

# Asset Allocation via Machine Learning

Zhenning Hong<sup>1</sup>, Ruyan Tian<sup>2</sup>, Qing Yang<sup>3</sup>, Weiliang Yao<sup>4</sup>, Tingting Ye<sup>5</sup> & Liangliang Zhang<sup>6</sup>

<sup>1</sup> Director and the Head of Quantitative Investment at the Asset Management Division, Dongxing Securities, Co., Ltd. Yangshupu Rd, No. 248, Hongkou District, Shanghai, China

<sup>2</sup> Ph.D. student, School of Economics, Fudan University, Guoquan Rd. No. 600, Yangpu District, Shanghai, China

<sup>3</sup> Professor of Finance at School of Economics, Fudan University, Guoquan Rd. No. 600, Yangpu District, Shanghai, China

<sup>4</sup> M.S. student, School of Economics, Fudan University, Guoquan Rd. No. 600, Yangpu District, Shanghai, China

<sup>5</sup> Assistant of Professor of Accounting at Maine Business School, University of Maine, USA

<sup>6</sup> SVP in the team of Derivatives Trading at the Asset Management Division, Dongxing Securities. Yangshupu Rd, No. 248, Hongkou District, Shanghai, China

Correspondence: Liangliang Zhang, Asset Management Division, Dongxing Securities, Co., Ltd., Yangshupu Rd, No. 248, Hongkou District, Shanghai, China.

*The authors are indebted to the MOE (Ministry of Education of China) General Program of Humanities, Philosophies and Social Sciences "A Study on the Sentiment and Behavior of Financial Market Investors Based on Deep Learning Technology" (Grant No.19YJA790104) for providing financial support.*

Received: October 7, 2021

Accepted: November 8, 2021

Online Published: November 10, 2021

doi:10.5430/afr.v10n4p34

URL: <https://doi.org/10.5430/afr.v10n4p34>

## Abstract

In this paper, we document a novel machine learning-based numerical framework to solve static and dynamic portfolio optimization problems, with, potentially, an extremely large number of assets. The framework proposed applies to general constrained optimization problems and overcomes many major difficulties arising in current literature. We not only empirically test our methods in U.S. and China A-share equity markets, but also run a horse-race comparison of some optimization schemes documented in (Homescu, 2014). We record significant excess returns, relative to the selected benchmarks, in both U.S. and China equity markets using popular schemes solved by our framework, where the conditional expected returns are obtained via machine learning regression, inspired by (Gu, Kelly & Xiu, 2020) and (Leippold, Wang & Zhou, 2021), of future returns on pricing factors carefully chosen.

**Keywords:** portfolio optimization, machine learning, hierarchical clustering, k-means clustering, deep learning regression, mean-variance-skewness-kurtosis, reinforcement learning, monte-carlo simulation, top-down and bottom-up approaches

## 1. Introduction

### 1.1 Outline of This Paper

In this paper, we propose a novel Monte-Carlo simulation and machine learning-based (unsupervised and supervised learning) static and dynamic portfolio optimization framework. This framework supports arbitrary objective functions and constraints, which can be either linear or nonlinear. Moreover, the number of assets being considered can be very large. Our methodology is fast, accurate and convergent to the global optimum under mild assumptions, which are easy to verify for static portfolio optimization problems, extending the original method introduced in (Zhang, 2020b).

The framework consists of an input preparation module, a hierarchical clustering-based asset space decomposition module, a simulation and portfolio weights computation module and a profit and loss evaluation module. We will mainly introduce the second and third modules as they are of most importance to this paper. The input preparation module applies supervised-learning approach to estimate the inputs that feed into the portfolio optimization module. The hierarchical or regular clustering-based asset space decomposition module tries to partition the entire asset universe, which often includes a large number of individual assets, based on a predetermined set of firm characteristics. This results in similar factor values for the assets in each cluster. The third module, which is the

simulation and portfolio weights computation module, generates uniformly distributed portfolio weights in the constrained region and selects the globally optimal one corresponding to a certain objective utility function with nonlinear constraints. The method in this module is fast and convergent under mild assumptions. Most importantly, this module does not require the computation of joint higher order moments (Note 1) of a random vector (e.g., the rate of returns), which is computationally expensive especially in high dimensions. The last module evaluates the profit and loss of the selected portfolios and performs backtesting. We empirically test our methodology with numerical examples covering various objective functions and constraints, in U.S. and Chinese stock markets.

### *1.2 Literature Review*

Portfolio optimization is an important topic in financial economics that attracts both academic researchers and financial practitioners. Starting from the seminal work of modern portfolio theory in (Markowitz, 1952), where it was suggested that correlations between different assets should be included as inputs to portfolio optimization practice, in addition to volatility and expected returns, the theory of portfolio selection has become rigorous science rather than art. The modern portfolio theory later inspired the famous CAPM model proposed in (Treyner, 1962) and later extended in (Merton, 1973) to a dynamic setting. Through years, we have witnessed an explosion of the number of results on portfolio optimization as the advancement of theoretical and empirical research, and the increase of computational power. Refinements on modern portfolio theory have been studied in (Brodie, Daubechies, De Mol, Giannone & Loris, 2008) in a context of sparse portfolios that stabilizing the output of mean-variance model. The carrier portfolio strategy, described in (Kusiak, 2013), is another approach which can deliver sparse and stable portfolios. It relies on a simple, mathematical linear program that directly considers and treats each return observation as individual data, with no assumptions on joint return distributions. Other examples, in a single period setting, include the Black and Litterman approach that was extended by (Black & Litterman, 1992), in which a Bayesian type analysis was used to incorporate investors' ex-ante views that correct the equilibrium CAPM expected asset returns. Because the first order moments of asset returns are notoriously hard to estimate, simple return-agnostic strategies were created to account for this phenomenon. For example, the minimum variance portfolio and equal weight (1-over-N) portfolio are studied in (Demiguel, Plyakha, Uppal & Vilkov, 2013) and (Allen, Lizieri & Satchell, 2012). In addition to the first and second moments, there have been strategies based on higher order moments, namely, skewness and kurtosis, appearing in the work of (Harvey, Liechty, Liechty & Mueller, 2010) and many others. The computation of higher order co-moments may rely on factor models, such as the single-factor method considered in (Martellini & Ziemann, 2010), or forward-looking information obtained from option prices as documented in (Christoffersen, Jacobs & Chang, 2012). But whichever approach we choose, the methods suffer greatly from the curse of dimensionality as the dimension of asset span increases. Since the proposal of (Ang, 2013) on factor investing, factor-based strategies start to emerge. References can be found in (Koedijk, Slager & Stork, 2014) and (Roncalli & Weisang, 2012), among others. Additional return agnostic, or risk-based strategies, such as maximum diversification strategy, risk-parity strategies, and their variations have been proposed. Research results can be found in (Anderson, Bianchi & Goldberg, 2014) and references therein. Another strand of literature focuses on dynamic portfolio allocation in a randomly varying market environment and considers multi-period optimization problems. The pioneering work attributes to (Merton, 1971) with numerous later refinements, see (Cvitanic & Karatzas, 1992) and (Schroder & Skiadas, 1999) for example. The characteristics of this type of problems are that they often involve dynamic programming and in a continuous time setting, a PDE, or equivalently a BSDE, system often needs to be solved.

Theoretically being sound, the Markowitz's mean-variance optimization method, still popular in financial industry, has many obvious drawbacks, which are well-documented in the literature, for example, (Homescu, 2014) and (Perrin & Roncalli, 2019). There are three major difficulties with respect to the mean-variance approach. The first is that the method output is extremely sensitive to the model input, which is often difficult to estimate accurately. The second is that the analytical solution of the quadratic programming problem involved requires the computation of a large covariance matrix and its inverse, when the number of assets is large. Third, the computational burden increases sharply if multiple linear or nonlinear constraints are added. To address the input estimation problem, many deep-learning based methods have been proposed recently, such as (Yang, Liu & Wu, 2018), (Gu, Kelly & Xiu, 2020), (Babiak & Barunik, 2020) and references therein. To address the second and third questions, (Perrin & Roncalli, 2019) reviewed machine learning optimization approaches to solve the constrained quadratic programming problem in high dimensions. In (Homescu, 2014), general formulations of static portfolio optimization are outlined, taking into consideration the reward, risk and various constraints on optimal portfolios. Compared to the reference in the literature, our method enjoys all the advantages while being theoretically simple and easy to implement in practice.

### 1.3 Main Contributions

The contributions of this paper are five-folds. First, it proposes a fast and convergent numerical framework, which is universal and applies to arbitrary constrained optimization problems with unique solutions without calling explicitly any optimization routines, unlike the current problem-specific deep learning-based methods in the literature. Second, our methodology involves no estimation of cross higher order moments of the asset span by its construction. This is crucial for the methodology to overcome the curse of dimensionality when higher order moments are involved, and the number of assets is large. Third, the paper proposes to use the (hierarchical) clustering method to reduce the dimension of the optimization problem, when there are many assets in the portfolio. Fourth, the paper advocates using deep learning techniques to estimate the model inputs. Fifth, we provide empirical studies on portfolio choice among a large number of stocks in China and U.S. equity markets and evaluate the performance.

### 1.4 Organization of This Paper

The paper is organized as following. Section 2 describes the methodology and the optimization framework. Section 3 performs numerical experiments and Section 4 concludes.

## 2. Main Methodology

In this section, we document and present first three of the aforementioned 4 modules. As the first step, we prepare the inputs for the optimization process, which are often the first and higher order moments of asset return vectors. Second, we decompose the asset span, which usually consists of thousands of assets, with the help of a set of risk factors, into small subsets in a hierarchical manner, perform optimization on each sub-level and obtain the final optimal weight on every asset based on the intermediate weights on each cluster. Third, most importantly, we perform portfolio optimization at each clustering level in the hierarchy based on Monte-Carlo simulation. The last step is to compute the results of the back testing and present the transaction cost at each point in time. Theoretical convergence results are provided in this section and we will illustrate how to perform portfolio choice numerically for static problems.

### 2.1 Input Preparation and Orthogonalization

The current portfolio choice literature often requires users input various moments of the asset returns. To compute the  $\alpha$ th order conditional moments  $\mathbb{E}_t [R_{t+h}^\alpha]$ , where

$R = (R^1, \dots, R^n)$  denotes the vector of asset returns and  $\alpha = (\alpha_1, \dots, \alpha_n)$  represents the standard

multi-index notation, i.e.,  $R_{t+h}^\alpha = \prod_{i=1}^n [R_{t+h}^i]^{\alpha_i}$  and  $|\alpha| = \sum_{i=1}^n \alpha_i$ , we first look at the general semi-martingale decomposition below, written in matrix and vector notations

$$R_{t+h} = \mathbb{E}_t [R_{t+h}] + (R_{t+h} - \mathbb{E}_t [R_{t+h}]) \quad (1)$$

$$= \mu_t + \sigma_t U_{t,t+h}. \quad (2)$$

Here the random source term  $U_{t,t+h}$  satisfies  $\mathbb{E}_t [U_{t,t+h}] \equiv 0$  and it has unit variance-covariance matrix. We will have the higher order moments of  $U_{t,t+h}$  as functions of  $(\mu, \sigma)$  if we assume that its distribution is elliptic.

Single or multi-variate time series models have been proposed to estimate  $(\mu, \sigma)$ , and even higher order joint moments of the asset return vector. From the simplest single variable linear autoregressive (AR) and moving average models (MA) of order 1 to vector auto-regressive Models (VAR) or even non-linear parametric models, we can develop complicated tools for the estimation purposes, dependent on the availability of sufficient data set. This set of parametric models often assumes a predetermined functional dependency of the moments on the past data. Another route resorts to non-parametric models, to recover the unknown functional relationship between the quantities that we want to estimate and the available data, like the kernel density estimation approach, to name a few. However, the problem with both approaches are that they are either too stringent on the model assumptions, or too sensitive to the data sets chosen for the estimation task. Both suffer severely from the curse of dimensionality problems.

Alternatively, the conditional moments of asset returns can be assumed to be functions of some selected risk factors

$f$ . For example,  $\mu$  and  $\sigma$  can be computed via machine learning methods (see (Gu, Kelly & Xiu, 2020)) or a Monte-Carlo simulation and clustering-based method introduced in (Zhang, 2020). More detailed analysis of the choice of factors and a review of popular regression methodologies can be found in (Gu, Kelly & Xiu, 2020) for U.S. market and (Leippold, Wang & Zhou, 2021) for China A-share market. The machine learning methods, as (Gu, Kelly & Xiu, 2020) reviewed, range from linear regression, LASSO, ridge regression, elastic-net, partial least squares (PLS), to methods introducing non-linearity, such as artificial neural networks, which constitute a big category with deep neural networks, convolution neural networks, recursive neural networks, generative adversal networks, graph-based neural networks and so on. Popular methods also include regression trees, random forests and support vector machines. Those tools listed above are quite flexible and are able to approximate any continuous functions in a compact domain of the Euclidean space. Empirically being effective, the machine learning methods also enjoy certain mathematical rigor, e.g., the universal approximation theorem that neural networks rely upon. One might argue that the machine learning methods are not only complicated to implement but also require huge amounts of data for inference purposes because they often depend on a large number of unknown parameters that one needs to recover. On the contrary, as computational power becomes cheaper and the development of programming languages, those are no longer the obstacles that prevent people from using ML tools. In Python programming language, there are multiple packages, such as keras, sklearn or even xgboost, that enable users to build machine learning structures, i.e., DNN, boosting regression trees, and perform regression or classification prediction tasks. The packages are quite efficient in terms of computational speed with moderate memory requirements for most financial applications. To implement the methods, we need to determine the most effective ML method for the data set we choose, as well as a set of factors, features or characteristics on which we can express the conditional expected moments of asset returns, as a known function. The choice of factors is very important and directly relates to the accuracy of the predictions. In what follows, we will outline the factors in the empirical studies for the completeness of the paper.

In empirical studies, we often have  $|\alpha| = 1$ , meaning that we will only compute the first order moments. The estimation of cross higher order moments, for example, the second order moments, also known as the covariance

matrix, is not necessary. To understand this, consider a lead-lag panel regression of  $(R_{t+1}^i)^2 = g(t, f_t^i) + \varepsilon_{i,t,t+1}^i$ .

The second order moment of the weighted assets  $w \cdot R$  can be represented by  $\mathbb{E}_t [(w \cdot R_{t+1})^2] = g(t, f_t^w)$ , where  $f_t^w$  is the associated factor value of the synthetic asset  $w \cdot R$ . This is an interpolation problem when  $0 < w < 1$ , on which the machine learning methods work well. The same applies to higher order cross moments.

## 2.2 Asset Space Decomposition

In order to apply the bottom-up approach to construct the optimal portfolios, we first use a topdown (hierarchical) clustering method (see (Raffinot, 2019) as an introduction to hierarchical clustering in portfolio choice) to decompose the asset universe into stratified sub-spaces. To understand this, suppose we are dealing with a portfolio optimization problem of 6400 assets, which is a very large number. Each assets  $i$  is associated with a vector of characteristics  $f_t^i$ . Our methodology aims at decomposing the asset universe into, say, 80 small clusters, according to the factor values  $f_t^i$ , meaning, in each cluster, the factor values of the associated assets are sufficiently close (Note 2). On average, there are 80 assets in each of the clusters. If we determine that 80 is still a relatively large number, we can, inside each cluster, divide it into 10 sub-clusters. Now, we have  $80 \times 10 = 800$  clusters with each cluster containing 8 assets on average. Then, we perform portfolio optimization separately in each of the sub-clusters. After portfolio optimization is done on each of the sub-clusters, we treat each optimized sub-cluster as one synthetic asset based on the optimal weights and perform portfolio optimization among the 10 synthesized assets.

In this way, starting from the sub-clusters of the lowest level, we obtain the optimal portfolio weights based on the parameters input from Section 2.1 and the methodology to be illustrated in Section 2.3. By working from the lowest to highest level, we will obtain the optimal portfolio weights corresponding to each of the intermediate sub-spaces, and finally the weights on each asset.

The above analysis uses iterated clustering approach to build the hierarchy of clusters. Of course, hierarchical clustering can be directly applied here. In addition to the clustering approach, we can create various criteria, industry classification, as an example, based on the factor values to partition the asset universe into small buckets, with the assets in each bucket representing some similar behaviors (Note 3).

### 2.3 Portfolio Optimization

#### 2.3.1 Static Portfolio Optimization

##### The Algorithm

Assume that we are going to maximize an objective function  $G(w, R)$  (Note 4), where  $w$  denotes the portfolio weights and  $R$  is the rate of return vector of a certain class of assets. There are constraints on  $w$ , namely

$$F(w) \geq 0 \tag{3}$$

$$H(w) = 0 \tag{4}$$

$$w \in (a, b). \tag{5}$$

$F$  and  $H$  are, potentially, nonlinear functions of  $w$ . A general formulation of the static portfolio optimization problems can be found in (Homescu, 2014). In this section, we outline a simulation and machine learning-based approach to obtain the optimized weights  $w$ . Assume that Equation (4) can be rewritten

$$w^n = h(w^1, w^2, \dots, w^{n-1}). \tag{6}$$

The method works as following:

1. Generate  $M \times (n - 1)$  uniform random numbers  $\{w_m^j\}_{j=1, m=1}^{n-1, M}$ , which satisfy  $a^j < w_m^j < b^j$
2. Compute  $w_m^n = h(w_m^1, w_m^2, \dots, w_m^{n-1})$  for  $m = 1, 2, \dots, M$
3. Find a subset of  $\{w_m^j\}_{j=1, m=1}^{n-1, M}$  that satisfies Equation (3) and denote it by  $S^F$
4. Use hierarchical or regular clustering method to decompose  $S^F$  into  $K$  disjoint clusters, denoted by  $\{S_k^F\}_{k=1}^K$
5. Denote the center of  $\{S_k^F\}_{k=1}^K$  by  $\{\bar{w}^k\}_{k=1}^K$  and compute  $k^* = \operatorname{argmax}_{1 \leq k \leq K} [G(\bar{w}^k, R)]$
6. Use hierarchical or regular clustering method to decompose  $S_{k^*}^F$  into  $K$  disjoint clusters, denoted by  $\{S_k^{F, k^*}\}_{k=1}^K$
7. Repeat Step 4 to Step 6 until convergence.

Sometimes, the constraints can be complicated, highly nonlinear and therefore difficult for users of our framework to generate valid samples falling in the constrained region. A brute-force way, of course, is to generate sample weights and only keep those in the valid hyper-surface. Alternatively, we can transform the problem to make the users life easier. Consider the same problem as shown above in Equations (3), (4) and (5). For Constraint (3), we add an indicator function  $\lambda_1 \mathbf{1}_{F(w) \geq 0}$  to the objective. For Constraint (4), we can introduce an  $L^2$ -regularization type term  $\lambda_2 \|H(w)\|_{L^2}$  to the objective function. Last, the Constraint (5) can be addressed by introducing, again, an indicator function  $\lambda_3 \mathbf{1}_{w \in (a, b)}$ . To summarize, the new objective function is, then,  $G(w, R) + \lambda_1 \mathbf{1}_{F(w) \geq 0} - \lambda_2 \|H(w)\|_{L^2} + \lambda_3 \mathbf{1}_{w \in (a, b)}$ , where the candidate weights  $w$  lie in the whole Euclidean space.  $(\lambda_1, \lambda_2, \lambda_3)$  are three positive real numbers, representing the importance of the constraints.

It would be interesting to discuss the utility loss introduced by this hierarchical construction. It can be easily understood that, at the last level of clustering, we perform optimization for each of the subsets and the final global weights are proportional to the weights in each final cluster. Of course, this methodology is sub-optimal compared to the global optimization on the whole asset universe.

However, we have gains in terms of a faster computational speed, less resources requirement, less severe propagation of estimation errors, and the elimination of the potential corner solutions or local optimum. Moreover, the portfolio optimization is done in a bottom-up way, but the clustering is top-down, therefore our methodology enjoys the benefit of both approaches. Moreover, the reason to use clustering approach in Step 4 above is to reduce the computation burden when M is very large and the evaluations of objective function are time consuming.

**Theoretical Convergence**

In this section, we discuss the global convergence of the proposed approach based on three critical assumptions below.

**Assumption 1 (Completeness)** *The random number generator  $\Upsilon$  satisfies the following. Suppose that the size of random numbers generated is  $N$ , and the random numbers generated by  $\Upsilon$  form a set  $\beta_N$ . Then we have the fact that  $\cup_{N=1}^{\infty} \beta_N$  is always dense in the compact set  $S^F$*

Assumption 1 ensures that, as we sample more random realizations for the portfolio weight vectors, any point in  $S^F$  is reachable with the samples generated. Therefore, the solution obtained is truly the global optimum.

**Assumption 2 (Existence and Uniqueness)** *The constrained optimization problem  $O(w)$  introduced in Section 2.3.1 has a unique solution.*

**Assumption 3 (Continuity)** *The solution to the optimization problem is continuous with respect to  $\beta_N$ . This means that*

$$\lim_{K \rightarrow \infty} \operatorname{argsup}_{\cup_{N=1}^K \beta_N} O(w) = \operatorname{argsup}_{\cup_{N=1}^{\infty} \beta_N} O(w) \tag{7}$$

where  $O(w)$  is the original optimization problem with constraints.

This assumption above ensures that, given more simulated samples, the approximate solution will converge to the true one. The assumption is the property of the optimization problem itself and should be verified in a case-by-case manner. Because of the smooth dependency of the solution of, for example, the mean-variance optimization problem to its input variables, it is easy to see that the above three assumptions are automatically satisfied. Combining the above three assumptions, we have the theorem below as our main theoretical result.

**Theorem 1 (Global Convergence)** *Under Assumptions 1, 2 and 3, our algorithm output is convergent to the unique global optimal solution.*

2.3.2 Dynamic Portfolio Optimization

In a dynamic portfolio optimization problem, we try to solve the Bellman equation

$$V^\pi(s) = R(s) + \gamma \times \left[ \max_{\pi} \sum_{s' \in S} P_{s, \pi(s)}(s') V^\pi(s') \right] \tag{8}$$

where  $R$  is the immediate reward function,  $\pi : S \rightarrow A$  is a mapping from the state space  $S$  to the action space  $A$  and is called the policy function.  $P_{s, \pi(s)}(s')$  denotes the probability transition matrix and  $V^\pi(s)$  is the value function. Last,  $\gamma \in (0, 1)$  is the discount factor process, which is often taken as a constant. The goal is to find an optimal policy function  $\pi$  such that the value function is maximized. Of course, in general,  $\pi$  is nonlinear in both time  $t$  and state  $s$  (Note 5). However, it can be approximated locally in an open and sufficiently small region by its tangent space, which is represented by a linear equation. Further suppose that the state space  $S$  and action space  $A$  are compact sub-spaces of Euclidean space and we can generate uniform random numbers in  $S$  and  $A$ . Decompose

$U = S \cup A$  into small disjoint sub-spaces  $\{U_k\}_{k=1}^K$ , and we have  $\pi(s)|_{U_k} \cong \delta_k^0 + \delta_k^1 \cdot s + \varepsilon_k$ , where  $\varepsilon_k$  is

the approximation error term. The functional form of  $\pi$  is solely determined by  $(\delta_k^0, \delta_k^1)$  for each  $k$ . For the transition probability matrix  $P_{s,\pi(s)}(s')$ , one way to represent it is to assume a parametric model  $s_{t+1} = f(s_{0:t}, a_{0:t}, e_{t,t+1})$ , where  $s_{0:t} = (s_0, s_1, \dots, s_t)$  and likewise for  $a_{0:t}$ . To solve the optimization problem in Equation (8), we generate  $M$  independent copies of  $(\delta_k^0, \delta_k^1)_{k=1}^K$ , therefore  $M$  different functional forms of  $\pi$ , and for each copy, compute the value function via Monte-Carlo simulation based on the data generating process for  $s_t$  and Equation (8), and use the method proposed in Section 2.3.1 to determine the best choice of  $(\delta_k^0, \delta_k^1)_{k=1}^K$  among the  $M$  independent samples. Last, the data generating process for  $s_t$  can, alternatively, be replaced by a non-parametric inference directly using historical relationship.

### 3. Empirical Studies

#### 3.1 Simulated Data and Static Portfolio Choice

Assume that there is an  $m$ -dimensional vector process  $f_t$ , whose data generating process (DGP) is

$$f_{t+h} = g(f_t, \vartheta_t) + e_{t,t+h}^f \tag{9}$$

Where  $\vartheta$  is another stochastic process (Note 6) and  $\mathbb{E}_t [e_{t,t+h}^f] = 0$ . The asset return vector is denoted by  $r_t$ , which is  $n$ -dimensional. We have approximately the following regression relationship

$$r_t = h(f_t) + e_t^r \tag{10}$$

Here  $e_t^r$  is considered as a small perturbation term, which mimicks the effect of missing factors or measurement errors. Further observe that

$$\mathbb{E}_t [r_{t+h}] = h(t, h, f_t, \vartheta_t) + u_{t,t+h}^{r,f} \tag{11}$$

where  $h$  is potentially a nonlinear function of  $(f, \vartheta)$  and  $u_{t,t+h}^{r,f}$  is the pricing error term. The detailed configurations are described below. The DGP for the factor process of each stock is chosen as an ARMA-GARCH model

$$f_t = \mu + \phi f_{t-h} + \sigma_t \epsilon_t \tag{12}$$

$$\sigma_t^2 = \alpha + \beta \sigma_{t-h}^2 + \gamma f_{t-h}^2 \tag{13}$$

The factor  $f$  is  $n$ -dimensional (Note 7),  $\mu < 0.05$  is  $n$ -dimensional,  $0 < \phi < 1$  is  $n \times n$ ,  $\sigma$  is  $n \times 1$  and the error term  $\epsilon_t = P \cdot u_t$ , where the correlation generator  $P$  is an  $n \times n$  lower triangular matrix with squared sum of each row being 1.  $u_t$  is an  $n$ -dimensional independent Gaussian process with mean 0 and variance

1. The parameter set  $(\mu, \phi, \alpha, \beta, \gamma, P)$  is generated randomly according to uniform distributions and the values ensure that the ARMA-GARCH models are stationary. The  $n$ -dimensional return process satisfies  $r_t = 0.02 \times \sin(f_t)$  (Note 8) and  $\epsilon \cong \text{Unif}(-0.0015, 0.0015)$  is an  $n$ -dimensional uniformly distributed random



vector serving as the perturbation term, accounting for missing factors or measurement errors. The number of factors is 1, the number of assets  $n = 1000$  and the number of time periods  $T = 250$ . The number of clusters is  $\lceil \sqrt{1000} \rceil$ , i.e., the integer part of  $\lceil \sqrt{1000} \rceil$ . The portfolio weights are constrained within  $[0, 1]$  and sum up to 1. The objective function is the classical mean-variance quadratic one. The equity curve of the out-of-sample optimization results is displayed in Exhibit 1 below.

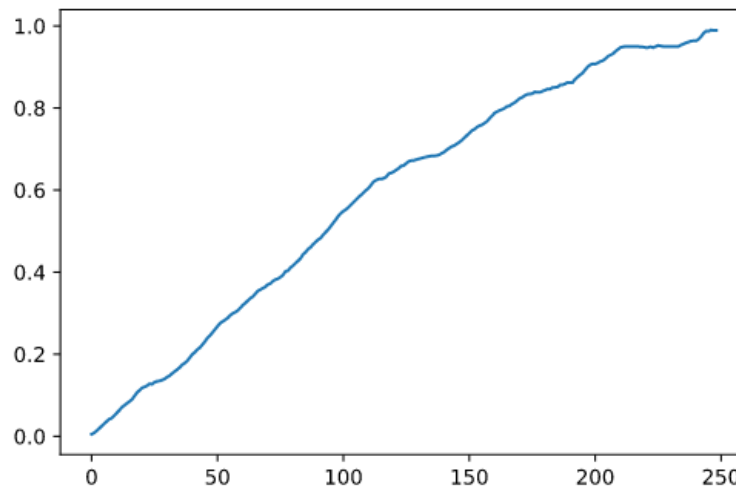


Figure 1. Equity Curve for Simulation Study

It is obvious that the method performs well in an artificial simulation environment, according to the equity curve. The  $x$ -axis is the number of periods, which is up to 250, and the  $y$ -axis is the value of the equity curve. From the plot we can see that the equity curve increases stably from 0 to approximately 1 with very limited drawdowns. This is inline with our expectations: in a simulation environment, we know and are able to recover exactly the functional relationship between factors and expected future returns. Of course, there are also small negative returns appearing along the equity curve. This is caused by the small perturbation term  $e_t^r$  in Equation (10) and the fact that the realization of future returns can deviate from their expected values, which is illustrated by the pricing error term  $u_{t,t+h}^{r,f}$ .

### 3.2 Static Portfolio Optimization in U.S. Equity Market

#### 3.2.1 The Data

The data are from WRDS. Stock returns, prices, and the number of shares outstanding are downloaded from CRSP monthly files, while the accounting data are from Compustat fundamentals annual files. We exclude financial firms (SIC code between 6000 and 6999) from the sample and consider only NYSE, AMEX, and NASDAQ firms with common stocks (SHRCD = 10 or 11). We sample accounting data used for the construction of characteristics in calendar year  $t$  from the statements

with the fiscal year end  $t-1$ . Based on the data that are available from CRSP, Compustat, IBES, and researchers' websites, we analyze the out-of-sample predictability of 94 firm characteristics that are collected from (Green, Hand & Zhang, 2017). The sample period ranges from 198001 to 201812.

#### 3.2.2 The Methodology

To carry out the analysis, some details have to be determined. The objective function is  $f(w) = \frac{\mu(w)}{\sigma(w)}$ , where  $(\mu, \sigma)$



are the conditional expected return, computed via machine learning regression, and empirical volatility.  $\sigma$  is the empirical value computed for every simulated portfolio weight vector  $w$  using past 6 months' asset return data. Portfolio weights are constrained within (0; 1) and they sum up to 1. The number of sample weights is 1; 000; 000 universally and that of the clusters is the squaredroot of the number of assets to be optimized. The conditional expected returns are estimated via a 94-factor (Note 9) regression-based asset pricing model implemented with Python function XGBRegressor provided by module xgboost. The cross-section regression is done in a rolling window manner. The prediction is based on monthly data. The clustering is done by the scores computed via equal weights on the factor values. The optimal portfolios are generated at the beginning of each period and are held until the end of the period.

### 3.2.3 The Results

The backtesting results are summarized in the equity curve plot in Figure 2, where the upper plot contains log-net equity curve of portfolio optimization strategies and equal weight portfolios and the lower plot represents the excess returns of the portfolio choice strategies relative to the  $\frac{1}{N}$ -portfolio, and the performance metrics in Table 3. From Figure 2, we can see that the mean-variance strategy performs best, and is followed by minimum variance strategy.

Both curves are above the  $\frac{1}{N}$ -portfolio curve. The performance of all three curves (Note 10) are significantly above the S&P500 index. Table 3 contains performance metrics. Vol means annualized volatility, IR means information ratio, SR is Sortino ratio, MDD represents maximum drawdown and CR stands for Calmar ratio.

As is known, the quality of the optimized portfolio depends crucially on the inputs. An accurate estimate of conditional expected returns and volatilities ensures good performance of portfolio optimization results. We attribute the good performance in U.S. market to a long period of bull market, which makes the prediction easier, and the asset pricing factors carefully chosen, which can be a good set of proxies for prediction purposes. Last, our numerical framework enables a fast and efficient computation of the optimal portfolios.

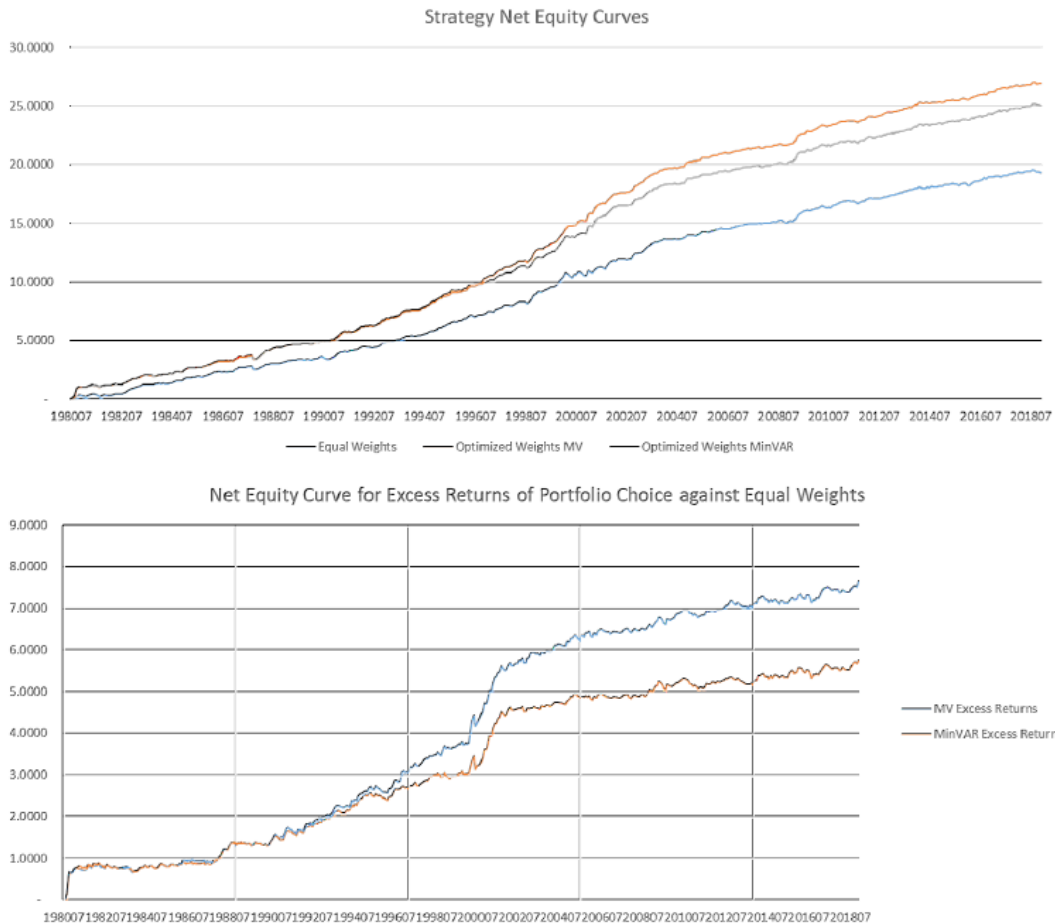


Figure 2. log-Equity Curves for U.S. Stock Market, TOP 100 Stocks

Table 3. Performance Metrics in U.S. Stock Market. MinVAR means minimum variance and MV stands for mean-variance

Index	Return	Vol	IR	SR	MDD	CR
MV100	77:44%	36:07%	2:14	5:88	29:12%	2:65
MinVAR100	71:63%	34:22%	2:09	6:48	21:34%	3:35
S&P500	11:88%	14:82%	0:80	1:07	50:21%	0:23

### 3.3 Static Portfolio Optimization in China A-Share Market

#### 3.3.1 The Data

The data for backtesting are downloaded from UQER platform. The stocks that we study are listed on Shanghai and Shenzhen Stock Exchanges and are included in CSI800 index. We exclude the firms marked ST or have been through the IPO process for less than two months. Similar to the U.S. case, we take fundamental data used for the construction of characteristics in calendar year  $t$  from the financial statements with the fiscal year ending in year  $t-1$ . Due to the difference in data availability and market structure between China and U.S., we resort to the appendix of (Fang, 2019), instead of (Green, Hand & Zhang, 2017), for the construction of firm characteristics. The factors used by (Fang, 2019) are largely the same as those used by (Green, Hand & Zhang, 2017), but (Fang, 2019) has adapted the individual factors based on Chinese market realities. Finally, we constructed 93 firm characteristics. The sample period ranges from 200701 to 202012.

#### 3.3.2 The Methodology

The objective function is  $f(w) = \frac{\mu(w)}{\sigma(w)}$ , where  $(\mu, \sigma)$  are the conditional expected return and empirical volatility

associated with the portfolio  $w, \sigma$  is empirical values computed for every simulated portfolio weight vector  $w$  using past 6 months' asset return data. Portfolio weights are constrained within (0; 1) and they sum up to 1. The number of sample weights is 1; 000; 000 universally and that of the clusters is the squared-root of the number of assets to be optimized. The conditional expected

returns are estimated via a 94-factor (Note 11) lead-lag regression model implemented with Python function RidgeCV provided by module sklearn. The cross-section regression is done in a rolling window manner. The prediction is based on monthly data. The cross section is either the CSI300 or CSI800 index stocks. The optimal portfolios are computed at the beginning of each period and are held until the end of the period.

### 3.3.3 The Results

The backtesting results are summarized in the equity curve plots in Figures 4, 5 and 6 and the performance metrics in Table 7. Here MV means mean-variance, MinVAR represents minimum variance, RP denotes risk parity and EW is  $\frac{1}{N}$ -portfolio. In China A shares market, the simple single period optimization methods reveal significant excess returns as can be observed in Figures 4 to 6. Tables 7 and 8, again, documents the performance metrics. Various risk and reward indexes point out that mean-variance method is the best strategy among the three competing methods and CSI300 index bears minimum IR, SR and CR.

## 4. Conclusion

In this paper, inspired by the methodology introduced in (Zhang, 2020b), we document a novel four-step portfolio optimization framework and test it with simulated and real financial data in U.S. equity and China A-share markets. Our results reveal superior returns over the out-of-sample testing periods

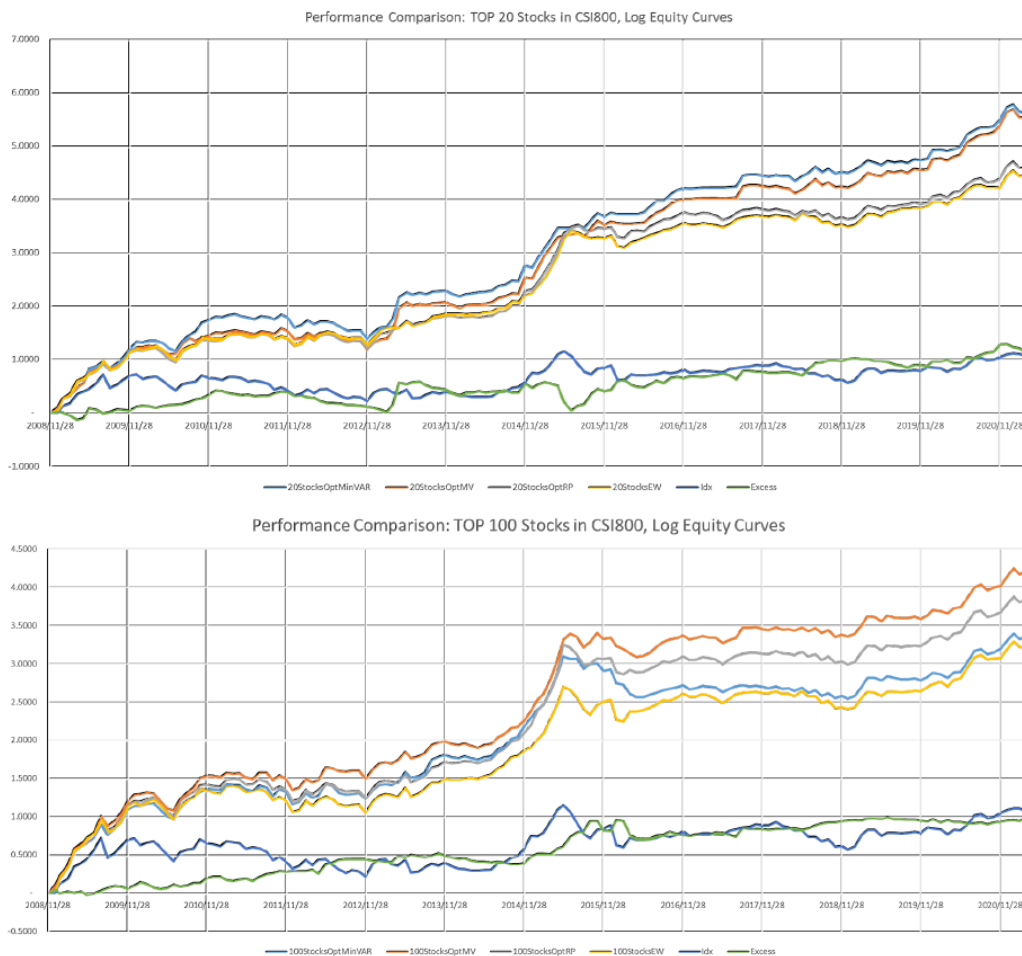


Figure 3. Natural log-Equity Curves for CSI Indexes of China A Share Market

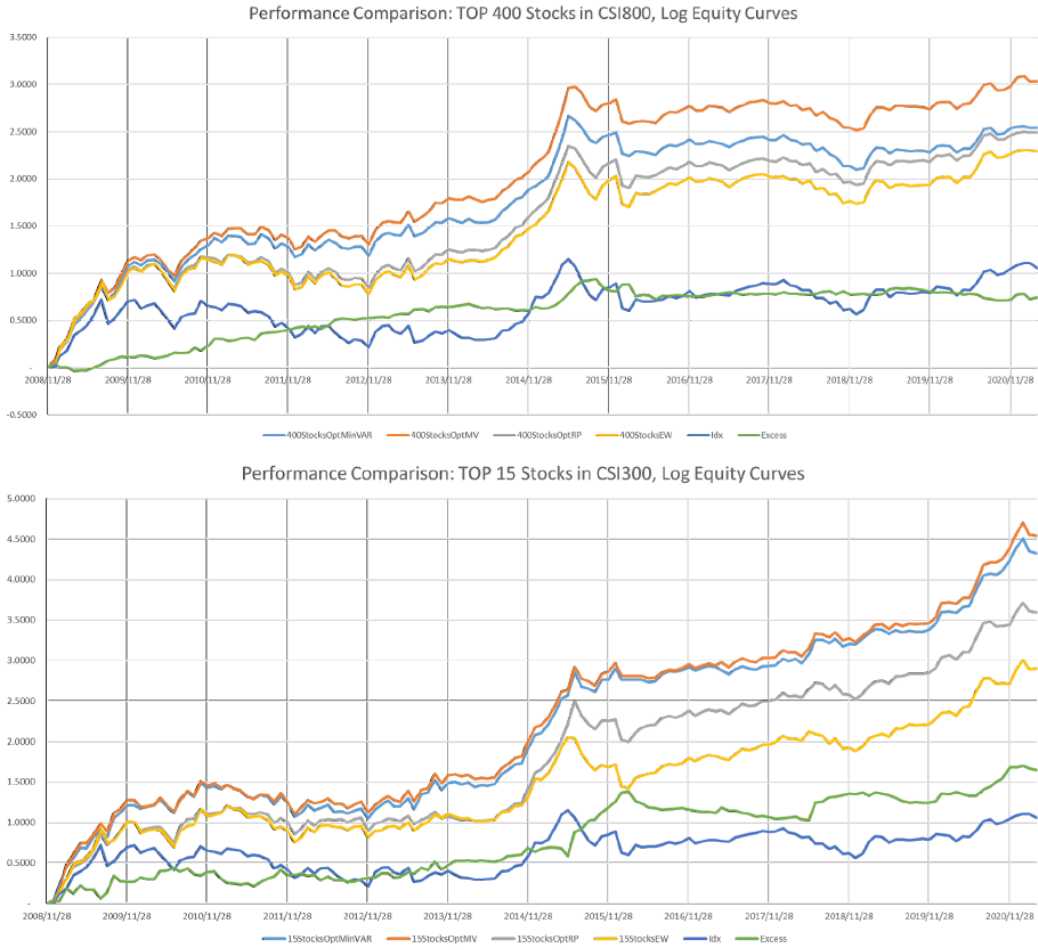


Figure 4. log-Equity Curves for CSI Indexes of China A Share Market

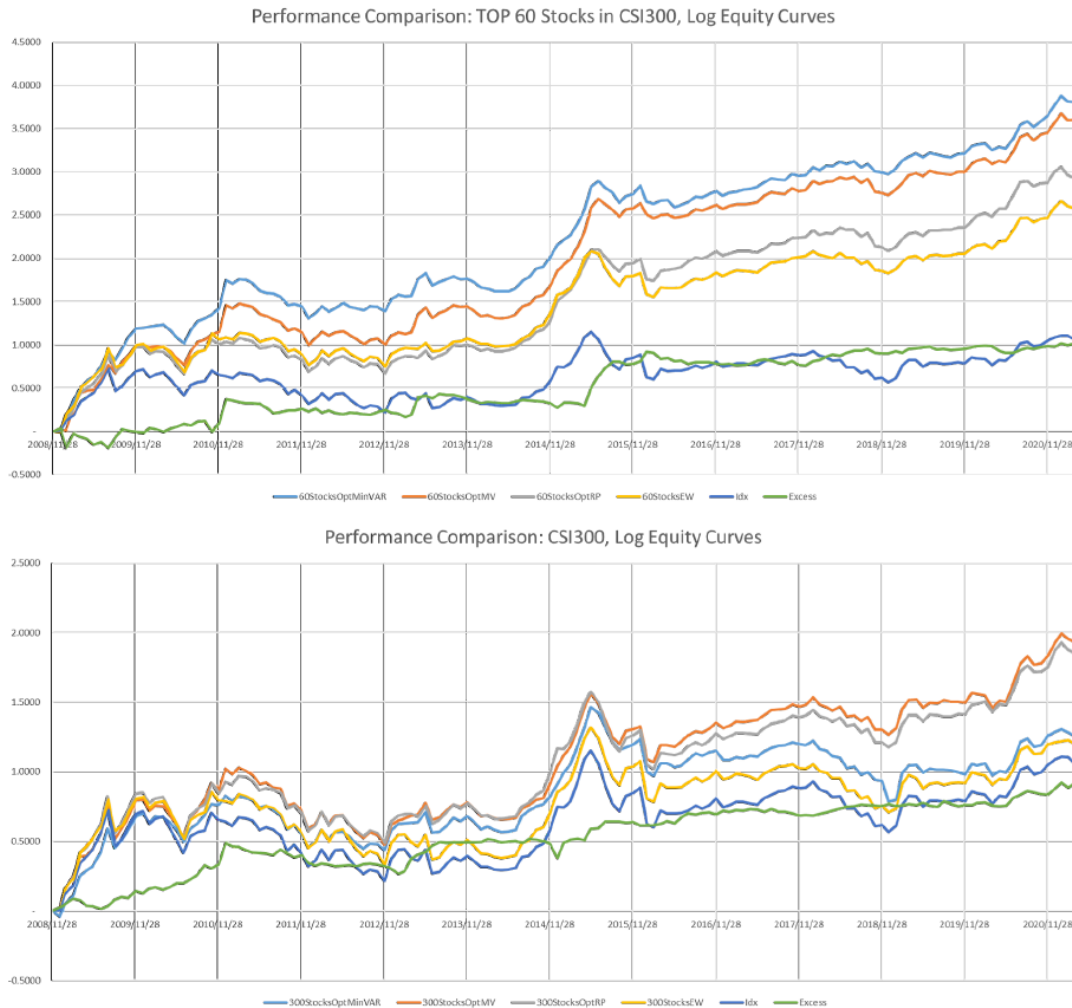


Figure 5. log-Equity Curves for CSI Indexes of China A Share Market

Table 7. Performance Metrics for CSI800 Stocks in China A Share Market

Index	Return	Vol	IR	SR	MDD	CR
CSI800-20StocksMV	50.12%	32.79%	1.52	4.07	32.37%	1:54
CSI800-20StocksMinVAR	50.74%	32.64%	1.55	3.88	36.38%	1.39
CSI800-20StocksRP	41.69%	29.84%	1.39	3.04	23.97%	1.73
CSI800-20StocksEW	40.55%	29.84%	1.39	3.04	27.87%	1.45
CSI800-100StocksMV	37.90%	27.40%	1.38	3.12	26.71%	1.41
CSI800-100StocksMinVAR	30.67%	27.40%	1.16	2.29	42.41%	0.72
CSI800-100StocksRP	35.27%	28.88%	1.22	2.52	32.10%	1.09
CSI800-100StocksEW	29.78%	27.61%	1.07	1.87	36.17%	0.82
CSI800-400StocksMV	27.82%	25.52%	1.08	1.94	37.06%	0.75
CSI800-400StocksMinVAR	23.60%	24.86%	0.94	1.73	43.43%	0.54
CSI800-400StocksRP	23.45%	25.58%	0.91	1.48	35.81%	0.65
CSI800-400StocksEW	22.16%	26.94%	0.82	1.29	37.86%	0.58
CSI800	10.51%	18.87%	0.56	0.97	27.53%	0.38

Table 8. Performance Metrics for CSI300 Stocks in China A Share Market

Index	Return	Vol	IR	SR	MDD	CR
CSI300-15StocksMV	41.83%	31.66%	1.32	2.97	33.01%	1.26
CSI300-15StocksMinVAR	39.75%	30.65%	1.29	2.67	34.56%	1.15
CSI300-15StocksRP	33.27%	28.86%	1.15	2.01	39.61%	0.83
CSI300-15StocksEW	27.59%	28.82%	0.95	1.55	46.47%	0.59
CSI300-60StocksMV	33.06%	28.36%	1.16	2.85	38.03%	0.86
CSI300-60StocksMinVAR	34.68%	27.97%	1.23	2.70	36.85%	0.94
CSI300-60StocksRP	27.16%	26.19%	1.03	1.75	33.72%	0.80
CSI300-60StocksEW	24.60%	27.41%	0.89	1.50	41.30%	0.59
CSI300-300StocksMV	18.65%	24.60%	0.75	1.22	42.55%	0.44
CSI300-300StocksMinVAR	12.57%	22.25%	0.56	0.92	49.18%	0.25
CSI300-300StocksRP	18.28%	25.78%	0.71	1.08	42.67%	0.42
CSI300-300StocksEW	12.99%	25.85%	0.50	0.79	45.57%	0.28
CSI300	10.51%	18.87%	0.56	0.97	27.53%	0.38

in both markets, which illustrates the usefulness of our methodology, that is not only a numerical framework, but also contributes to the literature of large scale optimizations. The empirical study of our proposed dynamic portfolio choice method via reinforcement learning is both interesting and important, yet postponed to future research. In addition, our methodology can be extended to fixed income and option portfolio selection, combining the work of (Zhang, 2020a) and (Zhang, 2020b), which we leave to the interested readers as exercises.

#### Notes

Note 1. For example, covariance matrix or the tensor of third order moments.

Note 2. The definition of being close depends on which distance we choose.

Note 3. For example, we can calculate scores based on a set of predetermined factors for each asset, rank and divide the asset space by the scores.

Note 4. For example, the objective function can be a reward minus a coherent risk measure on  $R$ .

Note 5. We will often assume that time variable  $t$  is included in state vector  $s$ .

Note 6. For example, the DGP can be an ARMA-GARCH process and  $\mathcal{V}$  is therefore the stochastic variance.

Note 7. This means the factor is asset specific and we have only  $m = 1$  factor for each stock.

Note 8. This functional form is to ensure that the return series generated are mostly around  $(-0:02, 0:02)$ .

Note 9. The factors are selected from (Green, Hand & Zhang, 2017).

Note 10. The three curves include mean-variance, minimum variance and equal-weight curves.

Note 11. The factors are constructed in the same way as U.S. equity market.

#### References

- Allen, D., Lizieri, C., & Satchell, S. (2012). 1/N Versus Mean-Variance: What If We Can Forecast. *Working Paper*. <http://dx.doi.org/10.17863/CAM.1020>. –DOI 10.17863/CAM.1020
- Anderson, R., Bianchi, S., & Goldberg, L. (2014). Will My Risk Parity Strategy Outperform? *Financial Analyst Journal*, 75-94. <https://doi.org/10.2469/faj.v68.n6.7>
- Ang, A. (2013). Factor Investing. *Working Paper*. <https://doi.org/10.2139/ssrn.2277397>
- Babiak, M., & Barunik, J. (2020). Deep Learning, Predictability, and Optimal Portfolio Returns. *Working Paper*. <https://doi.org/10.2139/ssrn.3688577>
- Black, F., & Litterman, R. (1992). Global Portfolio Optimization. *Financial Analyst Journal*, 48, Nr. 5, S. 28-43. <https://doi.org/10.2469/faj.v48.n5.28>

- Brodie, J., Daubechies, I., De Mol, C., Giannone, D., & Loris, I. (2008). Sparse and Stable Markowitz Portfolios. *Working Paper*. <https://doi.org/10.1073/pnas.0904287106>
- Christoffersen, P., Jacobs, K., & Chang, B. (2012). Forecasting with Option-Implied Information. *Handbook of Economic Forecasting*, 2(2012). <https://doi.org/10.1016/B978-0-444-53683-9.00010-4>
- Cvitanic, J., & Karatzas, I. (1992). Convex duality in constrained portfolio optimization. *Annals of Applied Probability*, 2(1992), S. 767-818. <https://doi.org/10.1214/aoap/1177005576>
- Demiguel, V., Plyakha, Y., Uppal, R., & Vilkov, G. (2013). Improving Portfolio Selection Using Option-Implied Volatility and Skewness. *JFQA*, 48(2013), Nr. 6, S. 1813-1845. <https://doi.org/10.1017/S0022109013000616>
- Fang, Q. (2019). Replicating Anomalies in China. *SSRN* (2019). <https://doi.org/10.2139/ssrn.3263990>
- Green, J., Hand, J., & Zhang, X. (2017). The Characteristics that Provide Independent Information about Average U.S. Monthly Stock Returns. *Review of Financial Studies*, 30(2017), S. 4389-4436. <https://doi.org/10.1093/rfs/hhx019>
- Gu, S., Kelly, B., & Xiu, D. (2020). Empirical Asset Pricing via Machine Learning. *Review of Financial Studies*, 33(2020), S. 2223-2273. <https://doi.org/10.1093/rfs/hhaa009>
- Harvey, C., Liechty, J., Liechty, M., & Mueller, P. (2010). Portfolio Selection with Higher Moments. *Quantitative Finance*, S. 469-485. <https://doi.org/10.1080/14697681003756877>
- Homescu, C. (2014). Many Risks, One (Optimal) Portfolio. *Working Paper*. <https://doi.org/10.2139/ssrn.2473776>
- Koedijk, C., Slager, A., & Stork, P. (2014). Factor Investing in Practice: A trustees' Guide to Implementation. *Working Paper*. <https://doi.org/10.2139/ssrn.2391694>
- Kusiak, S. (2013). Carrier Portfolios. *Journal of Mathematical Finance*, 40(2013), Nr. 1, S. 61-70. <https://doi.org/10.3905/jpm.2013.40.1.061>
- Leippold, M., Wang, Q., & Zhou, W. (2021). Machine Learning in Chinese Factor Zoo. *SSRN*. <https://doi.org/10.2139/ssrn.3754339>
- Markowitz, H. (1952). Portfolio Selection. *Journal of Finance*, 7(1952), Nr. 1, S. 77-91. <https://doi.org/10.1111/j.1540-6261.1952.tb01525.x>
- Martellini, L., & Ziemann, V. (2010). Improved Estimates of Higher-Order Comoments and Implications for Portfolio Selection. *Review of Financial Studies*, 23(2010), S. 1467-1502. <https://doi.org/10.1093/rfs/hhp099>
- Merton, R. (1971). Optimum Consumption and Portfolio Rules in a Continuous-Time Model. *Journal of Economic Theory*, 3(1971), Nr. 4, S. 373-413. [https://doi.org/10.1016/0022-0531\(71\)90038-X](https://doi.org/10.1016/0022-0531(71)90038-X)
- Merton, R. (1973). An Intertemporal Capital Asset Pricing Model. *Econometrica*, 41(1973), Nr.5, S. 867-887. <https://doi.org/10.2307/1913811>
- Perrin, S., & Roncalli, T. (2019). Machine Learning Optimization Algorithms & Portfolio Allocation. *SSRN*. <https://doi.org/10.2139/ssrn.3425827>
- Raffinot, T. (2019). Hierarchical Clustering Based Asset Allocation. *SSRN*. <https://doi.org/10.3905/jpm.2018.44.2.089>
- Roncalli, T., & Weisang, G. (2012). Risk Parity Portfolios with Risk Factors. *SSRN*. <https://doi.org/10.1080/14697688.2015.1046907>
- Schroder, M., & Skiadas, C. (1999). Optimal Consumption and Portfolio Selection with Stochastic Differential Utility. *Journal of Economic Theory*, 89(1999), Nr. 1, S. 68-126. <https://doi.org/10.1006/jeth.1999.2558>
- Treynor, J. (1962). Toward a Theory of Market Value of Risky Assets. *Working Paper*. <https://doi.org/10.2139/ssrn.628187>
- Yang, H., Liu, X., & Wu, Q. (2018). A Practical Machine Learning Approach for Dynamic Stock Recommendation. *17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (2018)*. <https://doi.org/10.1109/TrustCom/BigDataSE.2018.00253>
- Zhang, L. (2020a). A Clustering Method to Solve Backward Stochastic Differential Equations with Jumps. *Journal of Mathematical Finance*, 10(2020), Nr. 1, S. 1-9. <https://doi.org/10.4236/jmf.2020.101001>



Zhang, L. (2020b). A General Framework of Derivatives Pricing. *Journal of Mathematical Finance*, 10(2020), Nr. 2, S. 255-266. <https://doi.org/10.4236/jmf.2020.102016>

### **Copyrights**

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).