y}^k)\}$ be the sequence generated by Algorithm 1. If Algorithm 1 terminates within finite steps, then it generates an ϵ_1-feasible solution to (P_m) satisfying the $\boldsymbol{\eta}$-near first-order stationary condition, where $\|\boldsymbol{\eta}\|$ is in the order of $O(\sigma_k\epsilon_2)$. Otherwise, there must exist a convergence subsequence of $\{(\mathbf{x}^k,\mathbf{y}^k)\}$, and the accumulation point of such a subsequence is an ϵ_1-feasible solution to (P_m) satisfying the first-order stationary condition (8).*

Proof. See the online supplement. □

In Algorithm 1, we just select $\boldsymbol{\lambda}^k$ as a constant for all k. It is a combination of a penalty method and the Lagrangian multiplier method without updating the multiplier. From the proof of Theorem 1 (see the online supplement, formulation (3)), we have

$$\bar{\boldsymbol{\lambda}} = \lim_{k_i\to+\infty}[\boldsymbol{\lambda}^{k_i}+\sigma_{k_i}(\mathbf{y}^{k_i+1}+R\mathbf{x}^{k_{i+1}})],$$

where $\bar{\boldsymbol{\lambda}}$ is an "optimal" Lagrangian multiplier vector. This suggests the following rule for updating the Lagrangian multiplier vector for each iteration:

$$\boldsymbol{\lambda}^{k+1}=\boldsymbol{\lambda}^k+\sigma_k(\mathbf{y}^{k+1}+R\mathbf{x}^{k+1}).$$

Actually, this kind of multiplier update is the essential point of the multiplier method, which has already been proved to be more efficient and robust than the pure penalty method or the pure Lagrangian multiplier method. Based on this observation, we propose the following BCD method with updating the Lagrangian multiplier vector.

Algorithm 2 (BCD Algorithm 2 for (P_m))

Step 0. Choose two accuracy tolerance parameters $\epsilon_1>0$ and $\epsilon_2>0$. Choose penalty parameters $\sigma_0\ge 0$, $c>1$ and initial Lagrangian multiplier vector $\boldsymbol{\lambda}^0\in\Re^N$. Choose an initial $\mathbf{x}^0\in\Omega$. Set $k:=0$.

Step 1. Set $\boldsymbol{\lambda}^k=\boldsymbol{\lambda}^0$. Solve problem (16) to obtain an optimal solution $\mathbf{y}^{k+1}$ and then solve (13) to obtain an optimal solution $\mathbf{x}^{k+1}$.

Step 2. If $\|\mathbf{y}^{k+1}+R\mathbf{x}^{k+1}\|\le\epsilon_1$ and $\|\mathbf{x}^{k+1}-\mathbf{x}^k\|\le\epsilon_2$, stop.

Step 3. Set

$$\boldsymbol{\lambda}^{k+1}=\boldsymbol{\lambda}^k+\sigma_k(\mathbf{y}^{k+1}+R\mathbf{x}^{k+1})$$
$$\sigma_{k+1}=\begin{cases}\sigma_k, & \text{if } \|\mathbf{y}^{k+1}+R\mathbf{x}^k\|\le\epsilon_1;\\ c\sigma_k, & \text{else}\end{cases}$$

and $k:=k+1$. Go back to Step 1.

Obviously, according to Theorem 1, the convergence property for Algorithm 2 can be established by just fixing $\boldsymbol{\lambda}^k$ after sufficiently many iterations, which is also a practical choice for a numerical implementation.

Similar to Algorithm 2, we can apply the idea of the BCD method to solve problem (P_c). We first notice that problem (P_c) can be rewritten as

$$\min\left\{-\boldsymbol{\mu}^T\mathbf{x}\,\middle|\,\mathbf{y}+R\mathbf{x}=0,\ \sum_{i=1}^N c_i\mathbf{y}_{[i]}\le\varsigma_0,\ \mathbf{x}\in\mathscr{X}\right\}.$$

Dualizing the constraint $\mathbf{y}+R\mathbf{x}=0$, the augmented Lagrangian function of (P_c) becomes

$$\mathscr{L}_\sigma^c(\mathbf{x},\mathbf{y},\boldsymbol{\lambda})=-\boldsymbol{\mu}^T\mathbf{x}+\boldsymbol{\lambda}^T(\mathbf{y}+R\mathbf{x})+\frac{\sigma}{2}\|\mathbf{y}+R\mathbf{x}\|^2, \tag{19}$$

where $\sigma>0$ is the penalty parameter and $\boldsymbol{\lambda}$ is the Lagrangian multiplier vector. Suppose that, at the kth iteration, we have a tuple $(\mathbf{x}^k,\mathbf{y}^k,\boldsymbol{\lambda}^k)$ such that $\mathbf{x}^k\in\mathscr{X}$ and $\sum_{i=1}^N c_i\mathbf{y}^k_{[i]}\le\varsigma_0$. The two subproblems at the kth iteration of the BCD method are as follows:

$$\mathbf{y}^{k+1}=\arg\min\left\{\mathscr{L}_{\sigma_k}^c(\mathbf{x}^k,\mathbf{y},\boldsymbol{\lambda}^k)\,\middle|\,\sum_{i=1}^N c_i\mathbf{y}_{[i]}\le\varsigma_0\right\}, \tag{20}$$

$$\mathbf{x}^{k+1}=\arg\min\{\mathscr{L}_{\sigma_k}^c(\mathbf{x},\mathbf{y}^{k+1},\boldsymbol{\lambda}^k)\mid\mathbf{x}\in\mathscr{X}\}. \tag{21}$$

We see from (19) that subproblem (21) is a convex quadratic program. Also, similar to Lemma 3, we can show that (20) can be reduced to the following convex quadratic program:

$$\begin{aligned}\min\ & \frac{\sigma_k}{2}\|\mathbf{y}-\boldsymbol{\omega}^k\|^2-\boldsymbol{\mu}^T\mathbf{x}^k-\frac{1}{2\sigma_k}\|\boldsymbol{\lambda}^k\|^2\\ \text{s.t.}\ & y_{i_j}\le y_{i_{j+1}},\quad j=1,\dots,N-1,\\ & \sum_{j=1}^N c_j y_{i_j}\le\varsigma_0,\end{aligned}$$

where $\{\omega^k_{i_j}\}$ is ranked in an ascending order the same as in (15).

The iteration process of the BCD method for (P_c) is the same as Algorithm 2 except that in Step 1, we solve subproblems (20) and (21), instead of (12) and (13). A convergence property that is similar to the one in Theorem 1 can also be established for the BCD method for problem (P_c).

4. Computational Results

In this section, we present computational results of the proposed BCD methods for solving the two portfolio selection models (P_m) and (P_c), respectively. The purpose of our computational experiment is to evaluate the effectiveness of the methods when applied to the test problems with realistic sizes using data from a real market.

4.1. Test Problems

To build the test bed, we use the Thomson Reuter database to collect 2,039 daily returns of the constituents of Standard and Poor's 500 index ranging from November 2004 to November 2012. After excluding the constituents with missing data, we have 460 stocks as our portfolio candidates. For each pair of (n, N), we randomly generate five instances of (P_m) and (P_c), respectively, where the n stocks are randomly chosen from the 460 stocks and the N samples $(\mathbf{r}^1, \ldots, \mathbf{r}^N)$ are randomly chosen from the 2,039 daily returns of the corresponding stocks.

Note that the decomposition formulations of the kernel VaR and the quadratic VaR are the same except for the values of the smoothing weights: $\mathrm{VaR}^k_\alpha(\mathbf{x}) = \sum_{i=1}^N w_i V(\mathbf{x})_{[i]}$ with w_i defined in (4) and $\mathrm{VaR}^q_\alpha(\mathbf{x}) = \sum_{i=1}^N u_i V(\mathbf{x})_{[i]}$ with u_i defined in (7). In our test, we only consider test problems for (P_m) and (P_c) using kernel VaR. We set $c_i = w_i$ in (P_m) and (P_c), where w_i, $i = 1, \ldots, N$, are computed by (4) with the Gaussian kernel function $K(t) = (1/\sqrt{2\pi})e^{-t^2/2}$. Also, the bandwidth constant is set as $h = 1.06N^{-0.2}\bar{\sigma}$ with $\bar{\sigma} = \sqrt{\bar{\mathbf{x}}^T\Sigma\bar{\mathbf{x}}}$, where Σ is the sample covariance matrix and $\bar{\mathbf{x}}$ is the portfolio with equal weight, that is, $\bar{\mathbf{x}} = (1/n, \ldots, 1/n)^T$. In problem ($P_m$), we set the prescribed return level at $\rho = 0.1\%$. In problem (P_c), we set the risk level as $\varsigma_0 = 0.02$. The set $\mathscr{X}$ in (P_m) and (P_c) is set as $\mathscr{X} = \{\mathbf{x} \in \Re^n \mid \sum_{i=1}^n x_i = 1, 0 \le x_i \le 0.5, i = 1, \ldots, n\}$. For all the test problems, we set the confidence level $\alpha = 95\%$.

4.2. Implementation Details

In our implementation of Algorithm 2 for (P_m) and its variant for (P_c), the initial point x^0 is generated by solving the CVaR approximation problems of (P_m) and (P_c), where $\varrho(\mathbf{x})$ is replaced by the sample CVaR of portfolio $\mathbf{x}$. It is well known that these CVaR approximation problems can be formulated as linear programming problems (see Rockafellar and Uryasev 2000). In Algorithm 2, we set the accuracy tolerance $\epsilon_1 = 2 \times 10^{-5}$ and $\epsilon_2 = 10^{-4}$. Other parameters in Step 0 are set as $\sigma_0 = 0.01$, $c = 3$, and $\boldsymbol{\lambda}^0 = 0$.

The coefficients M_t in formulation (MIP) for problem (P_m) are generated by a strengthening procedure similar to the coefficient strengthening scheme in Qiu et al. (2014). Notice that the lower bound and upper bound of γ_t for all t can be given, respectively, by

$$\underline{\gamma} := \min_{t=1,\ldots,N} \min_{x\in\Omega} -(\mathbf{r}^t)^T\mathbf{x} \quad \text{and} \quad \bar{\gamma} := \max_{t=1,\ldots,N} \max_{x\in\Omega} -(\mathbf{r}^t)^T\mathbf{x}.$$

We first set the value $M_t^0 = \max_{\mathbf{x}\in\mathscr{X}}[-(\mathbf{r}^t)^T\mathbf{x}] - \underline{\gamma}$ as the initial candidate of M_t for each t and then generate a strengthened coefficient based on M_t^0. Denote

$$F(M^0) = \left\{ \mathbf{x} \in \Omega \;\middle|\; \exists (\mathbf{z}, \boldsymbol{\gamma}) \in [0,1]^{N^2} \times [\underline{\gamma}, \bar{\gamma}]^N, \right.$$
$$\text{s.t. } -(\mathbf{r}^t)^T\mathbf{x} \le \gamma_i + M_t^0 z_t^i, \quad t, i = 1, \ldots, N,$$
$$\left. \sum_{t=1}^N z_t^i \le N - i, \; i = 1, \ldots, N \right\}.$$

Then a strengthened upper bound of $-(\mathbf{r}^t)^T\mathbf{x} - \gamma_i$ can be obtained by

$$M_t' := \max\{-(\mathbf{r}^t)^T\mathbf{x} \colon \mathbf{x} \in F(M^0)\} - \underline{\gamma}.$$

In our experiment, problem (MIP) is constructed with M_t' as the coefficient of binary variable z_t^i. The corresponding coefficients in the MIP formulation for problem (P_c) are generated in a similar procedure.

Algorithm 2 and its variant for problem (P_c) have been implemented in Matlab and been run on a PC (3.2 GHz and 16 G RAM). All the linear program, quadratic program, and mixed-integer programming problems in our computational experiments are solved by the `LP`, `QP`, and `MIP` solvers in `CPLEX 12.5` with MATLAB interface, respectively. The parameters of the CPLEX procedure for MIP formulations of problems (P_m) and (P_c) are set by default, where the absolute and relative gaps of the lower bound and upper bound are 10^{-6} and 10^{-4}, respectively.

4.3. Numerical Results

To evaluate the effectiveness of the BCD method, we first compare the method with the `MIP` solver in `CPLEX 12.5` when it applies to the mixed-integer 0-1 linear programming reformulation (MIP) for test problems with number of sample $N \le 500$. For test problems with number of sample $N = 1{,}000, 1{,}500, 2{,}000$, CPLEX fails to find any feasible solution for reformulation (MIP) and terminates because of the memory limitation.

To compare the performance of the BCD method with formulation (MIP), we set the maximum CPU time of CPLEX as 3,600 seconds and record the objective value of the best feasible solution when terminated. If the test problem is not solved within the maximum CPU time, we also record the relative final gap of CPLEX, which is defined by

$$\text{gap} = \frac{(\text{upper bound} - \text{lower bound})}{\text{upper bound}},$$

where "upper bound" is the objective value of the current best feasible solution. For problem (P_m), we calculate the following improvement ratio of BCD method over CPLEX:

$$\text{imp.ratio} = \frac{\varrho(\mathbf{x}_C^*) - \varrho(\mathbf{x}_B^*)}{\varrho(\mathbf{x}_C^*)},$$

where $\mathbf{x}_C^*$ denotes the best feasible solution found by CPLEX for formulation (MIP) and $\mathbf{x}_B^*$ is the local solution found by Algorithm 2. For problem (P_c), the improvement ratio of the BCD method over CPLEX is defined by

$$\text{imp.ratio} = \frac{\boldsymbol{\mu}^T\mathbf{x}_B^* - \boldsymbol{\mu}^T\mathbf{x}_C^*}{\boldsymbol{\mu}^T\mathbf{x}_C^*}.$$

Tables 1 and 2 summarize the comparison between the BCD methods and MIP formulations for (P_m) and (P_c), respectively, where "fval" denotes the objective value of the feasible solution found, "time" is the CPU time (seconds) used by the BCD methods or CPLEX for MIP formulations, "iter" stands for the number of iterations of the BCD methods, "node" is the number of nodes explored by the MIP solver of CPLEX when applied to the MIP reformulations of (P_m) or (P_c). In our experiment, we choose the sample size N not smaller than portfolio size n to guarantee that $R^TR > 0$, which is also a reasonable requirement in portfolio management. All the results in the tables are averaged for five test problems.

From Tables 1 and 2, we see that when sample size $N \le 300$, CPLEX cannot solve most of the test problems in the maximum CPU time (3,600 seconds) and terminates with a large final gap, meanwhile BCD methods can find feasible solutions for all test problems of (P_m) and (P_c) in 70 seconds, which are similar to the incumbent feasible solutions of CPLEX obtained when reaching the maximum CPU time. For the test problem with $N = 400, 500$, BCD methods generally generate solutions in 80 seconds, which are better than the incumbent feasible solutions obtained by CPLEX in 3,600 seconds, especially when the portfolio size is large. For the test problems with $(n, N) = (200, 500), (300, 500), (400, 500)$, the average improvement ratios of BCD methods for problem (P_m) are 16.5%, 21.8%, and 31.7%, respectively, while for problem (P_c), the average improvement ratios are 15.4%, 22.8%, and 33.6%, respectively. The efficiency of the BCD methods also can be seen from the number of iterations in Tables 1 and 2. Since Algorithm 2 and its variant for (P_c) solve two convex quadratic program subproblems at each iteration, the total number of quadratic programs solved before reaching a local solution is less than 800 on average for all the test problems. On the other hand, the MIP solver of CPLEX solves one linear programming relaxation problem at each node of the branch-and-bound process for the MIP formulation. From Tables 1 and 2, we see that the number of linear programming problems solved during the branch-and-bound process ranges from 3,853 to 1,605,825, which is significantly larger than the number of quadratic programs solved by BCD methods. Another important observation during our experiment is that the strengthening process for coefficient M_t is quite time consuming, especially when sample size N is large. This could be because N linear problems with at least N^2 variables are solved during the strengthening procedure.

For test problems with large sample size ($N = 1{,}000, 1{,}500, 2{,}000$), we report in Table 3 the numerical results of the BCD methods for (P_m) and (P_c), where f_{CVaR} denotes the objective value of the initial point $\mathbf{x}^0$ obtained from the CVaR approximation. We can see from Table 3 that for both problems (P_m) and (P_c) the BCD methods are able to improve the initial points obtained from the CVaR approximation considerably within a reasonable amount of computing time. Interestingly, we observe that the number of iterations for the BCD methods to converge to local solutions does not exhibit an increasing trend as the number of assets or the number of samples increases.

5. Analysis on Approximation Performance

In this section, we conduct simulation analysis on the approximation performance of the three nonparametric VaR estimators: kernel VaR, quadratic VaR, and historical VaR. The purpose of the simulation analysis is to compare the effectiveness and accuracy of the VaR estimators for the optimal portfolios generated by the two models (P_m) and (P_c).

To see the approximation effect of different VaR estimators, we consider two market scenarios where the asset return follows the following distributions: (1) multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$; and (2) mixed distribution

$$(1 - I(\epsilon))\mathcal{N}(\boldsymbol{\mu}, \Sigma) + I(\epsilon)(Y\mathbf{e} + \mathbf{f}), \tag{22}$$

where $I(\epsilon)$ is a Bernoulli random variable with parameter ϵ, $\mathbf{e}$ is n-dimensional all-one vector, $\mathbf{f}$ is a constant vector, and Y is nonpositive exponential random variable with density

$$p(Y = y) = \begin{cases} \delta e^{\delta y}, & \text{if } y \le 0; \\ 0, & \text{else.} \end{cases}$$

Here we set $\epsilon = 0.05$, $\delta = 0.01$ and constant vector $\mathbf{f} = \boldsymbol{\mu} - \text{diag}(\Sigma)$. The mean $\boldsymbol{\mu}$ and covariance matrix Σ in the above normal distribution and the mixed distribution are estimated by the 2,036 historical daily data of the 460 stocks in Section 4. We point out that the mixed distribution is often used to represent a market scenario with heavy tail return distribution (see, e.g., Lim et al. 2011).

We generate four types of portfolios from (P_m) and (P_c), which were constructed in the same way as in Section 4. More precisely, we generate $\mathbf{x}_m^k$ and $\mathbf{x}_m^q$ from (P_m)

Table 1. Comparison Results of BCD Method and Formulation (MIP) for (P_m)

		BCD			Formulation (MIP)				
n	N	fval	Time	Iter	fval	Time	Node	Gap(%)	imp.ratio(%)
50	100	0.00903	2.5	367	0.00889	404.3	307,040	0.01	−1.6
50	200	0.01008	2.8	361	0.01047	3,600	565,685	73.2	3.7
50	300	0.01322	4.3	364	0.01312	3,600	410,522	78.1	−0.7
50	400	0.01451	5.1	362	0.01444	3,600	128,575	83.4	−0.5
50	500	0.01345	5.9	369	0.01352	3,600	64,898	88.3	0.5
100	100	0.00740	5.2	374	0.00713	416.6	698,430	0.01	−3.7
100	200	0.00867	6.3	393	0.00838	3,600	482,292	68.1	−3.5
100	300	0.00968	7.5	371	0.00962	3,600	178,390	103.4	−0.6
100	400	0.01576	8.1	361	0.01614	3,600	77,724	98.5	2.4
100	500	0.01282	8.7	361	0.01342	3,600	25,479	132.9	4.5
200	200	0.00838	57.1	367	0.00817	3,600	296,515	124.1	−2.6
200	300	0.00948	53.9	387	0.00941	3,600	88,523	130.4	−0.7
200	400	0.01066	57.7	360	0.01102	3,600	21,613	143.9	3.1
200	500	0.00843	58.6	364	0.01012	3,600	6,924	191.3	16.6
300	300	0.00907	53.2	362	0.00897	3,600	37,199	165.3	−1.1
300	400	0.00790	53.2	370	0.00895	3,600	11,533	411.5	11.7
300	500	0.00842	52.2	365	0.01078	3,600	5,834	378.3	21.8
400	400	0.00555	71.8	366	0.00619	3,600	9,960	450.3	10.3
400	500	0.00661	71.6	363	0.00968	3,600	3,853	393.9	31.7

Table 2. Comparison Results of BCD Method and Formulation (MIP) for (P_c)

		BCD			Formulation (MIP)				
n	N	fval	Time	Iter	fval	Time	Node	Gap(%)	imp.ratio(%)
50	100	0.00284	1.76	335	0.00286	4.41	8,874	0.01	−0.6
50	200	0.00307	2.57	353	0.00314	3,600	1,014,622	34.2	−2.2
50	300	0.00294	3.47	365	0.00292	3,600	220,941	81.8	0.6
50	400	0.00114	4.17	363	0.00118	3,600	31,458	76.9	−3.3
50	500	0.00156	3.39	250	0.00143	3,600	15,604	66.2	9.1
100	100	0.00805	4.90	369	0.00811	465.2	1,605,825	0.01	−0.7
100	200	0.00331	4.42	282	0.00347	3,600	812,806	24.1	−4.6
100	300	0.00130	4.65	278	0.00128	3,600	79,945	38.8	1.6
100	400	0.00323	7.21	369	0.00325	3,600	43,006	22.4	−0.6
100	500	0.00151	6.08	276	0.00133	3,600	5,788	46.3	13.5
200	200	0.00275	56.59	317	0.00277	3,600	330,590	31.0	−0.7
200	300	0.00301	64.61	322	0.00302	3,600	38,207	32.7	−0.3
200	400	0.00305	32.79	323	0.00296	3,600	24,598	29.1	3.0
200	500	0.00239	30.29	297	0.00207	3,600	4,286	47.8	15.4
300	300	0.00273	41.35	401	0.00274	3,600	26,174	20.6	−0.3
300	400	0.00229	60.51	304	0.00219	3,600	14,410	33.1	4.5
300	500	0.00285	33.93	338	0.00232	3,600	3,109	72.1	22.8
400	400	0.00204	68.52	344	0.00190	3,600	4,066	48.8	7.3
400	500	0.00199	55.92	282	0.00149	3,600	1,956	82.1	33.6

using the kernel VaR and the quadratic VaR, respectively, and $\mathbf{x}_c^k$ and $\mathbf{x}_c^q$ from (P_c) using the kernel VaR and the quadratic VaR, respectively.

For a VaR estimator $\mathrm{VaR}_{\mathrm{app}}$, we measure its approximation accuracy over the "true" VaR value of $\mathbf{x}$ by the relative error defined by

$$\text{Relative error} = \frac{|\mathrm{VaR}_{\mathrm{app}}(\mathbf{x}) - \mathrm{VaR}^*(\mathbf{x})|}{\mathrm{VaR}^*(\mathbf{x})}(\%),$$

where $\mathrm{VaR}^*(\mathbf{x})$ is estimated by Monte-Carlo simulation with 50,000 samples drawn from the multivariate distribution or the mixed distribution.

5.1. Effect of Bandwidths

We first compare the approximation accuracy of the nonparametric VaR using different bandwidth h. We set the bandwidth interval within the interval of $[0.002, 0.04]$. We choose 20 different values of h from $[0.002, 0.04]$ and generate 100 groups of samples for

Table 3. Numerical Results of BCD Methods for (P_m) and (P_c) with Large Sample Size

		(P_m)				(P_c)			
n	N	f_{CVaR}	fval	Time	Iter	f_{CVaR}	fval	Time	Iter
50	1,000	0.03404	0.03294	7.5	298	0.00065	0.00101	7.3	232
100	1,000	0.02377	0.02227	11.4	336	0.00026	0.00095	7.9	213
200	1,000	0.02600	0.02422	35.6	341	0.00079	0.00167	29.4	251
300	1,000	0.01417	0.01334	59.1	336	0.00104	0.00163	30.9	229
400	1,000	0.01597	0.01505	60.9	305	0.00150	0.00222	61.1	238
50	1,500	0.03432	0.03353	12.3	274	0.00102	0.00143	8.38	202
100	1,500	0.03454	0.03401	14.7	281	0.00043	0.00145	14.9	246
200	1,500	0.02654	0.02496	76.5	399	0.00108	0.00170	29.9	234
300	1,500	0.02605	0.02441	45.6	319	0.00085	0.00127	51.7	247
400	1,500	0.02318	0.02169	64.7	314	0.00132	0.00192	56.5	232
50	2,000	0.03916	0.03777	18.6	276	0.00037	0.00084	16.5	229
100	2,000	0.03302	0.03216	22.8	294	0.00041	0.00107	16.0	212
200	2,000	0.03031	0.02853	50.9	281	0.00080	0.00126	32.8	237
300	2,000	0.02408	0.02348	60.7	303	0.00096	0.00146	46.4	249
400	2,000	0.02348	0.02298	74.8	292	0.00097	0.00152	52.8	213

each value, while each group contains 200 samples from the multivariate distribution and the mixed distribution, respectively.

Figures 2 and 3 illustrate the average relative errors of the kernel VaR and quadratic VaR for the four different portfolios when the 100 groups of samples are drawn from the normal distribution and the mixed distribution. We see that when the bandwidth h is between 0.002 and 0.03, the kernel VaR tends to give smaller relative error than quadratic VaR. However, when $h \geq 0.03$, the relative error of the kernel VaR increases dramatically. This suggests that while the kernel VaR can give good approximation if the bandwidth is chosen properly, it is more sensitive to the bandwidth than the quadratic VaR.

5.2. Effect of Sample Sizes

Next, we compare approximation performance of the two nonparametric VaRs with the historical VaR $V(\mathbf{x})_{[p]}$ ($p = \lceil \alpha N \rceil$) and the parametric VaR estimator calculated by assuming that asset returns follow a normal distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ (although this may not be true), which is denoted by $\mathrm{VaR}^n_\alpha(\mathbf{x})$. It is well known that $\mathrm{VaR}^n_\alpha(\mathbf{x})$ has the following closed-form expression in such a case:

$$\mathrm{VaR}^n_\alpha(\mathbf{x}) = z_\alpha \sqrt{\mathbf{x}^T \Sigma \mathbf{x}} - \boldsymbol{\mu}^T \mathbf{x}, \tag{23}$$

where $z_\alpha = -\Psi^{-1}(1-\alpha)$ with Ψ being the cumulative standard normal distribution function. Since $\Psi^{-1}(1-\alpha) < 0$ when $\alpha \in (0.5, 1)$, we have $z_\alpha > 0$. In practice, the parametric VaR is calculated according to (23) with Σ and $\boldsymbol{\mu}$ estimated by the samples.

Figures 4 and 5 show the relative errors of the VaR estimators when varying the sample size. From Figure 4, we see that the parametric VaR, $\mathrm{VaR}^n_\alpha(\mathbf{x})$, has the smallest error as the sample size increases, which is reasonable as the samples were drawn from the normal distributions. However, we observe from Figure 5 that $\mathrm{VaR}^n_\alpha(\mathbf{x})$ is not preferable and often gives bigger relative errors than the other three VaR estimators because of the wrong distribution assumption. We also observe from Figures 4 and 5 that the historical VaR tends to have similar approximation accuracy as the kernel VaR and the quadratic VaR, but is unstable when sample size is small. The kernel VaR and quadratic VaR, on the other hand, appear to be more robust for samples from both the normal distribution and the mixed distribution in terms of the relative errors. From our simulation, one more point might deserve mentioning: no matter which estimator is adopted, the risk measure seems unstable when there are not enough data points. In this case, the so-called *robust* portfolio selection approach using worst-case analysis can possibly add value (see Zhu et al. 2015 and the references therein).

6. Study on Investment Performance

In this section, we conduct in-sample simulation analysis and an out-of-sample empirical study on the performance of portfolios generated by the two models (P_m) and (P_c) using nonparametric VaR. The purpose of this section is to evaluate the effects of using nonparametric VaR in the mean-risk models in terms of in-sample performance and out-of-sample performance.

6.1. In-Sample Analysis

In this section, we carry out in-sample analysis of the portfolios generated by different mean-VaR models via

Figure 2. (Color online) Average Relative Error of Kernel VaR and Quadratic VaR with Different Bandwidth h for Samples Drawn from Normal Distribution

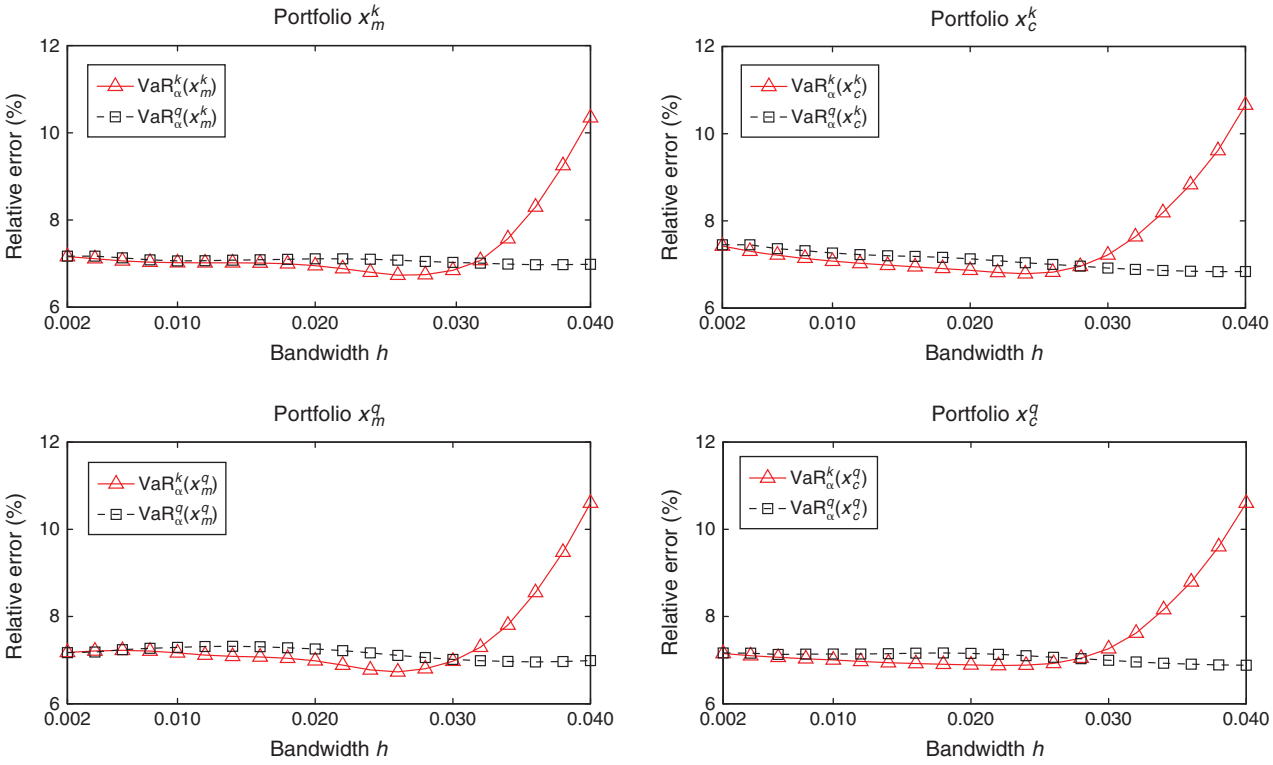


Figure 3. (Color online) Average Relative Error of Kernel VaR and Quadratic VaR with Different Bandwidth h for Samples Drawn from Mixed Distribution

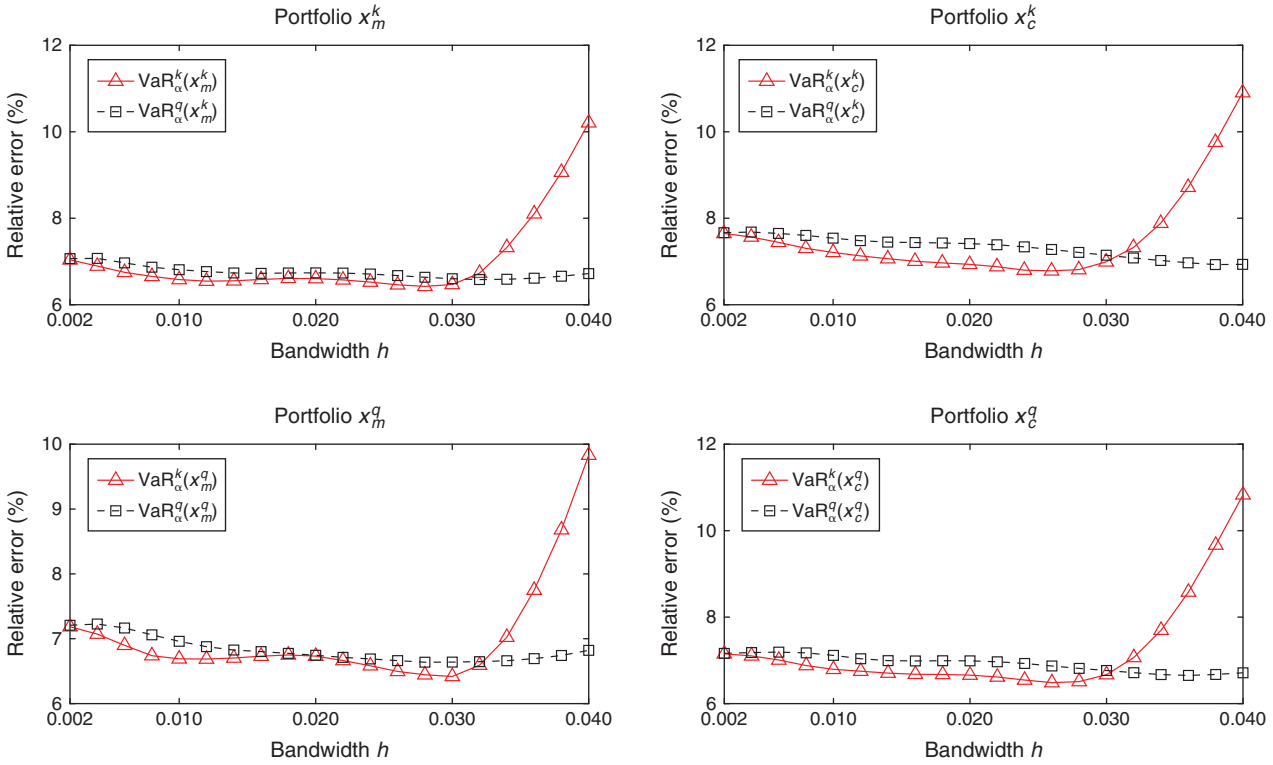


Figure 4. (Color online) Relative Error of Different VaR Estimators with Varying Sample Size Under Normal Distribution

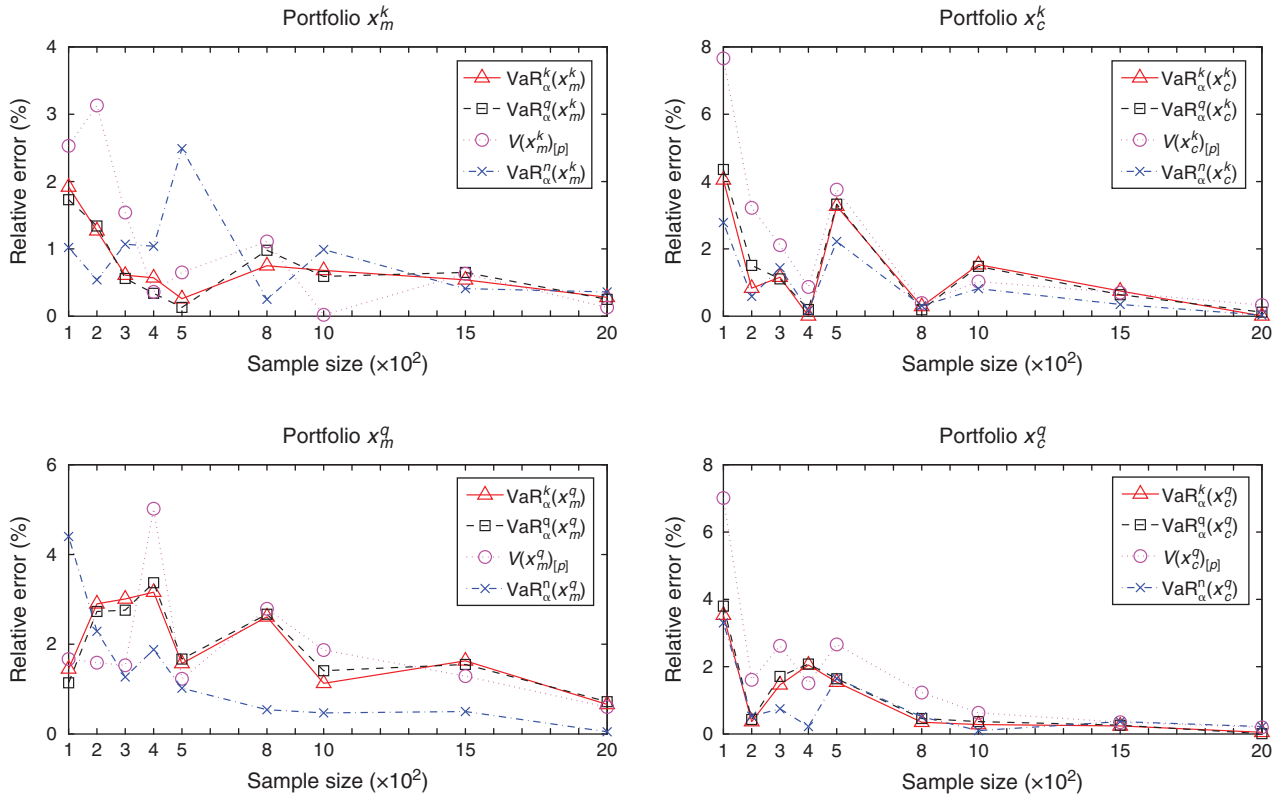
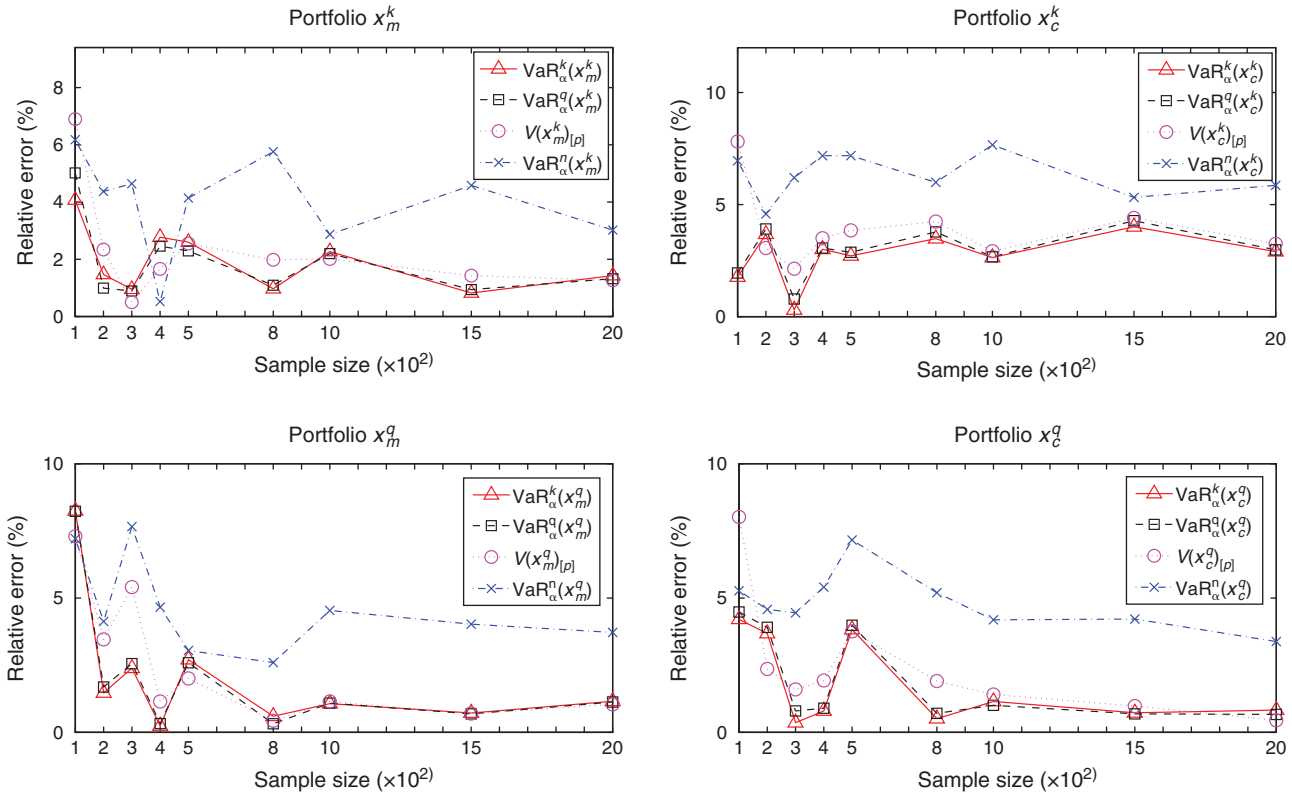


Figure 5. (Color online) Relative Error of Different VaR Estimators with Varying Sample Size Under Mixed Distribution



comparing the mean-VaR efficient frontiers. We consider the following four different portfolio selection models:

(1) *Mean-kernel VaR* models (M_1): $\min_{\mathbf{x}\in\Omega} \sum_{i=1}^{N} w_i \cdot V(\mathbf{x})_{[i]}$, where w_i is calculated by (4).

(2) *Mean-quadratic VaR* models (M_2): $\min_{\mathbf{x}\in\Omega} \sum_{i=1}^{N} u_i \cdot V(\mathbf{x})_{[i]}$, where u_i is calculated by (7).

(3) *Mean-historical VaR* models (M_3): $\min_{\mathbf{x}\in\Omega} V(\mathbf{x})_{[p]}$, where $p = \lceil \alpha N \rceil$.

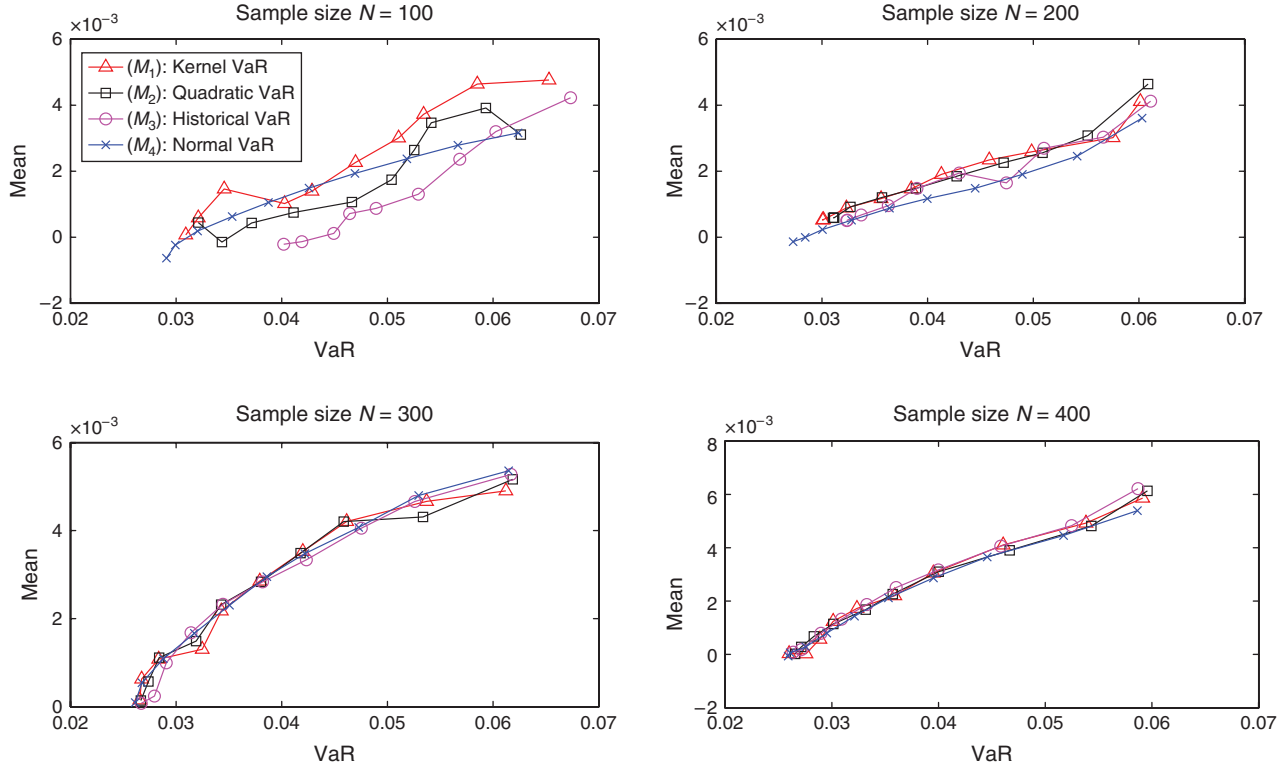
(4) *Mean-normal VaR* model (M_4): $\min_{\mathbf{x}\in\Omega} \mathrm{VaR}_{\alpha}^{n}(\mathbf{x})$, where asset returns are assumed to follow normal distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$.

From (23), we see that model (M_4) can be reduced to a second-order cone programming (SOCP) problem (see Lobo et al. 1998) by introducing an additional variable $\gamma = \mathrm{VaR}_{\alpha}^{n}(\mathbf{x})$ and an SOCP constraint: $z_{\alpha}\sqrt{\mathbf{x}^T\Sigma\mathbf{x}} - \boldsymbol{\mu}^T\mathbf{x} \le \gamma$.

We assume that the asset returns follow the mixed distribution (22), where the mean $\boldsymbol{\mu}$ and covariance matrix Σ are estimated by 473 weekly returns data of the 460 constituents of Standard and Poor's 500 index from January 2004 to January 2013. We then randomly generate samples of asset returns from the mixed distribution with sample size $N = 100, 200, 300, 400$ and build up the four models (M_1)–(M_4), respectively. We set $\mathcal{X} = \{\mathbf{x} \in \Re^n \mid \sum_{i=1}^{n} x_i = 1, 0 \le x_i \le 0.5, i = 1, \ldots, n\}$. By setting different return levels of ρ, we solve these three nonconvex models, (M_1)–(M_3), by Algorithm 2 and the convex model (M_4) by the QCP solver in CPLEX, and obtain a group of portfolios to generate the mean-VaR efficient frontiers for all these individual models. In calculating the efficient frontiers, the mean values of the portfolios are calculated by $\boldsymbol{\mu}^T\mathbf{x}$, while the VaR values of the portfolios are computed by Monte Carlo simulation using 50,000 samples drawn from the mixed distribution (22).

The mean-VaR efficient frontiers generated by the four models are illustrated in Figure 6. From the figure, we can see that when the sample size is small ($N = 100$ or $N = 200$), the two portfolio selection models using the kernel and quadratic VaRs appear to generate portfolios with better mean-VaR pairs than the models using the historical VaR and the parametric normal VaR in most cases. However, for relatively larger sample sizes of $N = 300$ and $N = 400$, the difference between the four efficient frontiers becomes smaller and there is no obvious dominance relation among these four efficient frontiers. This suggests that when a large number of samples is available, these four models tend to give similar portfolios, but when there are limited historical data or valid samples, the two nonparametric VaR-based models outperform the historical VaR-based model and the parametric normal VaR-based model. It is worth pointing out that constructing portfolio selection models using only recent historical data is reasonable since the market situations could be

Figure 6. (Color online) Efficient Frontiers of Portfolio Selection Models (M_1)–(M_4) with Different Sample Size



such that only the recent data are relevant. This makes the mean-kernel VaR model or mean-quadratic VaR model advantageous when the number of valid data, especially weekly or monthly historical data, is limited.

6.2. Out-of-Sample Analysis

In this section, we conduct out-of-sample analysis for the portfolios generated by the four portfolio selection models (M_1)–(M_4) using back testing strategy. Again, we use the 473 weekly returns data of the 460 constituents of Standard & Poor's 500 Index from February 2003 to February 2012. We choose the 52 weeks from February 2011 to February 2012 as the out-of-sample period. We set the initial portfolio value equal to 1 at the beginning of the period. At the end of each week, we calculate the values of the four portfolios and update the four portfolios by resolving (M_1)–(M_4) with updated parameters computed with the new data set. More precisely, at the end of each week, we calculate the parameters of these four models and resolve the four models in the same way as in the in-sample analysis using the most recent 200 weekly return data before that week.

Figures 7–9 illustrate the evolution of the portfolio values of the portfolios generated by the four models under different prescribed return levels during the out-of-sample period using the weekly rebalancing strategy. We observe from these figures that when the required return level is $\rho = 0.1\%$ or 0.4%, both (M_1) and (M_2) are able to generate portfolios with higher portfolio values during the out-of-sample period than (M_3) and (M_4), while (M_1), which uses the kernel VaR risk measure, appears to have the best performance among the four models. For the case $\rho = 0.7\%$, (M_1) still constructs portfolios with the best portfolio value and return, while model (M_4) with VaR under the normal distribution gives slightly lower portfolio value. We report in Table 4 statistical results of the four portfolios during the out-of-sample period, including the final portfolio value (FinVal), average return (R_{ave}), maximum return (R_{max}), minimum return (R_{min}), standard deviation (Std), and the ratio of average return over standard deviation R_{ave}/Std. Notice that these statistics are calculated with the weekly realized returns of the strategies during the out-of-sample period. We mark the best-performing portfolios under different criteria in bold fonts. The statistical results show that the portfolio selection model (M_1) using the kernel VaR has the best performance in terms of the final value, average return, and the ratio of average return over the standard deviation under all required return levels. Meanwhile, model (M_4) usually generates portfolios with the smallest volatilities among the four models. Model (M_3), which uses the historical VaR, gives the smallest ratio of average return over standard deviation. This could be mainly because historical VaR is more sensible to the tail information than nonparametric VaRs when the sample size is small.

Figure 7. (Color online) Evolution of Portfolio Values for Different Portfolio Selection Models During Out-of-Sample Period: $\rho = 0.1\%$

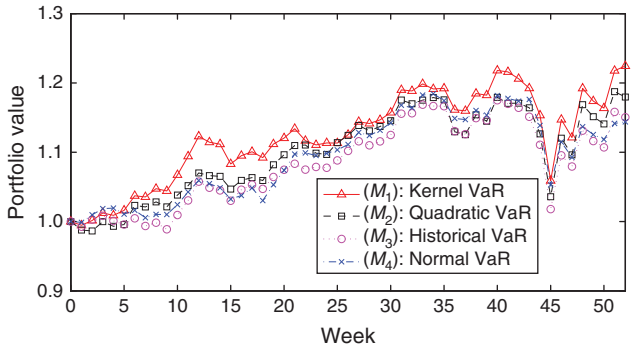


Figure 8. (Color online) Evolution of Portfolio Values for Different Portfolio Selection Models During Out-of-Sample Period: $\rho = 0.4\%$

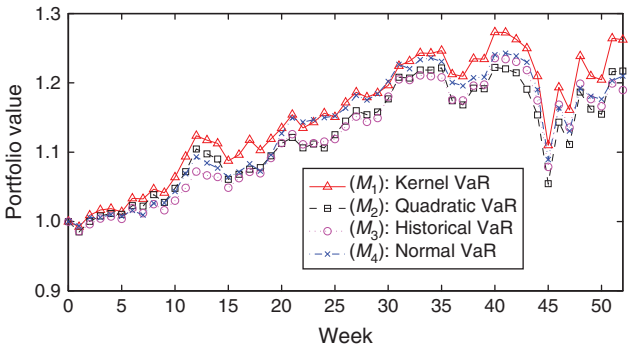
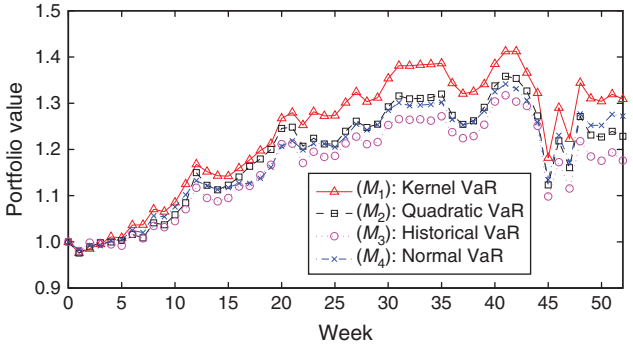


Figure 9. (Color online) Evolution of Portfolio Values for Different Portfolio Selection Models During Out-of-Sample Period: $\rho = 0.7\%$



7. Conclusion

In this paper, we have investigated the adoption of nonparametric VaR in mean-VaR portfolio selection

Table 4. Statistical Results of Portfolio Values and Returns During Out-of-Sample Period

ρ (%)	Model	FinVal	R_{ave}(%)	R_{max}(%)	R_{min}(%)	Std(%)	R_{ave}/Std
0.1	(M_1): kernel VaR	**1.22**	**0.42**	**8.32**	−8.17	2.37	**0.1772**
	(M_2): quadratic VaR	1.18	0.34	8.15	−8.05	2.36	0.1441
	(M_3): historical VaR	1.15	0.29	7.62	−8.36	2.23	0.1300
	(M_4): normal VaR	1.14	0.27	5.87	**−7.52**	**1.89**	0.1429
0.4	(M_1): kernel VaR	**1.26**	**0.48**	7.53	−8.23	2.41	**0.1981**
	(M_2): quadratic VaR	1.22	0.41	**8.41**	−8.59	2.54	0.1616
	(M_3): historical VaR	1.18	0.36	8.32	**−8.15**	2.29	0.1571
	(M_4): normal VaR	1.21	0.39	6.84	−8.46	**2.15**	0.1816
0.7	(M_1): kernel VaR	**1.31**	**0.57**	**9.94**	−10.60	3.11	**0.1823**
	(M_2): quadratic VaR	1.22	0.45	9.55	−11.74	3.19	0.1400
	(M_3): historical VaR	1.18	0.36	9.18	−12.21	3.03	0.1180
	(M_4): normal VaR	1.27	0.50	8.96	**−9.82**	**2.78**	0.1804

models. Our main motivation is to develop an efficient solution methodology for the portfolio selection models using nonparametric VaR, as a nonparametric method is robust in VaR calculation. By exploiting the special structure of the nonconvex optimization problems resulting from the mean-kernel VaR and the mean-quadratic VaR models, we have developed some efficient block coordinate descent methods. Numerical results reveal that the proposed BCD methods are capable of finding good local solutions for large-scale problems and compare favorably with the branch-and-bound method-based global optimization procedure. We have also conducted simulation and empirical analysis to evaluate the in-sample and out-of-sample performance of nonparametric VaRs. Our empirical results suggest that the kernel VaR and quadratic VaR are promising to serve as robust risk measures in the mean-risk portfolio selection modeling.

Acknowledgments

The fifth author is also grateful to the support from the Patrick Huen Wing Ming Chair Professorship of Systems Engineering and Engineering Management. The authors very much appreciate the two anonymous reviewers and the associate editor for their constructive and insightful comments that substantially helped improve the paper.

References

Alexander GJ, Baptista AM (2004) A comparison of VaR and CVaR constraints on portfolio selection with the mean-variance model. *Management Sci.* 50(9):1261–1273.

Artzner P, Delbaen F, Eber JM, Heath D (1999) Coherent measures of risk. *Math. Finance* 9(3):203–228.

Benati S, Rizzi R (2007) A mixed integer linear programming formulation of the optimal mean/value-at-risk portfolio problem. *Eur. J. Oper. Res.* 176(1):423–434.

Bertsekas DP (1999) *Nonlinear Programming* (Athena Scientific, Belmont, MA).

Bickel PJ (1967) Some contributions to the theory of order statistics. *Proc. Fifth Berkeley Sympos. Math. Statist. Probab., Volume* 1: *Statist.* (University California Press, Berkeley, CA).

Bonami P, Lejeune MA (2009) An exact solution approach for portfolio optimization problems under stochastic and integer constraints. *Oper. Res.* 57(3):650–670.

Boyd S, Vandenberghe L (2004) *Convex Optimization* (Cambridge University Press, Cambridge, UK).

Butler JS, Schachter B (1998) Estimating value-at-risk with a precision measure by combining kernel estimation with historical simulation. *Rev. Deriv. Res.* 1(4):371–390.

Chang YP, Hung MC, Wu YF (2003) Nonparametric estimation for risk in value-at-risk estimator. *Commun. Stat. Simul. Comput.* 32(4):1041–1064.

Chen SX, Tang CY (2005) Nonparametric inference of value-at-risk for dependent financial returns. *J. Financial Econom.* 3(2): 227–255.

Cheng MY, Peng L (2002) Regression modeling for nonparametric estimation of distribution and quantile functions. *Stat. Sinica* 12(12):1043–1060.

Clarke FH (1983) *Optimization and Nonsmooth Analysis* (Wiley, New York).

Cui XT, Zhu SS, Sun XL, Li D (2013) Nonlinear portfolio selection using approximate parametric value-at-risk. *J. Banking Finance* 37(6):2124–2139.

Duffie D, Pan J (1997) An overview of value at risk. *J. Derivatives* 4(3):7–49.

Gaivoronski AA, Pflug G (2005) Value at risk in portfolio optimization: Properties and computational approach. *J. Risk* 7(2): 1–31.

Goldstein T, Osher S (2009) The split Bregman method for ℓ_1-regularized problems. *SIAM J. Imaging Sci.* 2(2):323–343.

He BS, Tao M, Yuan XM (2012) Alternating direction method with Gaussian back substitution for separable convex programming. *SIAM J. Optim.* 22(2):313–340.

Heyde CC, Kou SG (2004) On the controversy over tailweight of distributions. *Oper. Res. Lett.* 32(5):399–408.

J. P. Morgan (1996) RiskMetrics™. Technical report, J. P. Morgan Company, New York.

Jorion P (2007) *Value at Risk: The New Benchmark for Managing Financial Risk* (McGraw-Hill, New York).

Kou S, Peng X, Heyde CC (2013) External risk measures and basel accords. *Math. Oper. Res.* 38(3):393–417.

Li Q, Racine JS (2007) *Nonparametric Econometrics: Theory and Practice* (Princeton University Press, Princeton, NJ).

Lim AEB, Shanthikumar JG, Vahn GY (2011) Conditional value-at-risk in portfolio optimization: Coherent but fragile. *Oper. Res. Lett.* 39(3):163–171.

Linsmeier TJ, Pearson ND (2000) Value at risk. *Financial Analysts J.* 56(2):47–67.

Lobo MS, Vandenberghe L, Boyd S, Lebret H (1998) Applications of second-order cone programming. *Linear Algebra Appl.* 284(1–3): 193–228.

Luedtke J (2014) A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Math. Programming* 146(1–2):219–244.

Mausser H, Rosen D (1999) Beyond VaR: From measuring risk to managing risk. *ALGO Res. Quart.* 1:5–20.

Parzen E (1979) Nonparametric statistical data modeling. *J. Amer. Statist. Assoc.* 74(365):105–121.

Qiu F, Ahmed S, Dey SS, Wolsey LA (2014) Covering linear programming with violations. *INFORMS J. Comput.* 26(3): 531–546.

Rockafellar RT, Uryasev S (2000) Optimization of conditional value-at-risk. *J. Risk* 2(1):21–42.

Rockafellar RT, Uryasev S (2002) Conditional value-at-risk for general loss distributions. *J. Banking Finance* 26(7):1443–1471.

Sheather SJ, Marron JS (1990) Kernel quantile estimators. *J. Amer. Statist. Assoc.* 85:410–416.

Shen Y, Wen ZW, Zhang Y (2014) Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization. *Optim. Methods Softw.* 29(2):239–263.

Wen ZW, Peng XH, Liu X, Bai XD, Sun XL (2013) Asset allocation under the Basel Accord risk measures. Technical report, http://www.optimization-online.org/DB_FILE/2013/01/3730.pdf.

Xu YY, Yin WT, Wen ZW, Zhang Y (2012) An alternating direction algorithm for matrix completion with nonnegative factors. *Front. Math. China* 7(2):365–384.

Yang SS (1985) A smooth nonparametric estimator of a quantile function. *J. Amer. Statist. Assoc.* 80(392):1004–1011.

Yao HX, Li ZF, Lai YZ (2013) Mean-CVaR portfolio selection: A nonparametric estimation framework. *Comput. Oper. Res.* 40(4): 1014–1022.

Yin W, Osher S, Goldfarb D, Darbon J (2008) Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. *SIAM J. Imaging Sci.* 1(1):143–168.

Zhu SS, Ji XD, Li D (2015) A robust set-valued scenario approach for handling modeling risk in portfolio optimization. *J. Comput. Finance* 19(1):11–40.