Original Article
International Journal of Fuzzy Logic and Intelligent Systems
Vol. 14, No. 2, June 2014, pp. 73-83
http://dx.doi.org/10.5391/IJFIS.2014.14.2.73

ISSN(Print) 1598-2645
ISSN(Online) 2093-744X

Modern Probabilistic Machine Learning and Control Methods for Portfolio Optimization

Jooyoung Park[1], Jungdong Lim[1], Wonbu Lee[1], Seunghyun Ji[1], Keehoon Sung[1], and Kyungwook Park[2]
[1]Department of Control & Instrumentation Engineering, Korea University, Sejong City, Korea
[2]School of Business Administration, Korea University, Sejong City, Korea

Abstract

Many recent theoretical developments in the field of machine learning and control have rapidly expanded its relevance to a wide variety of applications. In particular, a variety of portfolio optimization problems have recently been considered as a promising application domain for machine learning and control methods. In highly uncertain and stochastic environments, portfolio optimization can be formulated as optimal decision-making problems, and for these types of problems, approaches based on probabilistic machine learning and control methods are particularly pertinent. In this paper, we consider probabilistic machine learning and control based solutions to a couple of portfolio optimization problems. Simulation results show that these solutions work well when applied to real financial market data.

Keywords: Machine learning, Portfolio optimization, Evolution strategy, Value function.

1. Introduction

Recent theoretical progress in the field of machine learning and control has many implications for related academic and professional fields. The field of financial engineering is one particular area that has benefited greatly from these advancements. Portfolio optimization problems [1–8] and the pricing/hedging of derivatives [9] can be performed more effectively using recently developed machine learning and control methods. In particular, since portfolio optimization problems are essentially optimal decision-making problems that rely on actual data observed in a stochastic environment, theoretical and practical solutions can be formulated in light of recent advancements. These problems include the traditional mean-variance efficient portfolio problem [10], index tracking portfolio formulation [6–8, 11], risk-adjusted expected return maximizing strategy [1, 2, 12], trend following strategy [13–17], long-short trading strategy (including the pairs trading strategy) [13, 18–20], and behavioral portfolio management.

Modern machine learning and control methods can effectively handle almost all of the portfolio optimization problems just listed. In this paper, we consider a solution to the trend following trading problem based on the natural evolution strategy (NES) [21–23, 25] and a risk-adjusted expected profit maximization problem based on an approximate value function (AVF) method [27–30].

This paper is organized as follows: In Section 2, we briefly discuss relevant probabilistic machine learning and control methods. The exponential NES and iterated approximate value

Received: May 21, 2014
Revised : Jun. 24, 2014
Accepted: Jun. 24, 2014

Correspondence to: Jooyoung Park
(parkj@korea.ac.kr)

function method, which are the two main tools employed in this paper, are also summarized. Solutions to the trend following trading problem and the risk-adjusted expected profit maximization problem as well as simulation results using real financial market data are presented in Section 3. Finally, in Section 4, we present our concluding remarks.

2. Modern Probabilistic Machine Learning and Control Methods

In this section, we describe relevant advanced versions of the NES and AVF methods that will be applied later in this paper.

The NES method belongs to a family of evolution strategy (ES) type optimization methods. Evolution strategy, in general, attempts to optimize utility functions that cannot be modeled directly, but can be efficiently sampled by users. A probability distribution (typically, a multi-variate Gaussian distribution) is utilized by NES to generate a group of candidate solutions. In the process of updating distribution parameters based on the utility values of candidates, NES employs the so-called natural gradient [23, 31] to obtain a sample-based search direction. In other words, the main idea of NES is to follow a sampled natural gradient of expected utility to update the search distribution. The samples of NES are generated according to the search distribution $\pi(\cdot|\theta)$, and by utilizing these samples, NES tries to locate a parameter update direction that will increase the performance index, $J(\theta)$. This performance index is defined as the expected value of the utility, $f(z)$, under the search distribution:

$$J(\theta) = E_\theta[f(z)] = \int f(z)\pi(z|\theta)dz. \tag{1}$$

Note that by the log-likelihood strategy, the gradient of the expected utility with respect to the search distribution parameter θ can be expressed as

$$\nabla_\theta J(\theta) = E[f(x)\nabla_\theta \log(\pi(x|\theta))]. \tag{2}$$

Hence, when we have independent and identically distributed samples, $z_i, i \in \{1, \cdots, n\}$, a sample-based approximation to the regular policy gradient (often referred to as the vanilla gradient) of the expected utility can be expressed as

$$\nabla_\theta J(\theta) \approx (\sum_{i=1}^{n} f(z_i)\nabla_\theta \log \pi(z_i|\theta))/n. \tag{3}$$

It is widely accepted that using the natural gradient is more advantageous than the vanilla gradient when it is necessary to search optimal distribution parameters while staying close to the present search distribution [23, 31]. In the natural gradient based search method, the search direction is obtained by replacing the gradient $\nabla_\theta J(\theta)$ with the natural gradient defined by $F^{-1}(\theta)\nabla_\theta J(\theta)$, where $F(\theta) = E[\nabla_\theta \log \pi(x|\theta)\nabla_\theta \log \pi(x|\theta)^T]$ is the Fisher information matrix. Note that the Fisher information matrix can be estimated from samples. Therefore, the core procedure of the NES algorithm can be summarized as follows [22]:

Preliminary steps:

1. Choose the learning rate, η, number of samples in each generation, n, and utility function, f.
2. Initialize parameter θ of the search distribution $\pi(\cdot|\theta)$.

Main steps: Repeat the following procedure until the stopping condition is met.

1. For $i = 1, \cdots, n$
   - Draw a sample z_i from the current search distribution.
   - Compute the utility of the sample, $f(z_i)$.
   - Compute the gradient of the log-likelihood, $\nabla_\theta \log \pi(z_i|\theta)$.

   end
2. Obtain the Monte-Carlo estimate of the gradient:
$$\hat{\nabla}_\theta J(\theta) = \frac{1}{n}\sum_{i=1}^{n} f(z_i)\nabla_\theta \log \pi(z_i|\theta). \tag{4}$$
3. Obtain the Monte-Carlo estimate of the Fisher information matrix:
$$\hat{F}(\theta) = \frac{1}{n}\sum_{i=1}^{n} f(z_i)\nabla_\theta \log \pi(z_i|\theta)\nabla_\theta \log \pi(z_i|\theta)^T. \tag{5}$$
4. Update parameter θ of the search distribution:
$$\theta \leftarrow \theta + \eta\hat{F}(\theta)^{-1}\hat{\nabla}_\theta J(\theta). \tag{6}$$

The procedure shown above is a basic form of NES and can be modified based on the application. For example, the concept of the baseline can be employed to reduce the estimation variance [21]. Recent improvements to the NES procedure can be found in [21–23, 25], and one of the most remarkable improvements is the exponential NES [23, 25]. The main idea of the exponential NES method is to represent the covariance matrix of the multivariate Gaussian distribution, $\pi(\cdot|\theta)$, using the following matrix exponential map:

$$\exp : M \mapsto \exp(M) = \sum_{k=0}^{\infty} \frac{M^k}{k!}. \tag{7}$$

Two remarkable advantages of using the matrix exponential are that it enables the covariance matrix to be updated in a vector space, and it makes the resultant algorithm invariant under linear transformations. The key idea of the exponential NES is the use of natural coordinates defined according to the following change of variables [23]:

$$\mu_{\text{new}} = \mu + A^T \delta,\ A_{\text{new}} = \exp(M/2)A, \tag{8}$$

where A and M are matrix variables satisfying $C = A^T A = \exp(M)$ for the covariance matrix C of the search distribution. The use of natural coordinates renders the step of inverting the Fisher information matrix unnecessary, hence, bypassing a major computational burden of the original NES. In Section 3, we apply the exponential NES, which is now a state-of-the-art evolution strategy, to the problem of finding a flexible long-flat-short type rule for a trend following trading strategy.

Another tool used in portfolio optimization applications of this paper is a special class of approximate value function methods [27–30]. In general, stochastic optimal control problems can be solved by utilizing state value functions, which estimate performance at a given state. The solutions of stochastic optimal control problems based on the value function are called dynamic programming. For more details on the various applications of dynamic programming, please refer to [32, 33]. Solving stochastic control problems by dynamic programming corresponds to finding the best state-feedback control policy

$$u(t) = \phi_t(x(t)), t = 0, 1, \cdots \tag{9}$$

to optimize the performance index of specified constraints and dynamics

$$x(t+1) = f(x(t), u(t), w(t)), \tag{10}$$

where $x(t)$ is the state, $u(t)$ is the control input, and $w(t)$ is the disturbance. The expected sum of discounted stage costs, $J = E[\sum_{t=0}^{\infty} \gamma^t l(x(t), u(t))]$, is widely used as the performance index. The minimal performance index value, J^*, is obtained by minimizing the performance index over all admissible control policies; the optimal control policy achieving J^* is denoted by ϕ_t^*. The state value function is defined as the minimum total expected cost achieved by an optimal control policy for the given initial state $x(0) = z$. Formally,

$$V^*(z) = \inf_{\phi_t} E(\sum_{t=0}^{\infty} \gamma^t l(x(t), \phi_t(x(t)))|x(0) = z). \tag{11}$$

Note that the optimal performance index value is $J^* = V^*(x_0)$ for the given initial condition x_0. The state value function in (11) is the fixed point of the following Bellman equation:

$$V^*(z) = \inf_v (l(z, v) + \gamma E[V^*(f(z, v, w))]. \tag{12}$$

In operator equation form, this fixed point property can be written as

$$V^* = TV^*, \tag{13}$$

where

$$(Tg)(z) = \inf_v (l(z, v) + \gamma E[g(f(z, v, w))]). \tag{14}$$

Because it is hard to compute optimal control policy ϕ_t^* that satisfies the Bellman equation except special case [32], the AVF, $\hat{V} : X \to U$, is utilized to obtain approximate solutions to the stochastic control problem. In an example considering a set of real financial market data, we utilize an ADP-based solution procedure utilizing the iterated AVF policy approach of O'Donoghue, Wang, and Boyd [29]. In the iterated AVF method [29], by letting the parameters of the approximate value functions satisfy the iterated Bellman inequalities

$$\hat{V}_t \leq T\hat{V}_{t+1},\ t = 0, \ldots, M \tag{15}$$

with $\hat{V}_M = \hat{V}_{M+1}$, it is ensured that $\hat{V}_0$ is a lower bound of the optimal state value function V^* [27, 29]. Also by optimizing this lower bound $\hat{V}_0$ via convex optimization, the iterated AVF approach finds the approximate state value functions, $\hat{V}_0, \hat{V}_1, \cdots, \hat{V}_{M+1}$, and the associated policies $\hat{\phi}_0, \hat{\phi}_1, \cdots$. Note that in this step, the associated iterated AVF policies are obtained by

$$\hat{\phi}_t(x) = \text{argmin}_u (l(x, u) + \gamma E[\hat{V}_{t+1}(f(x, u, w))]) \tag{16}$$

for $t = 0, \ldots, M$, and

$$\hat{\phi}_t(x) = \text{argmin}_u (l(x, u) + \gamma E[\hat{V}_{M+1}(f(x, u, w))]) \tag{17}$$

for $t > M$ [29].

3. Machine Learning and Control Based Portfolio Optimization

In this section, we present probabilistic machine learning and control based solutions to two important portfolio optimization problems: the trend following trading problem and the risk-adjusted expected profit maximization problem. The first topic of our portfolio application concerns trading strategy. There have been a great deal of theoretical studies on trading strategies in financial markets; however, a majority of them are focused on the contra-trend strategy, which includes trading policies for a mean-reverting market [14]. Research interests in trend-following trading rules are also growing. A strong mathematical foundation has led to some important theorems regarding the trend following trading strategy [14–17]. We consider an exponential NES based solution to find an efficient trend following strategy. One of the key references for our solution is the stochastic control approach of Dai et al. [14, 15]. In [14], the authors considered a bull-bear switching market, where the drift of the asset price switches back and forth between two values representing the bull market mode and the bear market mode. These switching patterns follow an unobservable Markov chain. More precisely, the governing equation for the asset price, S_r, in [14] is given by

$$dS_r = S_r[\mu(\alpha_r)dr + \sigma dB_r], S_t = S, t \le r \le T \le \infty, \quad (18)$$

where $\alpha_r \in \{1, 2\}$ is the mode of the market at time r, and its movement follows a two-state Markov chain. The first state, $\alpha_r = 1$, represents the bull market mode, and its second state, $\alpha_r = 2$, represents the bear market mode. Drifts, $\mu(1) = \mu_1$ and $\mu(2) = \mu_2$, represent the expected return rates in the bull and bear markets, respectively. Clearly, these drift values must satisfy $\mu_1 > 0$ and $\mu_2 < 0$. The Markov chain for the movement of α_r is described by the following generator [14]:

$$Q = \begin{bmatrix} -\lambda_1 & \lambda_1 \\ \lambda_2 & -\lambda_2 \end{bmatrix}. \quad (19)$$

Note that in this Markov chain generator, λ_1 and λ_2 are the switching intensities for the bull-to-bear transition and bear-to-bull transition, respectively. In [14], it is assumed that the switching market mode, $\{\alpha_r\}$, and the Brownian motion for the asset price variation, $\{B_r\}$, are independent. Moreover, Dai et al. [14] showed that the optimal trend following long-flat type trading rules can be found by solving the associated Hamilton-Jacobi-Bellman (HJB) equation and employing the conditional probability for the bull market to generate the trade signals. In their problem formulation, the transaction cost and risk-free interest rate were fixed at $100K[\%]$ and ρ, respectively. The resulting optimal buying times, $\tau_1, \tau_2, \cdots$, and selling times, $v_1, v_2, \cdots$, were obtained by optimizing the performance index given by

$$\begin{aligned} J_i(S, \alpha, t, \Lambda) =& E_t \left\{ \sum_{n=1}^{\infty} [e^{-\rho(v_n - t)} S_{v_n}(1-K) - \right. \\ & \left. e^{-\rho(\tau_n - t)} S_{\tau_n}(1+K)] I_{\{\tau_n < T\}} \right\} \end{aligned} \quad (20)$$

if initially flat, and

$$\begin{aligned} J_i(S, \alpha, t, \Lambda) =& E_t \left\{ e^{-\rho(v_1 - t)} S_{v_1}(1-K) \right. \\ & + \sum_{n=1}^{\infty} [e^{-\rho(v_n - t)} S_{v_n}(1-K) - \\ & \left. e^{-\rho(\tau_n - t)} S_{\tau_n}(1+K)] I_{\{\tau_n < T\}} \right\} \end{aligned} \quad (21)$$

if initially long [14]. The results of [14] are mathematically rigorous and establish a strong theoretical justification for the trend following trading theory. We utilize a less mathematical, but hopefully easier to understand approach, which is based on the exponential NES method [23, 25] for the trend following trading problem. Note that in the exponential NES approach, the performance index may be chosen with more flexibility. This paper extends our previous work on this topic [13] in two ways. First, we utilize a more advanced version of the NES–the exponential NES [23, 25]–that is now the state-of-the-art method in the field. Second, we focus on more flexible long-flat-short type trading rules whereas our previous paper considered only the long-flat strategy. According to [14], there exist two monotonically increasing optimal sell and buy boundaries, $p_s^*(t)$ and $p_b^*(t)$, in case of the finite-horizon problem for obtaining the optimal trend following long-flat type trading rule. When long-term investments are emphasized, the behavior of the system is similar to the infinite horizon case, and these threshold functions can be approximated by constants [14, 15]. Following this approximation scheme, we try to find threshold constants for all possible transitions that can occur in long-flat-short type trading, i.e., $p^*_{\text{flat-to-short}}$, $p^*_{\text{long-to-flat}}$, $p^*_{\text{short-to-flat}}$, $p^*_{\text{flat-to-long}}$, by applying the exponential NES method [23, 25]. The price-series sample paths generated in accordance with the switching geometric Brownian motion (18) are required during the training phase. Simulating both

Markov chains and geometric Brownian motions is not difficult; thus, the price-series sample path generation can be performed efficiently. By combining the thresholds found by the exponential NES method together with the Wonham filter [14, 34] to estimate the conditional probability that the mode of the market is bull, one can obtain a long-flat-short type trend following trading strategy. To illustrate the applicability of the exponential NES based trading strategy, we considered the problem of determining a trend following trading rule for the NASDAQ index. For the example, NASDAQ closing data from 1991 to 2008 was considered (see Fig. 2). According to the estimation results of [14], the parameters of the switching geometric Brownian motion for the NASDAQ are as follows:

$$\mu_1 = 0.875, \mu_2 = -1.028,$$
$$\sigma_1 = 0.273, \sigma_2 = 0.35, \sigma = 0.31,$$
$$\lambda_1 = 2.158, \lambda_2 = 2.3,$$

where σ is the simple average of σ_1 and σ_2. Furthermore, the ratio of slippage per transaction and the risk-free interest rate were fixed as

$$K = 0.001, \text{ and } \rho = 5.4\%,$$

respectively. For training data, we generated episodes by utilizing parameter estimation results. By performing the exponential NES based training for these episodes, we obtained the threshold values for the trend following trading rules.

Figures 1-6 show the simulation results for the exponential NES based trend following trading rules. For these simulations, we set the initial wealth to one. Figure 1 shows the learning curve, which graphs the average total cost sums versus the policy update over a set of 10 simulation runs. As shown in the curve, the exponential NES method exhibits desirable behavior within less than 250 policy updates. This indicates that the exponential NES method works well for finding a long-flat-short type trading strategy. Figure 2 shows the NASDAQ index values together with the long-flat-short trading signals resulting from a policy obtained by the exponential NES method over the entire period. For comparison purposes, we also show the long-flat trading signals obtained by the NES approach [13]. According to Fig. 2, the total number of position changes in the long-flat-short type trading strategy (2nd panel) is 345. Note that this value differs from the corresponding number of position changes in the long-flat type trading strategy (3rd panel, 100 changes) obtained by the NES approach [13]. Simulation results in Fig. 3 show that with the exponential NES based long-flat-short type trading strategy, trade returns are generally large and wealth steadily increases until it reaches 21.52 at the final time. In comparison, Fig. 4 shows that with the long-flat type trading rule, the trade returns are generally small, and wealth increases at a relatively slower rate. Its wealth value at the final time is only 10.65. From these simulation results, one can see that when $K = 0.001$, short positions slightly changed the number of trading and significantly improved wealth. To investigate the robustness of NES based trading rules against system changes, we performed simulations for various values of the transaction cost ratio. In particular, we considered the case when the transaction cost increased tenfold (i.e., $K = 0.01$); simulation results are shown in Figs. 5 and 6. Figure 5 shows the trading positions resulting from the long-flat-short type strategy (top) and the long-flat type strategy (bottom) when $K = 0.01$. When the trading cost increases, both the long-flat-short type strategy and the long-flat type strategy trade less frequently compared to the case when $K = 0.001$. Interestingly, the trading frequency of the long-flat-short type strategy decreased at a slower rate than that of the long-flat type trading strategy. We believe that this difference is due to the fact that the long-flat-short type strategy has more flexibility and can cope with system changes with less sensitivity. Finally, Fig. 6 shows the wealth resulting from the long-flat-short type policy (top) and the long-flat type policy (bottom) when $K = 0.01$. Note that the wealth values of the long-flat-short type strategy and the long-flat type strategy at the final time are 10.85 and 5.09, respectively. These values are still considerably larger than the wealth values of the buy-and-hold strategy (3.34) and the risk-free interest rate (2.74).

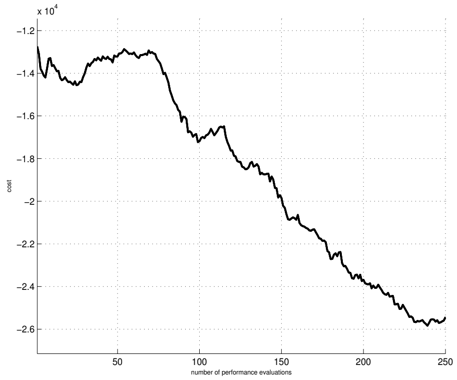

Figure 1. Learning curve.

For the second application example, we considered the risk-adjusted, expected profit maximization problem and utilized an AVF [27–30] based procedure to find an efficient solution. To

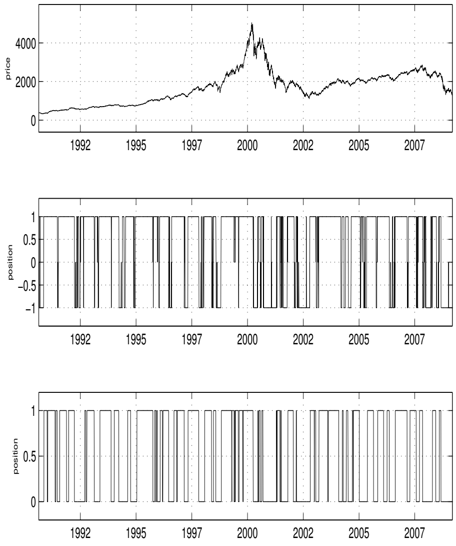

Figure 2. NASDAQ index and trading position when $K = 0.001$.

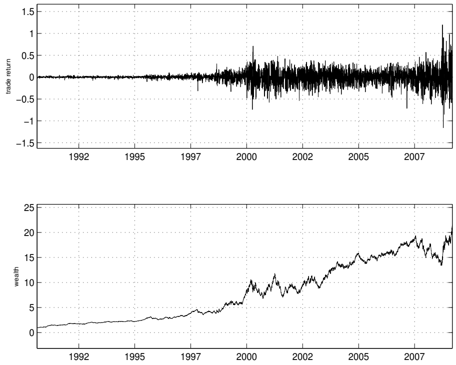

Figure 3. Trade return and wealth resulting from the long-flat-short type policy when $K = 0.001$.

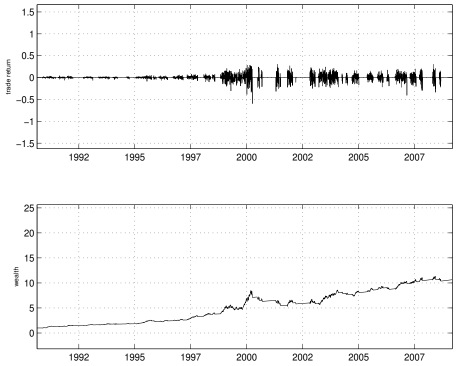

Figure 4. Trade return and wealth resulting from the long-flat type policy when $K = 0.001$.

express the risk-adjusted, expected profit maximization problem

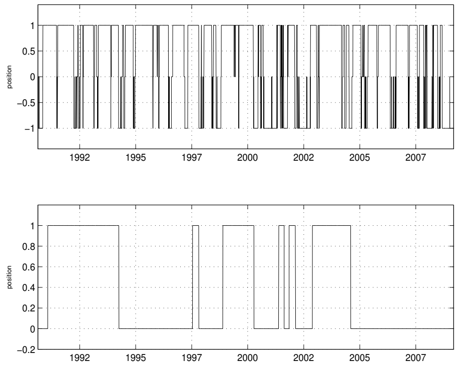

Figure 5. Trading position resulting from the long-flat-short type policy (top) and the long-flat type policy (bottom) when $K = 0.01$.

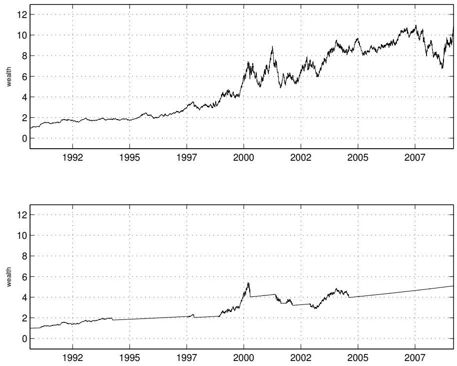

Figure 6. Wealth resulting from the long-flat-short type policy (top) and the long-flat type policy (bottom) when $K = 0.01$.

in state-space format, it is necessary to define the state and control input together with the performance index that is used as an optimization criterion. To do this, we follow the research of Boyd et al. [1, 28, 30]. We define the state vector as the collection of the portfolio positions. Let $x_i(t)$ denote the dollar value of asset i at the beginning of time t. Then the state vector is given by

$$x(t) = [x_1(t), \cdots, x_n(t)]^T. \qquad (22)$$

The control input considered for this problem is a vector of trades,

$$u(t) = [u_1(t), \cdots, u_n(t)]^T, \qquad (23)$$

executed for portfolio $x(t)$ at the beginning of each time step t. Note that $u_i(t)$ represents buying or selling assets; the asset associated with $x_i(t)$ is bought when $u_i(t) > 0$ and sold when $u_i(t) < 0$. Having these state and input definitions, the state transition is given by

$$x(t+1) = \mathrm{diag}(r(t))(x(t) + u(t)), \qquad (24)$$

where $r(t)$ is the vector of asset returns in period t. The return vector, $r(t)$, is independent and identically distributed with mean vector $E[r(t)] = \bar{r}$ and covariance matrix $E[(r(t) - \bar{r})(r(t) - \bar{r})^T] = \Sigma$. Note that the mean vector and covariance matrix do not change over time. For the performance index, we considered

$$\ell(x, u) = 1^T u + \lambda(x + u)^T \Sigma(x + u) + u^T \mathrm{diag}(s)u + \kappa|u|,$$

where the total gross cash entered in the portfolio is $1^T u$, $\lambda(x + u)^T \Sigma(x + u)$ is the quadratic post-trade risk penalty, $u^T \mathrm{diag}(s)u$ is the quadratic transaction cost, and $\kappa|u|$ is the linear transaction cost. Furthermore, $\lambda \geq 0$ is the risk aversion parameter, $s_i \geq 0$ is the price-impact cost for the ith asset, and κ is the ratio of slippage per transaction. We considered the case when the initial portfolio $x(0)$ was fixed at x_0. In general, when portfolio optimization problems are solved, the control input $u(t)$ should satisfy certain, naturally arising constraints. In particular, we considered the control input bound constraint:

$$\text{Constraint} : \|u(t)\|_\infty < U_{\text{bdd}},\ t = 0, 1, \cdots, \tag{25}$$

which means that only a limited amount of trading is allowed for each asset. Thus, the risk-adjusted profit maximization problem can be expressed as

$$\begin{array}{l} \min_{u(\cdot)}\ E[\sum_{t=0}^{\infty} \gamma^t \ell(x(t), u(t)] \\ \text{subject to} \\ x(t+1) = \mathrm{diag}(r(t))(x(t) + u(t)), \\ E[r(t)] = \bar{r}, \\ E[(r(t) - \bar{r})(r(t) - \bar{r})^T] = \Sigma, \\ \|u(t)\|_\infty < U_{\text{bdd}} \text{ for all } t, \\ \text{and } x(0) = x_0. \end{array} \tag{26}$$

To solve this optimization problem, we utilized the iterated AVF approach [29], which is one of the most advanced AVF methods. In the iterated AVF approach, convex quadratic functions

$$\hat{V}_t(x) = x^T P_t x + 2{p_t}^T x + q_t \tag{27}$$

are used to approximate the state value function at time t, and the $\hat{V}_t$ parameters satisfy a series of Bellman inequalities

$$\hat{V}_t \leq \mathcal{T}\hat{V}_{t+1},\ \hat{V}_{t+1} \leq \mathcal{T}\hat{V}_{t+2},\ \cdots,\ \hat{V}_{t+M-1} \leq \mathcal{T}\hat{V}_{t+M}, \tag{28}$$

with $\hat{V}_{t+M-1} = \hat{V}_{t+M}$. The Bellman inequalities guarantee that $\hat{V}_t$ is a lower bound of the optimal state value function V_t^* [27–29]. The iterated AVF method maximizes this lower bound using convex optimization [29]. Another constraint is $\|u\|_\infty \leq U_{\text{bdd}}$, which can be written in terms of quadratic inequalities as

$$1 - u^T(e_i e_i^T)u \geq 0,\ i = 1, \cdots, n, \tag{29}$$

where e_i is the ith column of the identity matrix I_n. This constraint enables us to obtain sufficient conditions for the constrained Bellman inequality requirements in the form of linear matrix inequalities (LMIs) using the S-procedure [28, 35]. As a real-market example of portfolio optimization, we examined an application of the iterated AVF approach [29] for a set of real financial market data [2, 6]. The data considered five major stocks: IBM, 3M, Altria, Boeing, and AIG (the ticker symbols of these are IBM, MMM, MO, BA, and AIG, respectively). For the training data, we used the weekly prices of the five major stocks from Jan. 2, 1990 to Dec. 27, 2004, and obtained the exponentially weighted moving average (EWMA) of the mean return vector, $\bar{r}$ (with the effective window size of three years), and covariance matrix, Σ. During the test period (2005 to 2007), iterated AVF based trading was performed every four weeks (20 trading days). For the risk-free rate, we assumed $\rho = 0.05$ as in [2], and the discount factor, γ, was defined accordingly (i.e., $\gamma = \exp(-\rho/(52/4))$). For the upper bound of the trading amount, we used $U_{\text{bdd}} = 20$. The coefficients of the risk penalty and transaction costs were

$$\lambda = 0.005, \kappa = 0.005, s_i = 0.005,\ i = 1, \cdots, 5.$$

For the Bellman inequalities in (28), we considered $M = 150$ time steps. Finally, the initial portfolio vector and the initial wealth level were chosen to be $x_0 = [0, 0, 0, 0, 0]^T$ and $W = 100$, respectively.

Figures 7-13 show the simulation results of the portfolio optimization example under the iterated AVF method. Figure 7 depicts the portfolio profile during the test period. From this figure, one can see that with the passage of time, the portfolio profile slowly changes its direction to increase the performance index. Figure 8 shows the gross cash put into the portfolio, and Fig. 9 plots the cumulative cost sums. From Fig. 8, it is clear cash is entered into the portfolio during the early stage of trade; as time progresses, the portfolio gains income. Furthermore, based on the trend of this figure, we can expect more profit to be derived in the later stage of trading. Figure 10 shows the transaction cost. When the portfolio is being built, the transaction cost is very high; however, it stabilizes over time. Figure 11 shows the risk penalty, which is always non-negative and increasing. Wealth history and amount of cash holdings are plotted in Figs. 12 and 13, respectively. Note that in this scenario, wealth steadily increases as trading proceeds and reaches approximately 220 (yielding a 120% profit at the end of 2007). Also note that according to Fig. 13, the amount of cash holding briefly remains at the initial value, rapidly decreases, and then slowly begins to restore itself. This behavior suggests that in the iterated AVF based trading strategy, a large amount of profit is obtained when aggressive initial investments are made with cashing out subsequently.

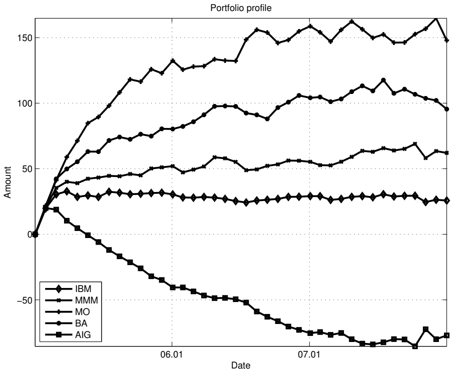

Figure 7. Portfolio profile.

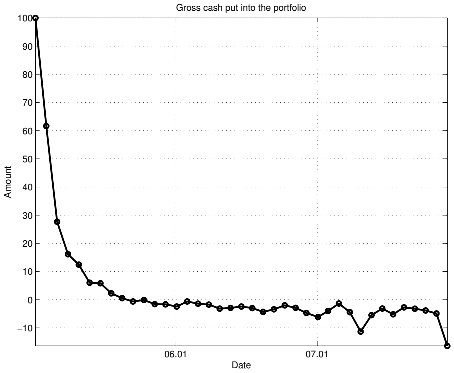

Figure 8. Gross cash put into the portfolio.

4. Conclusion

Machine learning and control methods have been applied to a variety of portfolio optimization problems. In particular, we considered two important classes of portfolio optimization problems: the trend following trading problem and the risk-adjusted profit maximization problem. The exponential NES and iterated approximate value function methods were applied to solve these problems. Simulation results showed that these probabilistic machine learning and control based solutions worked well when applied to real financial market data. In the future, we plan to consider more extensive simulation studies, which will further identify the strengths and weaknesses of probabilistic machine learning and control based methods, and applications of our methods to other types of financial decision making problems.

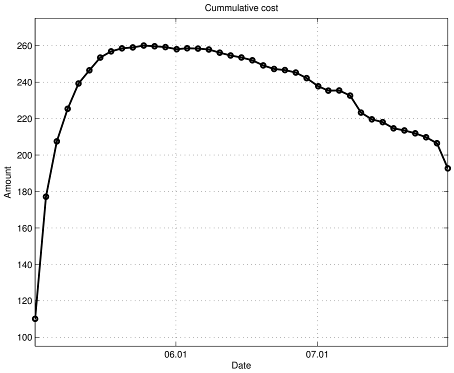

Figure 9. Cumulative cost.

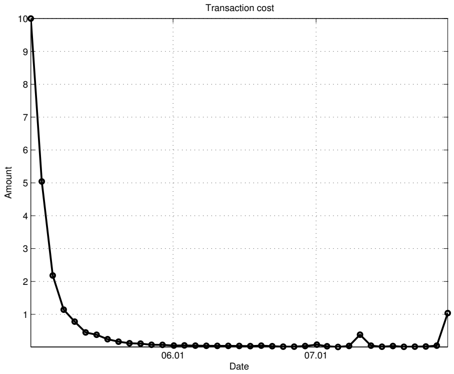

Figure 10. Transaction cost.

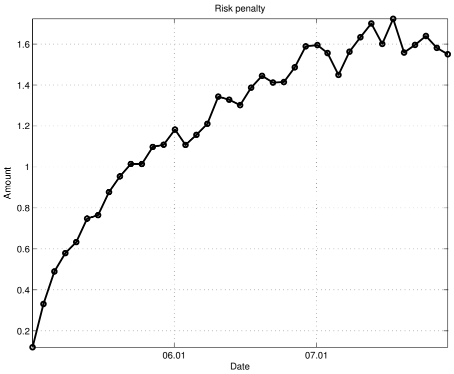

Figure 11. Risk penalty.

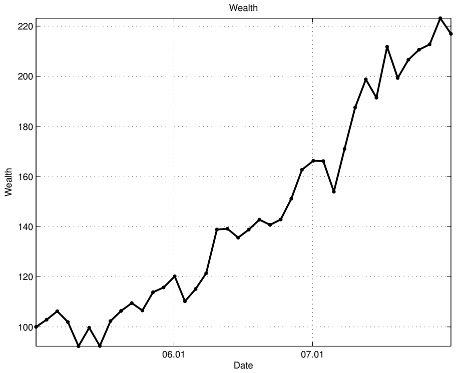

Figure 12. Wealth.

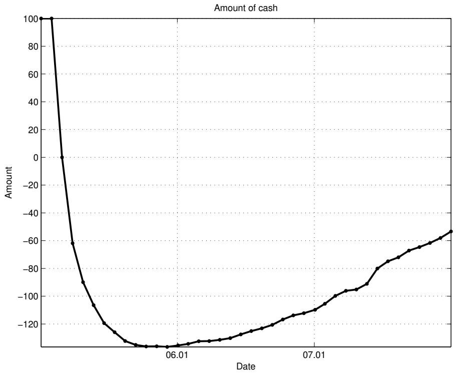

Figure 13. Amount of cash.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0021188).

References

[1] S. Boyd, M. T. Mueller, B. O'Donoghue, and Y. Wang, "Performance bounds and suboptimal policies for multi-period investment," *Foundations and Trends in Optimization*, vol. 1 no. 1, pp. 1-72, 2014. http://dx.doi.org/10.1561/2400000001

[2] J. A. Primbs, "Portfolio optimization applications of stochastic receding horizon control," in *Proceeding of the 2007 American Control Conference*, New York, NY, July 9-13, 2007, pp. 1811-1816. http://dx.doi.org/10.1109/ACC.2007.4282251

[3] G. C. Calafiore, "Multi-period portfolio optimization with linear control policies," *Automatica*, vol. 44, no. 10, pp. 2463-2473, Oct. 2008. http://dx.doi.org/10.1016/j.automatica.2008.02.007

[4] S. Alenmyr and A. ´Ogren, "Model Predictive Control for Stock Portfolio Selection," M.S. Thesis, Department of Automatic Control, Lund University, Sweden, 2010.

[5] B. R. Barmish, "On performance limits of feedback control-based stock trading strategies," in *Proceedings of 2011 American Control Conference*, San Francisco, CA, June 29-July 1, 2011, pp. 3874-3879.

[6] J. A. Primbs, and C. Sung, "A stochastic receding horizon control approach to constrained index tracking," *Asia-Pacific Financial Markets*, vol. 15, no. 1, pp. 3-24, Mar. 2008. http://dx.doi.org/10.1007/s10690-008-9073-1

[7] J. E. Beasley, N. Meade, and T. J. Chang, "An evolutionary heuristic for the index tracking problem," *European Journal of Operational Research*, vol. 148, no. 3, pp. 621-643, Aug. 2003. http://dx.doi.org/10.1016/S0377-2217(02)00425-3

[8] R. Jeurissen and J. van den Berg, "Index tracking using a hybrid genetic algorithm," in *Proceedings of the ICSC Congress on Computational Intelligence Methods and Applications*, , Istanbul, Turkey, 2005. http://dx.doi.org/10.1109/CIMA.2005.1662364

[9] J. A. Primbs, "Dynamic hedging of basket options under proportional transaction costs using receding horizon control," *International Journal of Control*, vol. 82, no. 10, pp. 1841-1855, Oct. 2009. http://dx.doi.org/10.1080/00207170902783341

[10] H. Markowitz, *Portfolio Selection: Efficient Diversification of Investments (Cowles Foundation for Research in Economics at Yale University Monograph 16)*, New York, NY: Wiley, 1959.

[11] J. Park, D. Yang, and K. Park, "Approximate dynamic programming-based dynamic portfolio optimization for constrained index tracking,"*International Journal of Fuzzy Logic and Intelligent Systems*, vol. 13, no. 1, pp. 19-28, Mar. 2013. http://dx.doi.org/10.5391/IJFIS.2013.13.1.19

[12] J. Park, J. Jeong, and K. Park, “An investigation on dynamic portfolio selection problems utilizing stochastic receding horizon approach,” *Journal of Korean Institute of Intelligent Systems*, vol. 22, no. 3, pp. 386-393, Jun. 2012. http://dx.doi.org/10.5391/JKIIS.2012.22.3.386

[13] J. Park, D. Yang, and K. Park, “Investigations on dynamic trading strategy utilizing stochastic optimal control and machine learning,” *Journal of Korean Institute of Intelligent Systems*, vol. 23, no. 4, pp. 348-353, Aug. 2013. http://dx.doi.org/10.5391/JKIIS.2013.23.4.348

[14] M. Dai, Q. Zhang, and Q. J. Zhu, “Trend following trading under a regime switching model,” *SIAM Journal on Financial Mathematics*, vol. 1, no. 1, pp. 780-810, 2010. http://dx.doi.org/10.1137/090770552

[15] M. Dai, Q. Zhang, and Q. J. Zhu, “Optimal trend following trading rules,” *Social Science Research Network Paper*, July 19, 2011. http://dx.doi.org/10.2139/ssrn.1762118

[16] H. T. Kong, Q. Zhang, and G. G. Yin, “A trend-following strategy: conditions for optimality,” *Automatica*, vol. 47, no. 4, pp. 661-667, Apr. 2011. http://dx.doi.org/10.1016/j.automatica.2011.01.039

[17] J. Yu and Q. Zhang, “Optimal trend-following trading rules under a three-state regime switching model,” *Mathematical Control and Related Fields*, vol. 2, no. 1, pp. 81-100, 2012. http://dx.doi.org/10.3934/mcrf.2012.2.81

[18] S. J. Kim, J. Primbs, and S. Boyd, “Dynamic spread trading,” 2008 manuscript submitted.

[19] J. A. Primbs, “A control systems based look at financial engineering,” 2009 manuscript submitted.

[20] S. Mudchanatongsuk, J. A. Primbs, and W. Wong, “Optimal pairs trading: a stochastic control approach,” in *Proceedings of the American Control Conference*, Seattle, WA, June 11-13, 2008, pp. 1035-1039. http://dx.doi.org/10.1109/ACC.2008.4586628

[21] D. Wierstra, T. Schaul, J. Peters, and J. Schmidhuber, “Natural evolution strategies,” in *Proceedings of the IEEE World Congress on Evolutionary Computation*, Hong Kong, June 1-6, 2008, pp. 3381-3387. http://dx.doi.org/10.1109/CEC.2008.4631255

[22] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, and J. Schmidhuber, “Natural evolution strategies,” June 22, 2011. http://arxiv.org/abs/1106.4487

[23] T. Glasmachers, T. Schaul, S. Yi, D. Wierstra, and J. Schmidhuber, “Exponential natural evolution strategies,” in *Proceedings of the 12th Genetic and Evolutionary Computation Conference*, Portland, OR, July 7-11, 2010.

[24]

[25] T. Schaul, “Benchmarking exponential natural evolution strategies on the noiseless and noisy black-box optimization testbeds,” in *Proceedings of the 14th Genetic and Evolutionary Computation Conference*, Philadelphia, PA, July 7-11, 2012. http://dx.doi.org/10.1145/2330784.2330816

[26]

[27] Y. Wang, B. O’Donoghue, and S. Boyd, “Approximate dynamic programming via iterated Bellman inequalities,” *International Journal of Robust and Nonlinear Control*, February 19, 2014, in press. http://dx.doi.org/10.1002/rnc.315

[28] B. O’Donoghue, W. Yang, and S. Boyd, “Min-max approximate dynamic programming,” in *Proceedings of the IEEE International Symposium on Computer-Aided Control System Design*, Denver, CO, September 28-30, 2011, pp. 424-431. http://dx.doi.org/10.1109/CACSD.2011.6044538

[29] B. O’Donoghue, Y. Wang, and S. Boyd, “Iterated approximate value functions,” in *Proceedings European Control Conference*, Zurich, Switzerland, July 17-19, 2013, pp. 3882-3888.

[30] A. Keshavarz and S. Boyd, “Quadratic approximate dynamic programming for input-affine systems,” *International Journal of Robust and Nonlinear Control*, vol. 24, no. 3, pp. 432-449, Feb. 2014. http://dx.doi.org/10.1002/rnc.2894

[31] J. Peters and S. Schaal, “Natural actor-critic,” *Neurocomputing*, vol. 71, no. 7-9, pp. 1180-1190, Mar. 2008. http://dx.doi.org/10.1016/j.neucom.2007.11.026

[32] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Belmont, MA: Athena Scientific, 1995.

[33] W. B. Powell, *Approximate Dynamic Programming : Solving the Curses of Dimensionality*, Hoboken, NJ: Wiley-Interscience, 2007.

[34] W. M. Wonham, “Some applications of stochastic differential equations to optimal non-linear filtering,” SIAM Journal on Control, vol. 2, pp. 347-369, 1965.

[35] S. P. Boyd, *Linear Matrix Inequalities in System and Control Theory*, Philadelphia, PA: Society for Industrial and Applied Mathematics, 1994

Jooyoung Park received his BS in Electrical Engineering from Seoul National University in 1983 and his PhD in Electrical and Computer Engineering from the University of Texas at Austin in 1992. He joined Korea University in 1993, where he is currently a professor at the Department of Control and Instrumentation Engineering. His recent research interests are in the areas of machine learning, control theory, and financial engineering.
E-mail: parkj@korea.ac.kr

Jungdong Lim received his BS in Control and Instrumentation Engineering from Korea University in 2013. Currently, he is a graduate student at Korea University majoring in Control and Instrumentation Engineering. His research areas include approximate dynamic programming, machine learning, and control applications.
E-mail: huhuvwvw@korea.ac.kr

Wonbu Lee received his BS in Control and Instrumentation Engineering from Korea University in 2013. Currently, he is a graduate student at Korea University majoring in Control and Instrumentation Engineering. His research areas include approximate inference, machine learning, and control applications.
E-mail: karimia@korea.ac.kr

Seunghyun Ji received his BS in Control and Instrumentation Engineering from Korea University in 2014. Currently, he is a graduate student at Korea University majoring in Control and Instrumentation Engineering. His research areas include control theory and machine learning.
E-mail: mysky5871@korea.ac.kr

Keehoon Sung received his BS in Control and Instrumentation Engineering from Korea University in 2014. Currently, he is a graduate student at Korea University majoring in Control and Instrumentation Engineering. His research areas include control theory and machine learning.
E-mail: skh0910@korea.ac.kr

Kyungwook Park received his BBA and MBA from Seoul National University and his PhD in Finance from the University of Texas at Austin in 1993. He joined Korea University in 1994, where he is currently a professor at the School of Business Administration. His recent research interests are in the areas of derivatives and hedging, control-theory-based assets and derivatives management, and cost of capital estimation with derivative pricing.
E-mail: pkw@korea.ac.kr