Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Transaction Costs-Aware Portfolio Optimization via Fast Löwner-John Ellipsoid Approximation

Weiwei Shen[†] **and Jun Wang**[‡]
[†]GE Global Research Center, Niskayuna, NY, USA
[‡]Institute of Data Science and Technology, Alibaba Group, Seattle, WA, USA
weiwei.shen@ge.com, j.wang@alibaba-inc.com

Abstract

Merton's portfolio optimization problem in the presence of transaction costs for multiple assets has been an important and challenging problem in both theory and practice. Most existing work suffers from curse of dimensionality and encounters with the difficulty of generalization. In this paper, we develop an approximate dynamic programing method of synergistically combining the Löwner-John ellipsoid approximation with conventional value function iteration to quantify the associated optimal trading policy. Through constructing Löwner-John ellipsoids to parameterize the optimal policy and taking Euclidean projections onto the constructed ellipsoids to implement the trading policy, the proposed algorithm has cut computational costs up to a factor of five hundred and meanwhile achieved near-optimal risk-adjusted returns across both synthetic and real-world market datasets.

Introduction

Dynamic portfolio choice problems have taken on increasing importance in finance industry such as mutual funds, endorsements and private wealth management (Brandt 2010). The seminal work of Merton (Merton 1969) shows that investors who wish to maximize the utility of their final wealth should always hold a constant fraction of total wealth on each asset. However, implementing such a strategy requires rebalancing continually as assets prices fluctuate, and therefore will lead to high or even infinite transaction costs. Since then researchers have tried to address this issue by solving Merton's portfolio problem in the presence of transaction costs. Thereinto, the proportional transaction costs model, as a suitable model for brokerage commissions and bid-ask spread costs, typifies the common situation for normal investors (Brandt 2010; Cvitanic 2001; Davis and Norman 1990).

Portfolio optimization with proportional transaction costs is naturally modeled as a multistage stochastic program. But the stochastic program framework limits the size of solvable problems due to the exponential growth of scenarios with investment periods. Thus, researchers often recast it to a dynamic program. However, the associated dynamic program with continuous decision and state spaces is still difficult as its computational costs increase exponentially with the number of involved assets, namely the *curse of dimensionality* (Cai and Judd 2014; Rust 2008). Therefore, most of the relevant studies focus on simple settings with only one or two risky assets. Representative methods include solving the associated Hamilton-Jacobi-Bellman (HJB) partial differential equation to achieve optimal solutions, which generally fails to scale up to higher dimensions (Davis and Norman 1990; Magill and Constantinides 1976; Shreve and Soner 1994; Muthuraman and Kumar 2006).

To numerically solve this type of dynamic programming problems, one can appeal to value function iteration (VFI) or policy parameterization (Powell 2007; Rust 2008). However, in a high-dimensional space, conventional VFI becomes intractable and suffers from error amplification in iteration. Although the policy parameterization consumes cheaper computing power, it falls short on accuracy due to the complexity of policy structures in high dimensions.

Realizing the limitations of the existing work, in this paper we propose an approximate dynamic programming (ADP) method to tackle the multi-asset portfolio optimization problems with proportional transaction costs in a discrete time and finite horizon setting. Through synergistically combining the VFI framework with policy parameterization, the proposed ADP method enjoys complementary advantages of low approximation errors from VFI and high computational efficiency from policy parameterization. Briefly, the components from VFI pave the way for effectively parameterizing a complex policy in a high-dimensional space; the components from policy parameterization provide a pathway to efficiently evaluating the strategy and bypassing the issue of error amplification. In particular, by adopting the Löwner-John ellipsoid approximation to the optimal policy, we are able to address the conundrum of parameterizing a complex policy through depicting its geometric shape and location in a high-dimensional state space. Once the parameterized policy is available, the determination of rebalancing is simply conducted as an Euclidean projection process onto the estimated ellipsoids. Hence, the computational costs are drastically reduced and meanwhile the generated strategy is robust and sufficiently flexible to capture complex features of the true policy for multiple assets. Further, to validate the proposed method, across various parameter settings we compare our strategy with several state-of-the-art strategies on both synthetic and real-world datasets. The results clearly illustrate the superiority of the proposed approach in both risk-adjusted returns and computational costs. Besides, this new methodology of handling dynamic programs with continuous decision and states spaces can be explored and exploited to other applications.

Background and Related Work

In this section, we first succinctly introduce the classic portfolio problem by Merton (Merton 1969; 1971). We then expand the discussion of the relevant work on portfolio optimization with proportional transaction costs and approximate dynamic programming. We will inherit the notations defined in this section and incorporate the transaction costs model into Merton's portfolio problem in the next section.

Merton's Portfolio Problem

Merton's portfolio problem is developed in a frictionless market. In a discrete time and finite horizon setting, the trading periods consist of $t_k = k\Delta t$, $k = 0, \ldots, m$, where the initial time is $t_0 = 0$ and the terminal time is $t_m = m\Delta t = T$. For simplicity, we use k as the time index to indicate the trading period at time t_k hereafter. We assume that investors with the constant relative risk aversion (CRRA) utility have access to one risk-free (cash account) and n risky assets (stocks). The risk-free asset pays a gross risk-free return R_f. The risky assets pay a stochastic return $\mathbf{R}_k = (R_{k,1}, \ldots, R_{k,i}, \ldots, R_{k,n})^\top$, where $R_{k,i}$ is the gross return of the i-th asset from time t_{k-1} to t_k. The stochastic return is modeled by a multivariate geometric Brownian motion as

$$\ln \mathbf{R}_k = \boldsymbol{\mu} - \frac{1}{2}\boldsymbol{\sigma}^2 + \boldsymbol{e}_k, \tag{1}$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_i, \ldots, \mu_n)^\top \in \mathbb{R}^n$ is the asset return vector, $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_i, \ldots, \sigma_n)^\top \in \mathbb{R}^n_+$ is the return volatility vector, and $\mathbf{e}_k \sim \mathcal{N}(0, \Sigma_e)$ is the stochastic increment from a multivariate normal distribution with the covariance matrix $\Sigma_e \in \mathbb{R}^{n\times n}$.[1]

Further, the positions on the risk-free and risky assets at time t_k are denoted by x_k and $\mathbf{y}_k = (y_{k,1}, \ldots, y_{k,i}, \ldots, y_{k,n})^\top$, respectively. In time, investors can either spend money from the risk-free account to buy risky assets or add money to the risk-free account by selling risky assets. At time t_k, denote $\mathbf{L}_k = (L_{k,1}, \ldots, L_{k,i}, \ldots, L_{k,n})^\top$ as an n-vector where $L_{k,i}$ represents the amount of money spent from the risk-free account to buy the i-th risky asset. Similarly, denote $\mathbf{U}_k = (U_{k,1}, \ldots, U_{k,i}, \ldots, U_{k,n})^\top$ as an n-vector where $U_{k,i}$ represents the amount of money received from selling the i-th risky asset. Given a policy $(\mathbf{U}_k, \mathbf{L}_k)$ at time t_k, the evolution of the dollar positions of risk-free and risky assets from time t_k to t_{k+1} is modeled by the following *transition equation*:

$$\begin{pmatrix} x_{k+1} \\ \mathbf{y}_{k+1} \end{pmatrix} = \begin{pmatrix} R_f \left(x_k + \sum_{i=1}^n (U_{k,i} - L_{k,i})\right) \\ \mathbf{R}_{k+1} \circ (\mathbf{y}_k + \mathbf{L}_k - \mathbf{U}_k) \end{pmatrix}, \tag{2}$$

where $\circ$ denotes the element-wise product of two vectors. Also, to prohibit short selling and borrowing, which is the typical case for normal investors, the trades at time t_k are restricted to a convex set, i.e., a solvency set:

$$\begin{aligned} \mathcal{C}_k = \{\mathbf{U}_k, \mathbf{L}_k \in \mathbb{R}^n_+ : x_k + \sum_{i=1}^n (U_{k,i} - L_{k,i}) \geq 0, \\ \mathbf{y}_k + \mathbf{L}_k - \mathbf{U}_k \succeq \mathbf{0}\}. \end{aligned} \tag{3}$$

Furthermore, the wealth W_k at time t_k is the sum of the dollar positions across the risk-free and risky assets, i.e.,

$$W_k(x_k, \mathbf{y}_k) = x_k + \sum_{i=1}^n y_{k,i}. \tag{4}$$

The objective of investors is to choose $(\mathbf{U}_k, \mathbf{L}_k)$ at each time t_k for $k = 0, \ldots, m-1$ to maximize the expected utility of the wealth at the final time $t_m = T$:

$$\max_{\substack{(\mathbf{U}_k, \mathbf{L}_k) \in \mathcal{C}_k \\ k=0,\ldots,m-1}} \mathbb{E}[\mathcal{U}(W_m)], \tag{5}$$

where the CRRA power utility function characterizes investors' risk tolerance as

$$\mathcal{U}(x) = \frac{x^\gamma}{\gamma}, \quad \gamma < 1, \quad \gamma \neq 0, \tag{6}$$

with γ as the degree of relative risk aversion.

To solve the multistage stochastic program (5), Merton first recasts it to a dynamic program with *state variables* $(x_k, \mathbf{y}_k)$ and *control variables* $(\mathbf{U}_k, \mathbf{L}_k)$ and then obtains a closed-form solution to the optimal trades that investors should always execute in time (Merton 1969; 1971).

Related Work

With transaction costs even in a simple proportional form, the above portfolio problem becomes more challenging and has no closed-form solution. Generally, the optimal policy to a stochastic program is a full-fledged characterization of the solution. To this end, many papers report results of the optimal policy for transaction costs problems. For the simplest case with one risky asset, a no-trading region serving as the optimal trading policy is conjectured, confirmed and analyzed in (Magill and Constantinides 1976; Davis and Norman 1990) and (Shreve and Soner 1994). They find that if the position of the risky asset lies within this region, the optimal policy is not to trade; if it lies outside, the optimal policy is to bring the position back to this region. Intuitively, trades should only take place if they can bring sufficient benefits to cover costs. Therefore, obtaining and understanding such a no-trading region is the crux. For

[1] This return model is originally adopted in Merton's portfolio problem (Merton 1969; 1971). Although a vast number of other return models are proposed to improve the predictability, no consensus has been achieved and this return model is still the building block for new methodologies and testing (Rapach and Zhou 2012).

example, (Muthuraman and Kumar 2006) and (Lynch and Tan 2010) identify a quadrilateral no-trading region for a two risky assets problem. However, the predominant difficulty in numerically solving the multi-asset problems roots in the lack of effective techniques to handle the *curse of dimensionality* (Rust 2008). Although computing and analyzing the optimal policy for cases with a few risky assets drives deep insights (Goodman and Ostrov 2010), the idea of utilizing policy structures to design efficient algorithms for high-dimensional problems is less investigated. Meanwhile, machine learning researchers focus on designing empirical algorithms to incorporate market signals (Agarwal et al. 2006; Borodin, El-Yaniv, and Gogan 2004; Dirk and Peter 2001; Kalai and Vempala 2003; Li and Hoi 2012; Shen, Wang, and Ma 2014; Moody and Saffell 1999; Neuneier 1996).

On the other hand, ADP is aiming at developing practical and high-quality approximated solutions when dynamic programs are hard to solve exactly (Powell 2007). It has previously been applied in various portfolio related applications (Bhat, Moallemi, and Farias 2012; Brandt et al. 2005; Garlappi and Skoulakis 2010; Lynch and Tan 2010). In particular, applying ADP in VFI starts with approximating value functions locally by kernel methods (Bhat, Moallemi, and Farias 2012; Garlappi and Skoulakis 2010; Lynch and Tan 2010) or globally by regression techniques (Brandt et al. 2005; Longstaff and Schwartz 2001). The approximation process is then repeated backward from the second-to-last period to the first period. In each period, the solution is found by maximizing the one-step ahead expectation of the approximated value function derived in the previous iteration. The multi-asset portfolio optimization problem with proportional transaction costs can be naturally formulated as a dynamic program with continuous decision and state spaces in a high dimension. However, applying the above algorithm to this type of dynamic programs remains challenging because accurate results require rigorous error control to reduce error amplification in backward iteration. More detailed illustrations and discussions over a wide range of applications and conclusions about ADP and VFI can be found in (Bertsekas 2011; Cai and Judd 2014; Judd, Maliar, and Maliar 2011; Powell 2007; Rust 2008), and the extensive references therein.

Methodology

In this section, we first formulate the portfolio optimization problem with proportional transaction costs. Then we will present a fast ADP solution that combines the Löwner-John ellipsoid approximation with the conventional VFI framework to ameliorate the difficulty in high-dimensional problems. Finally we will compute lower bounds for the problem and complement them with upper bounds.

Formulation

We adopt the same notations, including the discrete time and finite horizon setting, the objective function, and the return dynamics from the previous section. In a frictional market, let $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_i, \ldots, \beta_n)^\top \in \mathbb{R}_+^n$ be the factor of the transaction costs for buying and selling. Specifically, buying the i-th risky asset priced at $L_{k,i}$ will cost $(1+\beta_i)L_{k,i}$ in the risk-free account; selling the i-th risky asset priced at $U_{k,i}$ will add a dollar amount of $(1-\beta_i)U_{k,i}$ to the risk-free account. By incorporating the proportional transaction costs into the evolution of the positions (2), the new transition equation becomes:

$$\begin{pmatrix} x_{k+1} \\ \mathbf{y}_{k+1} \end{pmatrix} = \begin{pmatrix} R_f x_k^+ \\ \mathbf{R}_{k+1} \circ \mathbf{y}_k^+ \end{pmatrix}, \tag{7}$$

where x_k^+ stands for the post-trade risk-free position $x_k - \sum_{i=1}^n [(1+\beta_i)L_{k,i} - (1-\beta_i)U_{k,i}]$ and $\mathbf{y}_k^+$ represents the post-trade risky position $\mathbf{y}_k + \mathbf{L}_k - \mathbf{U}_k$. Accordingly, the new solvency set is:

$$\mathcal{C}_k = \{\mathbf{U}_k, \mathbf{L}_k \in \mathbb{R}_+^n : \ x_k^+ \geq 0, \mathbf{y}_k^+ \succeq \mathbf{0}\}. \tag{8}$$

Similar to Merton's portfolio problem, this stochastic program can be casted as a dynamic program with *state variables* $(x_k, \mathbf{y}_k)$ and *control variables* $(\mathbf{U}_k, \mathbf{L}_k)$. In particular, denote the *value function* at time t_k as the maximized expected utility of the final wealth by selecting a sequence of control variables $(\mathbf{U}_\tau, \mathbf{L}_\tau)$ for $\tau = k, \ldots, m-1$:

$$V_k(x_k, \mathbf{y}_k) = \max_{\substack{(\mathbf{U}_\tau, \mathbf{L}_\tau) \in \mathcal{C}_\tau \\ \tau = k, \ldots, m-1}} \mathbb{E}_k[\mathcal{U}(W_m)], \tag{9}$$

where the investment positions follow the transition equation (7) and the symbol $\mathbb{E}_k[\cdot]$ represents the conditional expectation conditioned on the information up to time t_k. For a fixed level of the wealth W_k, equation (9) shows that the value function V_k solely depends on the state variable $(x_k, \mathbf{y}_k)$.[2] Further, by Bellman's principle of optimality (Bertsekas 2011), equation (9) is equivalent to a series of one-period iteration problems:

$$V_k(x_k, \mathbf{y}_k) = \max_{(\mathbf{U}_k, \mathbf{L}_k) \in \mathcal{C}_k} \mathbb{E}_k[V_{k+1}(x_{k+1}, \mathbf{y}_{k+1})]. \tag{10}$$

To attack (10), a typical method starts with the terminal condition of the value function $V_m = \mathcal{U}(W_m)$ and performs backward value function iteration to compute earlier value functions V_k from time t_{m-1} to t_0. Since the optimization problem (10) has no closed-form solution, we appeal to an ADP approach to numerically conduct VFI.

VFI with Löwner-John Ellipsoid Approximation

Conventional VFI boils down to finding accurate functional approximation to the value function $V_k(x_k, \mathbf{y}_k)$ at each time step. In particular, the typical process consists of three key steps. First, it needs to construct a grid of the states $(x_k, \mathbf{y}_k)$ in a multidimensional space by taking the tensor-product of discretized points in each dimension. Second, it solves the corresponding optimization problem (10) for each grid point $(x_k, \mathbf{y}_k)$, where it numerically computes the expectation on the right hand side of (10) by a multidimensional Gauss-Hermit quadrature rule. Third, given the optimal values of the value function on the grid points, the VFI algorithm approximates the value function over the entire state space by

[2] According to the homothetic property of the problem, state space, value functions and optimal policy could all be normalized to the case with unit wealth level (Davis and Norman 1990; Goodman and Ostrov 2010).

Algorithm 1 Löwner-John Ellipsoid Construction with VFI

Input Parameters: γ, $\boldsymbol{\beta}$, T, $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$, I, Σ_e, $k = m - 1$
while the change of the constructed ellipsoid is larger than a threshold **do**
  Construct I grid points of current positions $(x_k, \mathbf{y}_k)$ by the Sobol low discrepancy numbers;
  Take numerical expectation by sampling return scenarios using the Sobol low discrepancy numbers;
  Compute the post-trade position $\mathbf{y}_k^+$ for each of the I grid points by numerically solving (10);
  Approximate $V_k(x_k, \mathbf{y}_k)$ by a complete set of polynomials of state variables to get $\hat{V}_k(x_k, \mathbf{y}_k)$;
  Derive the Löwner-John ellipsoid $\mathcal{E}_k$ by solving (12);
  $k \leftarrow k - 1$;
Output:
The ellipsoids $\mathcal{E}_k$, $k = j, \ldots, m - 1$, the break step j.

polynomial basis functions that are generated through taking the tensor-product of monomials from each dimension, i.e., $\hat{V}_k(x_k, \mathbf{y}_k) \approx V_k(x_k, \mathbf{y}_k)$.

However, conventional VFI severely suffers from the curse of dimensionality because the grid sampling in step one, the numerical expectation in step two, and the number of basis functions in step three, all confront with an exponential growth of computational costs with respect to the number of assets. To circumvent these issues, we sample grid points and return scenarios in the first two steps through the Sobol low discrepancy numbers (Niederreiter 1992), and employ the complete set of polynomials in the third step (Judd, Maliar, and Maliar 2011). After finishing VFI based on those novel elements, to evaluate the strategy characterized by $\hat{V}_k(x_k, \mathbf{y}_k)$ for $k = 0, \ldots, m-1$, we still need to expensively optimize over $\hat{V}_k$ at each time step for different return paths. Generally, a sub-optimal solution at an early time step will affect the whole trading trajectory of the later steps. Therefore, rigorous error control in each step of the algorithm should be set up to retard the error growth, which is effectively expensive for computing. To speed up the computation, alleviate error amplification in backward iteration and obviate sacrificing much utility, we propose to explore the *Löwner-John ellipsoid approximation* technique to quantify the optimal trading policy for the transaction costs problem.

In particular, for the transaction costs-aware portfolio problem, a no-trading region Ω_k exists in the state space at each time step representing the trading policy: when $\mathbf{y}_k$ falls into Ω_k, investors do not trade; otherwise investors rebalance the position $\mathbf{y}_k$ back to the boundary of Ω_k. Previous study shows that the no-trading region Ω_k in high dimensions is close to a polyhedron (Goodman and Ostrov 2010). In addition, the regions in different early investment periods are similar and moderately expand as the investment approaches the end (Lynch and Tan 2010; Dai and Zhong 2010). Besides, the transaction costs-aware portfolio problem will degenerate to Merton's portfolio problem when the transaction costs factor $\boldsymbol{\beta} = \mathbf{0}$. Early research has shown Merton's optimal solution $\mathbf{y}_k^*$ lies around the center of the no-trading region Ω_k (Shreve and Soner 1994; Goodman and Ostrov 2010). Hence, we consider approximating the no-trading region Ω_k by a Löwner-John ellipsoid $\mathcal{E}_k$ centered at Merton's optimal solution $\mathbf{y}_k^*$ at time t_k (John 1948). This approximation is an intuitively appealing means to lump its detailed geometry into a single quadratic surface. Denote by $\mathcal{E}_k$ the Löwner-John ellipsoid as the minimum volume ellipsoid that covers a bounded set with nonempty interior:

$$\mathcal{E}_k = \left\{\mathbf{z}_k : \|\mathbf{A}_k(\mathbf{z}_k - \mathbf{y}_k^*)\|_2 \leq 1\right\}, \quad \mathbf{A}_k \succ 0. \tag{11}$$

To employ the Löwner-John ellipsoid to approximate the no-trading region and derive the optimal policy, we develop the following two main steps.

Ellipsoid Construction: At time t_k, given I optimal post-trade risky asset positions $\mathbf{y}_k^{i,+}$, $i = 1, \ldots, I$, the minimum volume ellipsoid that covers all the post-trade positions of risky assets can be computed by solving the following convex optimization problem (Boyd and Vandenberghe 2004):

$$\begin{aligned} &\min_{\mathbf{A}_k \succ 0} \det \mathbf{A}_k^{-1} \\ &\text{s.t.} \quad \|\mathbf{A}_k(\mathbf{y}_k^{i,+} - \mathbf{y}_k^*)\|_2 \leq 1, \quad i = 1, \ldots, I, \end{aligned} \tag{12}$$

where the quantity $\det \mathbf{A}_k^{-1}$ is proportional to the volume of the ellipsoid. As summarized in Algorithm 1, backward from the last rebalance period t_{m-1}, each of the I optimal post-trade positions $\mathbf{y}_k^{i,+}$ is solved from (10) by applying the VFI framework with the Sobol low discrepancy numbers in the first two steps and the complete set of polynomials in the third step. In each step of VFI, given $\mathbf{y}_k^{i,+}$s and $\mathbf{y}_k^*$, convex optimization problem (12) approximates the no-trading region Ω_k by a Löwner-John ellipsoid $\mathcal{E}_k$. Since the constructed ellipsoids become identical after a few backward steps, an early break of the process is expected and computational costs are saved. Specifically, we will identify a break step j, before which all the ellipsoids are assumed the same, i.e., $\mathcal{E}_k = \mathcal{E}_j$ for $k = 0, \ldots, j-1$, after which all the ellipsoids are constructed by solving (12), i.e., $\mathcal{E}_k \approx \Omega_k$ for $k = j, \ldots, m-1$.

Algorithm 2 Lower Bound Evaluation using Monte Carlo Simulation with Ellipsoid Projection

Input: γ, $\boldsymbol{\beta}$, T, $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$, Σ_e, m, $\{\mathcal{E}_0, \ldots, \mathcal{E}_{m-1}\}$
Initialization: Simulate S returns paths;
for $s = 1 \rightarrow S$ **do**
  for $k = 0 \rightarrow m-1$ **do**
    if $\mathbf{y}_k$ falls outside $\mathcal{E}_k$ **then**
      Project $\mathbf{y}_k$ onto $\mathcal{E}_k$ to derive $\mathbf{y}_k^+$ and $(\mathbf{L}_k, \mathbf{U}_k)$ by solving equation (13)
Compute $\mathbb{E}[\mathcal{U}(W_m)] = \frac{1}{S}\sum_{s=1}^{S} \mathcal{U}(W_m^s)$
Output:
Performance of the proposed method measured by $\mathbb{E}[\mathcal{U}(W_m)]$.

Ellipsoid Projection: To implement a strategy we need to decide the trading policy in each rebalance time. When the risky position $\mathbf{y}_k$ falls outside the no-trading region $\mathcal{E}_k$ in the k-th period, it needs rebalancing back to the boundary of $\mathcal{E}_k$. We take Euclidean projection to map $\mathbf{y}_k$ onto the ellipsoid $\mathcal{E}_k$ as an approximation of the post-trade position $\mathbf{y}_k^+$. Being

Table 1: Annualized certainty equivalent rates of return for lower and upper bounds

Data		Lower Bounds				Upper Bounds		Best Performance		
NO.		EA	RBH	MO	CB	MT	MG	LB	UB	GAP
1	CER(%)	8.65	8.60	8.62	8.50	8.90	8.80	8.65	8.80	0.15
	CPU(s)	259	80745	59582	1	0	1527	EA	MG	
2	CER(%)	13.06	12.95	12.98	12.84	13.47	13.36	13.06	13.36	0.30
	CPU(s)	406	68982	42107	1	0	1498	EA	MG	
3	CER(%)	9.73	9.23	5.91	9.35	11.91	9.79	9.73	9.79	0.06
	CPU(s)	24	1031	781	1	0	1530	EA	MG	
4	CER(%)	11.36	10.51	5.93	10.96	11.91	11.42	11.36	11.42	0.06
	CPU(s)	15	7845	2216	3	0	48678	EA	MG	

slightly larger than the associated no-trading region, the constructed ellipsoid gives a conservative but suitable approximation. Numerically, the optimal post-trade position $\mathbf{y}_k^+$ can be computed by solving the following convex optimization problem:

$$\min_{\mathbf{y}_k^+ \in \mathcal{E}_k} \left\| \mathbf{y}_k^+ - \mathbf{y}_k \right\|_2^2. \tag{13}$$

This problem can be rewritten as an Euler Lagrange equation with respect to the Lagrange multiplier, which can be further simplified to be a polynomial equation (Boyd and Vandenberghe 2004). Therefore, an effectively closed-form for $\mathbf{y}_k^+$ is obtained and the computation is much faster than numerically solving the optimization problem (10) or (13). After obtaining the post-trade position $\mathbf{y}_k^+$, the trading policy $(\mathbf{U}_k, \mathbf{L}_k)$ can be trivially backed out from the transition equation (7).

Lower and Upper Bounds

According to the proposed algorithm, we will implement Monte Carlo simulation to evaluate its performance (Brown and Smith 2011). Denote the path of stochastic returns by $\vec{\mathbf{R}}_k = (\mathbf{R}_0, ..., \mathbf{R}_{k-1})$ and the policy vectors by $\vec{\mathbf{L}}_k = (\mathbf{L}_0, ..., \mathbf{L}_{k-1})$ and $\vec{\mathbf{U}}_k = (\mathbf{U}_0, ..., \mathbf{U}_{k-1})$. The expected utility of the final wealth is estimated by the average of the S simulation trails:

$$\mathbb{E}[\mathcal{U}(W_m)] = \frac{1}{S} \sum_{s=1}^{S} \mathcal{U}\big(W_T^s(\vec{\mathbf{R}}_m, \vec{\mathbf{L}}_m, \vec{\mathbf{U}}_m)\big), \tag{14}$$

where $W_m^s(\vec{\mathbf{R}}_m, \vec{\mathbf{L}}_m, \vec{\mathbf{U}}_m)$ represents the generated wealth for the s-th simulation trial by following the ellipsoid projection method. Implementing Monte Carlo simulation is vital as it offers an unbiased estimate of a strategy (Boyle, Broadie, and Glasserman 1997). We first simulate a return path $\vec{\mathbf{R}}_m$. Then, in each period, we compute optimal trading policy $(\mathbf{L}_k, \mathbf{U}_k)$ and post-trade positions $(x_k^+, \mathbf{y}_k^+)$ by the ellipsoid projection. Finally, we take the mean of the utility of the final wealth over the S simulated paths as the lower bound estimate (14). Algorithm 2 summarizes the detailed steps of applying Monte Carlo simulation to evaluate the proposed portfolio strategy. Intuitively, our portfolio strategy could only give lower bounds to the original problem. To objectively assess its performance, we also provide upper bounds to evaluate our lower bounds.

Briefly, the upper bound method developed by (Brown, Smith, and Sun 2010) is based on two elements: (1) relax the nonanticipativity constraints that require the trading decisions to depend on the information available when the decision is made; and (2) impose penalty that punishes violations of nonanticipativity constraints. Denote by $\mathcal{C}$ the feasible set in which each entity of $(\vec{\mathbf{L}}_m, \vec{\mathbf{U}}_m)$ is feasible, and denote by $\pi(\vec{\mathbf{R}}_m, \vec{\mathbf{L}}_m, \vec{\mathbf{U}}_m)$ the penalty function that depends upon the sequence of trades and returns in a given simulation trial. A penalty function π is called dual feasible if $\mathbb{E}[\pi(\vec{\mathbf{R}}_m, \vec{\mathbf{L}}_m, \vec{\mathbf{U}}_m)] \le 0$ for any feasible trading strategy. Intuitively, the dual feasible penalty function does not penalize nonanticipative trades but penalizes those trades that do not satisfy the nonanticipativity constraints. The following result in (Brown, Smith, and Sun 2010) gives the upper bound. For any feasible trading strategy $(\vec{\mathbf{L}}_m, \vec{\mathbf{U}}_m)$ and any dual feasible penalty π,

$$\begin{aligned}\mathbb{E}\left[\mathcal{U}(W_m)\right] \le \mathbb{E}[\max_{(\vec{\mathbf{L}}_m, \vec{\mathbf{U}}_m) \in \mathcal{C}} \{\mathcal{U}(W_m(\vec{\mathbf{R}}_m, \vec{\mathbf{L}}_m, \vec{\mathbf{U}}_m) \\ - \pi(\vec{\mathbf{R}}_m, \vec{\mathbf{L}}_m, \vec{\mathbf{U}}_m)\}].\end{aligned} \tag{15}$$

To compute the right hand side of (15), we once again apply Monte Carlo simulation to compute the expectation. For simulated return paths, the mean over all maximum solutions on the right hand side of (15) offers an upper bound. In particular, we will specify a gradient-based penalty function from a modified problem and take path-wise optimization for each simulated return path (Brown, Smith, and Sun 2010). Thus far the gaps between lower bounds calculated by equation (14) and upper bounds by equation (15) can quantify the quality of the proposed portfolio policy.

Empirical Studies

In this section, to understand the performance of the proposed strategy we compare our results with other lower bound strategies as well as upper bounds across both synthetic and real-world datasets.

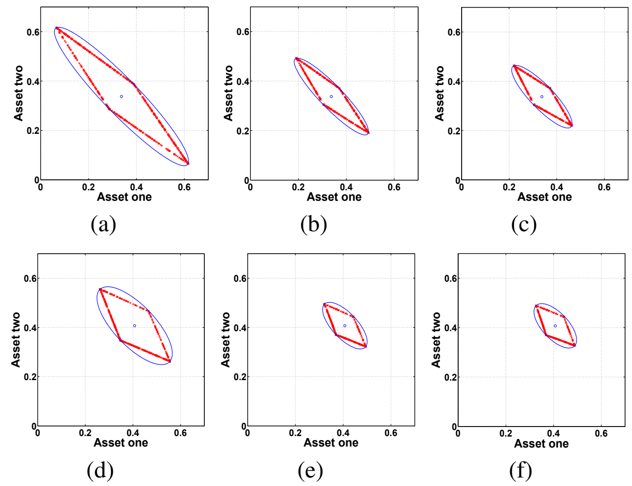

Figure 1: Ellipsoid approximation to no-trading regions in state spaces: the top row has correlation $\rho = 0.7$ and the bottom row $\rho = 0.4$. From left to right, it represents the trading periods from t_{10} to t_8. Common parameters: $n = 2$, $m = 10$, $T = 10$, $\gamma = -2$, $\beta_1 = \beta_2 = 2\%$, $\mu_1 = \mu_2 = 15\%$, $\sigma_1 = \sigma_2 = 35\%$, $r_f = 1\%$. The red curves represent the no-trading regions computed by VFI and the blue curves represent the approximate ellipsoids.

Settings and Baselines

Here we briefly describe the settings and evaluation metrics for our experiments and comparison.

Data and simulation settings: In our experiments, we test across four parameter settings based on synthetic and real-world datasets. Two parameter settings as synthetic data are taken from the relevant work (Muthuraman and Kumar 2006). The other two parameter settings based on market data are chosen from (Brown and Smith 2011).

Baselines: We first compare the proposed ellipsoid approximation strategy (EA) with three representative lower bound strategies (Brown and Smith 2011): (a) the rolling buy-and-hold strategy (RBH), (b) the myopic strategy (MO), and (c) the cost blind strategy (CB). Briefly, applying the RBH strategy, investors incorporate transaction costs into the transition equation of the current period and assume the portfolio will follow its dynamics freely without further rebalancing in future periods. It then maximizes the yield expected utility of the final wealth to attain the trades in the current period. For the MO strategy, investors use power utility function as value functions for all the periods. To follow CB, investors always trade to Merton's solution and then subtract the incurred transaction costs.

Then, we compare EA with two upper bounds: (i) Merton's model (MT) and (ii) the modified gradient-based penalty method (MG). As Merton's model does not incorporate transaction costs, it naturally offers an upper bound. MG is a gradient-based penalty method derived from an ad hoc modified model by (Brown and Smith 2011), which usually provides tighter upper bounds than MT in various settings. Through computing gaps between lower and upper bounds, we can better conclude how close lower bounds are to optimal solutions.

Evaluation metrics: For fair comparison, our results are reported in risk-adjusted return: *annualized certainty equivalent rates of return* (CER) (Brandt 2010) . We run a sufficient number of simulation trials to ensure the corresponding 95% confidence intervals are smaller than 1% of the mean. Given the initial investment W_0 and the investment period of T years, the annualized certainty equivalent rates of return is computed by

$$\text{CER} = \sqrt[T]{\frac{\mathcal{U}^{-1}(\mathbb{E}[\mathcal{U}(W_m)])}{W_0}} - 1. \tag{16}$$

Intuitively, investors with a higher CER can earn more return by taking the same level of risk. We also report the CPU time in second for the different strategies.

Results

Table 1 shows that among the tested lower bounds the proposed EA method constantly achieves the highest CER across all the datasets. In addition, compared with other transaction costs-aware lower bounds methods, EA improves the computational efficiency by 40-500 times. As CB method hinges on the closed-form Merton's solution, its computational time is negligible. But without considering transaction costs, CB always gets much lower risk-adjusted returns than EA. Further, we compare the lower bounds by EA with the upper bounds by MT and MG. As MT has an effectively closed-form solution, we can get its results instantaneously. On the other hand, MG always provides tighter upper bounds of CER than MT. Following the evaluation in (Brown and Smith 2011), we compute gaps of CER between the best lower and upper bounds in Table 1. The table illustrates that the lower bounds by EA are always higher than other strategies and near-optimal, i.e., that the gaps for the annualized returns range from 0.06% to 0.30%. Finally, Figure 1 demonstrates the constructed ellipsoids in an example with two risky assets. It shows that the ellipsoid approximation closely and robustly captures features of the true no-trading region computed by VFI. It also echoes the notable conclusion that the no-trading regions are almost identical in early periods.

Conclusions and Discussions

In this paper, we have studied a broadly existing investment problem for investors: portfolio optimization in the presence of transaction costs under a discrete time and finite horizon setting. Our approach leverages the Löwner-John ellipsoid approximation technique into the conventional VFI framework to characterize the associated optimal trading policy. The proposed solution has not only significantly reduced computational costs up to two orders of magnitude but also obtained near-optimal risk-adjusted returns across both synthetic and real market datasets. Besides those promising results for this particular problem, the novel idea of combining VFI and policy parameterization to attack dynamic programming problems can be explored and utilized in other applications. Our future work includes appropriately incorporating other market friction models into Merton's portfolio problem, such as capital gain taxes and market impacts.

References

Agarwal, A.; Hazan, E.; Kale, S.; and Schapire, R. E. 2006. Algorithms for portfolio management based on the Newton method. In *ICML*, 9–16.

Bertsekas, D. P. 2011. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific.

Bhat, N.; Moallemi, C. C.; and Farias, V. F. 2012. Nonparametric approximate dynamic programming via the kernel method. In *NIPS*, 395–403.

Borodin, A.; El-Yaniv, R.; and Gogan, V. 2004. Can we learn to beat the best stock? *Journal of Artificial Intelligence* 21:579–594.

Boyd, S., and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge University Press.

Boyle, P.; Broadie, M.; and Glasserman, P. 1997. Monte Carlo methods for security pricing. *Journal of Economic Dynamics and Control* 21:31267–1322.

Brandt, M.; Goyal, A.; Santa-Clara, P.; and Stroud, J. 2005. A simulation approach to dynamic portfolio choice with an application to learning about return predictability. *Review of Financial Studies* 18:831–873.

Brandt, M. 2010. Portfolio choice problems. In Ait-Sahalia, Y., and Hansen, L. P., eds., *Handbooks of Financial Econometrics*. Elsevier. 269–336.

Brown, D. B., and Smith, J. 2011. Dynamic portfolio optimiztion with transaction costs: Heuristics and dual bounds. *Management Science* 57:1110–1377.

Brown, D. B.; Smith, J.; and Sun, P. 2010. Information relaxations and duality in stochastic dynamic programs. *Operations Research* 58:785–801.

Cai, Y., and Judd, K. L. 2014. Advances in numerical dynamic programming and new applications. In *Handbook of Computational Economics*. Elsevier.

Cvitanic, J. 2001. Theory of portfolio optimization in markets with frictions. In Jouini, E.; Cvitanic, J.; and Musiela, M., eds., *Handbooks in Mathematical Finance*. Cambridge University Press.

Dai, M., and Zhong, Y. 2010. Penalty methods for continuous-time portfolio selection with proportional transaction costs. *The Journal of Computational Finance* 13(3):1–31.

Davis, M. H. A., and Norman, A. R. 1990. Portfolio selection with transaction costs. *Mathematics of Operations Research* 15:676–713.

Dirk, O., and Peter, W. G. 2001. Kernel-based reinforcement learning in average-cost problems: An application to optimal portfolio choice. In *NIPS*. 1068–1074.

Garlappi, L., and Skoulakis, G. 2010. Solving consumption and portfolio choice problems: The state variable decomposition method. *The Review of Financial Study* 23:3346–3400.

Goodman, J., and Ostrov, D. N. 2010. Balancing small transaction costs with loss of optimal allocation in dynamic stock trading strategies. *SIAM Journal on Applied Mathematics* 70(6):1977–1998.

John, F. 1948. Extremum problems with inequalities as subsidiary conditions. In Friedrichs, K. O.; Neugebauer, O. E.; and Stoker, J. J., eds., *Studies and Essays: Courant Anniversary Volume*. New York: Wiley-Interscience. 187–204.

Judd, K. L.; Maliar, L.; and Maliar, S. 2011. Numerically stable and accurate stochastic simulation approaches for solving dynamic economic models. *Quantitative Economics* 2(2):173–210.

Kalai, A., and Vempala, S. 2003. Efficient algorithms for universal portfolios. *The Journal of Machine Learning Research* 3:423–440.

Li, B., and Hoi, S. C. 2012. On-line portfolio selection with moving average reversion. In *Proceedings of the 29th international conference on machine learning*.

Longstaff, F. A., and Schwartz, E. S. 2001. Valuing American options by simulation: A simple least-squares approach. *The Review of Financial Studies* 14:113–147.

Lynch, A., and Tan, S. 2010. Multiple risky assets, transaction costs and return predictability: Allocation rules and implications for U.S. investors. *Journal of Financial and Quantitative Analysis* 45:1015–1053.

Magill, M., and Constantinides, G. M. 1976. Portfolio selection with transaction costs. *Journal of Economic Theory* 13(2):245–263.

Merton, R. C. 1969. Lifetime portfolio selection under uncertainty: The continuous time model. *Review of Economics and Statistics* 51:247–257.

Merton, R. C. 1971. Optimum consumption and portfolio rules in a continuous time model. *The Journal of Economic Theory* 3:373–413.

Moody, J. E., and Saffell, M. 1999. Reinforcement learning for trading. In *NIPS*. 917–923.

Muthuraman, K., and Kumar, S. 2006. Multidimensional portfolio optimization with proportional transaction costs. *Mathematical Finance* 16:301–335.

Neuneier, R. 1996. Optimal asset allocation using adaptive dynamic programming. In *NIPS*. 952–958.

Niederreiter, H. 1992. *Random number generation and quasi-Monte Carlo methods*, volume 63. SIAM.

Powell, W. B. 2007. *Approximate Dynamic Programming*. John Wiley & Sons.

Rapach, D. E., and Zhou, G. 2012. Forecasting stock returns. In Elliott, G., and Timmermann, A., eds., *Handbook of Forecasting II*. North-Holland.

Rust, J. 2008. Dynamic programming. In *The New Palgrave Dictionary of Economics*. London, UK, Palgrave Macmillan, Ltd.

Shen, W.; Wang, J.; and Ma, S. 2014. Doubly regularized portfolio with risk minimization. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*.

Shreve, S. E., and Soner, H. M. 1994. Optimal investment and consumption with transactions costs. *The Annals of Applied Probability* 4:609–692.