MACHINE LEARNING IN QUANTITATIVE FINANCE

Youngmin Ha

Submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy in Quantitative Finance

Adam Smith Business School
College of Social Sciences
University of Glasgow

October 2017

Abstract

This thesis consists of the three chapters.

Chapter 1 aims to decrease the time complexity of multi-output relevance vector regression from $O\left(VM^3\right)$ to $O\left(V^3+M^3\right)$, where V is the number of output dimensions, M is the number of basis functions, and $V<M$. The experimental results demonstrate that the proposed method is more competitive than the existing method, with regard to computation time. MATLAB codes are available at http://www.mathworks.com/matlabcentral/fileexchange/49131.

The performance of online (sequential) portfolio selection (OPS), which rebalances a portfolio in every period (e.g. daily or weekly) in order to maximise the portfolio's expected terminal wealth in the long run, has been overestimated by the ideal assumption of unlimited market liquidity (i.e. no market impact costs). Therefore, a new transaction cost factor model that considers market impact costs, estimated from limit order book data, as well as proportional transaction costs (e.g. brokerage commissions or transaction taxes in a fixed percentage) is proposed in Chapter 2 for both measuring OPS performance in a more practical way and developing a new OPS method. Backtesting results from the historical limit order book data of NASDAQ-traded stocks show both the performance deterioration of OPS by the market impact costs and the superiority of the proposed OPS method in the environment of limited market liquidity. MATLAB codes are available at http://www.mathworks.com/matlabcentral/fileexchange/56496.

Chapter 3 proposes an optimal intraday trading strategy to absorb the shock to the stock market when an online portfolio selection algorithm rebalances a portfolio. It considers real-time data of limit order books and splits a very large market order into a number of consecutive market orders to minimise overall transaction costs, consisting of market impact costs as well as proportional transaction costs. To be specific, it optimises both the number of intraday tradings and an intraday trading path for a multi-asset portfolio. Backtesting results from the historical limit order book data of NASDAQ-traded stocks show the superiority of the proposed trading algorithm in the environment of limited market liquidity. MATLAB codes are available at http://www.mathworks.com/matlabcentral/fileexchange/62503.

Contents

Introduction

Machine learning is computer programming to optimise a performance criterion using example data or past experience by using the theory of statistics (Alpaydin, 2014), and it has been used in financial markets: financial time series forecasting by using artificial neural networks (ANN) and evolutionary algorithms (Krollner *et al.*, 2010), bankruptcy prediction and credit score modelling (Lin *et al.*, 2012), financial statement fraud detection by using ANN, logistic regression, and support vector machines (SVMs) (Perols, 2011), credit rating based on the combination of ANN and logistic regression (Tsai and Chen, 2010), warning systems for currency crises by using the neuro-fuzzy modelling approach (Lin *et al.*, 2008), and bankruptcy prediction by applying SVMs (Min and Lee, 2005).

This thesis is intended to broaden the usage of machine learning in quantitative finance and consists of the three chapters. Chapter 1 aims to perform multi-input and multi-output (MIMO) nonlinear regression, applicable to multi-step-ahead financial forecasting (e.g. Ticlavilca *et al.* (2010) and Bao *et al.* (2014)), in short computation time. Both Chapter 2 and Chapter 3 aim to maximise investors' wealth when performing portfolio selection in an environment of limited liquidity by using limit order book (LOB) data.

The contributions of this thesis are as follows. Chapter 1 proposes a faster MIMO nonlinear regression algorithm as existing one is computationally time-consuming. Chapter 2 proposes a new transaction cost factor (TCF) model which calculates transaction costs when rebalancing a multi-asset portfolio. The proposed method considers market impact costs (MICs) quantified by LOB data as well as proportional transaction costs, whereas the existing one considers only proportional transaction costs. In addition, the proposed TCF model is employed to measure the performance among different online portfolio selection methods in an environment of limited liquidity. Chapter 3 proposes a new intraday trading algorithm by using the TCF model proposed in Chapter 2. The proposed trading algorithm considers real-time LOB data of all assets in a portfolio to measure the liquidity of all the assets and minimise overall MICs, while the existing ones do not directly use LOB data for optimal trading.

Historical NASDAQ LOB data, not randomly generated LOB data, is utilised to reflect limited liquidity for computer simulations in both Chapter 2 and Chapter 3. To be specific, the data was downloaded from Limit Order Book System: The Efficient Reconstructor (LOBSTER: `https://lobsterdata.com/`) that has NASDAQ LOB data from 27 Jun 2007 to the present. Hence, the backtesting in both the chapters emulates the real world and proves the practical usage of the proposed methods in Chapter 2 and Chapter 3.

MATLAB codes of the computer simulations of all the three chapters have been uploaded on MATLAB Central (`http://www.mathworks.com/matlabcentral/`), and these will not only avoid the potential ambiguity of the proposed algorithms in this thesis but also reduce redundant programming efforts by other researchers for their future research to replicate or modify the computer simulations. Besides, the parallel computing codes (parallel programming on a CPU, not a GPU, is used as CPU programming is much easier than GPU programming) for the Monte Carlo simulations of Chapter 2 and Chapter 3 contribute to the quantitative finance community by showing how to reduce the running time of computationally heavy algorithms.

The most relevant literature of each chapter is as follows. In Chapter 1, multi-output relevance vector regression by Thayananthan (2005, Chapter 6), Thayananthan *et al.* (2008) (it uses the Bayes' theorem and the kernel trick to perform regression), one of MIMO nonlinear regression methods, is the benchmark of the proposed method. In Chapter 2, online portfolio selection with proportional transaction costs but without MICs by Györfi and Vajda (2008) (it rebalances a portfolio daily or weekly under the assumption that stock returns are Markov processes) is the benchmark of the proposed method. In Chapter 3, a path-dependent trading strategy by Lorenz (2008, Chapter 2–3) (it considers an efficient frontier between the expected value and variance of MICs) is the most relevant literature although it cannot be directly compared with the proposed method.

The results of this thesis are as follows. In Chapter 1, the time complexity of the proposed MIMO nonlinear regression algorithm is almost independent of the number of output dimensions, whereas that of the existing algorithm is linearly dependent of it. In Chapter 2, the proposed TCF model is combined with the benchmark OPS method (Györfi and Vajda, 2008), and the combined OPS method is more profitable than the basic one in the real stock market with limited liquidity because the combined method trades less on illiquid stocks when rebalancing a portfolio. In Chapter 3, OPS with the proposed intraday trading algorithm is more profitable than that without the proposed algorithm because the new algorithm considers real-time limit order book data and splits a large maker order into a number of consecutive market orders.

References

Alpaydin, E., *Introduction to machine learning* (3rd edn), 2014, MIT press.

Bao, Y., Xiong, T. and Hu, Z., Multi-step-ahead time series prediction using multiple-output support vector regression. *Neurocomputing*, 2014, **129**, 482--493.

Györfi, L. and Vajda, I., Growth optimal investment with transaction costs. In *Algorithmic Learning Theory*, pp. 108--122, 2008, Springer.

Krollner, B., Vanstone, B. and Finnie, G., Financial time series forecasting with machine learning techniques: A survey. In *Proceedings of the 18th European Symposium on Artificial Neural Networks*, 2010.

Lin, C.S., Khan, H.A., Chang, R.Y. and Wang, Y.C., A new approach to modeling early warning systems for currency crises: Can a machine-learning fuzzy expert system predict the currency crises effectively?. *Journal of International Money and Finance*, 2008, **27**, 1098--1121.

Lin, W.Y., Hu, Y.H. and Tsai, C.F., Machine learning in financial crisis prediction: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2012, **42**, 421--436.

Lorenz, J.M., Optimal trading algorithms: Portfolio transactions, multiperiod portfolio selection, and competitive online search. PhD thesis, ETH Zurich, 2008.

Min, J.H. and Lee, Y.C., Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. *Expert systems with applications*, 2005, **28**, 603--614.

Perols, J., Financial statement fraud detection: An analysis of statistical and machine learning algorithms. *Auditing: A Journal of Practice & Theory*, 2011, **30**, 19--50.

Thayananthan, A., Template-based pose estimation and tracking of 3D hand motion. PhD thesis, University of Cambridge, 2005.

Thayananthan, A., Navaratnam, R., Stenger, B., Torr, P.H. and Cipolla, R., Pose estimation and tracking using multivariate regression. *Pattern Recognition Letters*, 2008, **29**, 1302--1310.

Ticlavilca, A., Feuz, D.M. and McKee, M., Forecasting agricultural commodity prices using multivariate Bayesian machine learning regression. In *Proceedings of the NCCC-134 Conference on Applied Commodity Price Analysis, Forecasting, and Market Risk Management.*, 2010.

Tsai, C.F. and Chen, M.L., Credit rating by hybrid machine learning techniques. *Applied soft computing*, 2010, **10**, 374--380.

Chapter 1. Fast multi-output relevance vector regression

1. Introduction

When it comes to multi-input nonparametric nonlinear regression or classification, the following three methods can be considered: support vector machine (SVM), relevance vector machine (RVM), and Gaussian process (GP) regression or classification.

SVM, invented by Cortes and Vapnik (1995), is a popular machine learning tool. It uses kernel trick (RVM and GP also use the kernel trick) and makes classification and regression computationally efficient for multidimensional inputs. However, its disadvantage is that a user needs to choose a proper value of the error/margin trade-off parameter 'C' (the proper value can be found by k-fold cross-validation).

RVM, invented by Tipping (2001),[1] avoids estimating the error/margin trade-off parameter 'C' of SVM (and in regression, the additional insensitivity parameter 'ε' (Vapnik 2000), or 'ν' (Schölkopf *et al.* 2000)), which wastes computation time. Moreover, RVM allows probabilistic predictions (i.e. both mean and variance of a Gaussian distribution) although SVM predicts only mean values (the error bar estimation of SVM is possible with additional computation (Gao *et al.* 2002, Chu *et al.* 2004)).

GP regression or classification, invented by Gibbs (1997), also does not need estimating 'C' (and the additional parameter of regression 'ε' or 'ν'). Furthermore, the predictive variance of GP regression or classification changes over an input vector $\mathbf{x}_*$: the predictive variance is smaller at the denser region of training samples, while the predictive variance of RVM is almost constant over $\mathbf{x}_*$.

Support vector regression (SVR), relevance vector regression (RVR), and GP regression are for multi-input but single-output regression, and they have been extended as multi-input and multi-output (MIMO) regression to model correlated outputs: multi-output SVR (Pérez-Cruz *et al.* 2002, Vazquez and Walter 2003, Tuia *et al.* 2011), multi-output RVR (Thayananthan 2005, Thayananthan *et al.* 2008), and multi-output GP regression (Boyle and Frean 2004, Bonilla *et al.* 2007, Alvarez and Lawrence 2009).

The multi-output relevance vector regression (MRVR) algorithm by Thayananthan (2005, Chapter 6), Thayananthan *et al.* (2008) uses the Bayes' theorem and the kernel trick to perform MIMO nonparametric nonlinear regression, but it has the limitation of low computational efficiency. Therefore, a new faster algorithm is proposed in this chapter: it uses the matrix normal distribution to model correlated outputs, while the existing algorithm uses the multivariate normal distribution.

The contributions of this chapter are:

- in Section 4, to propose a new algorithm with less time complexity than the existing MRVR algorithm by Thayananthan (2005, Chapter 6), Thayananthan *et al.* (2008);
- in Section 5, to present Monte Carlo simulation results to compare between the existing and the proposed MRVR algorithm in terms of accuracy and computation time.

The rest of this chapter is organised as follows: Section 2 lists related work. Section 3 and Section 4 describe the existing and proposed algorithms of MRVR, respectively. Section 5 shows the experimental results by using MATLAB, and Section 6 gives the conclusion.

[1] Single-output relevance vector regression is easily explained in `http://static1.squarespace.com/static/58851af9ebbd1a30e98fb283/t/58902f4a6b8f5ba2ed9d3bfe/1485844299331/RVM+Explained.pdf`.

2. Related work

The following subsections list the existing research works that have been conducted to reduce the computation time of single-output support vector machine (SVM), relevance vector machine (RVM), and Gaussian process (GP) regression.

2.1. *Single-output SVM*

Guo and Zhang (2007) reduced SVR training time by reducing the number of training samples. Their method consists of the two steps. Firstly, it extracts samples which are more likely to be support vectors from a full training set befor performing SVR training, based on the heuristic observation that the target value of support vector is usually a local extremum or near extremum. Secondly, the extracted samples are used to train a SVR machine. Their simulation results show its computational efficiency. In particular, the computation time of k-fold cross validation of SVR for a large data set can be reduced greatly.

Catanzaro *et al.* (2008) accelerated SVM computation by using both faster sequential algorithms and parallel computation on a graphics processing unit (GPU). To be specific, a sequential minimal optimisation algorithm is used to solve the quadratic programming problem of SVM, and a GPU, whose role has changed from the manipulation of computer graphics and image processing to general-purpose programming, is employed for high throughput floating-point computation.

Chang and Lin (2011) made fast and efficient C++ software of SVM. They reduced the computation time to minimise SVM quadratic programming problems since the quadratic programming is the most time consuming part of SVM. To be specific, shrinking and caching techniques are used. The shrinking technique tries to identify and remove some bounded elements of the SVM dual problem, and the caching technique reduces the computational time of the decomposition of a dense matrix.

2.2. *Single-output RVM*

Tipping and Faul (2003) proposed a new marginal likelihood maximisation algorithm with efficient addition/deletion of candidate basis functions. The efficiency comes from modifying the marginal likelihood function:

$$\mathcal{L}(\boldsymbol{\alpha}) = -\frac{1}{2}\left(N\log(2\pi) + \log|\mathbf{C}| + \mathbf{t}^{\intercal}\mathbf{C}^{-1}\mathbf{t}\right), \tag{2.1}$$

where $\boldsymbol{\alpha} = [\alpha_0\ \alpha_1\ \dots\ \alpha_N]^{\intercal}$ is a hyperparameter, N is the number of training samples, and $\mathbf{t}$ is a target. In the original algorithm by Tipping (2001), $\mathbf{C}$ is written as $\mathbf{C} = \sigma^2\mathbf{I} + \mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^{\intercal}$, where $\mathbf{\Phi}$ is a design matrix, and $\mathbf{A} = \mathrm{diag}\,(\alpha_0, \alpha_1, \dots, \alpha_N)$, whereas in the new algorithm by Tipping and Faul (2003), $\mathbf{C}$ is rewritten as $\mathbf{C} = \sigma^2\mathbf{I} + \sum\limits_{m\neq i}\alpha_m^{-1}\boldsymbol{\phi}_m\boldsymbol{\phi}_m^{\intercal} + \alpha_i^{-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^{\intercal}$, where $\boldsymbol{\phi}$ is a basis vector (the details of the mathematical expressions are in Section 3.3 and Section 4.3).

Ben-Shimon and Shmilovici (2006) partitioned training samples into small chunks to avoid the inverse of a large matrix (the matrix inversion is the most computationally expensive part of RVM). They suggested three methods to accelerate the computation of the basic algorithm by Tipping (2001). Firstly, relevance vectors (RVs) from two partitions are merged together into a new partition recursively (this is called the split and merge RVM). Secondly, RVs from the basic RVM and the previous partition are merged consecutively (this is called the incremental RVM). Thirdly, the incremental RVM is advanced: the merge operation is performed with the partition of the most informative basis functions.

Yang *et al.* (2010) accelerated RVM computation by parallelising the matrix inversion operation on a GPU. To be specific, the Cholesky decomposition for the matrix inversion is implemented on a GPU. RVM uses an expectation–maximization (EM) algorithm, an iterative method, to find maximum likelihood (this is explained in Section 3.4 and Section 4.4), and the Cholesky decomposition is performed iteratively based on the Cholesky decomposition results of the previous EM iteration.

2.3. *Single-output GP regression*

Seeger *et al.* (2003) reduced GP regression training time by approximating the likelihood of training data. The approximation consists of the following three components: i) likelihood approximation for an active set of training samples (the active set is a part of the whole training samples, and the reduced number of training samples decreases the computation time of the GP regression training), ii) how to select the active set (information gain is calculated to score a new point for inclusion to the active set), and iii) marginal likelihood approximation for the active set.

Shen *et al.* (2006) reduced both the training and prediction time of GP regression by using an approximation method, based on a tree-type multiresolution data structure. They assumed that the kernel of GP regression is monotonic isotropic (e.g. the Gaussian radial basis function kernel) and grouped data points together that have similar weights. For the grouping, a k-d tree, a binary tree that recursively partitions a set of data points, was employed.

Srinivasan *et al.* (2010) accelerated linear algebra operations of GP regression on a GPU, and Gramacy *et al.* (2014) made a GPU-accelerated version of GP regression approximation. Srinivasan *et al.* i) implemented the weighted summation of kernel functions by considering GPU memory architecture, ii) accelerated iterative algorithms by having data between iterations stay on a GPU, and iii) constructed kernel matrices in parallel on a GPU. Gramacy *et al.* converted a large problem of GP regression into many small independent ones for a cascade implementation on a GPU.

3. Existing method

The following subsections describe the existing method of MRVR (Thayananthan 2005, Chapter 6), (Thayananthan *et al.* 2008).

3.1. *Model specification*

V-dimensional multi-output regression upon an input $\mathbf{x} \in \mathbb{R}^{U\times 1}$ (U-dimensional column vector), a weight $\mathbf{W} \in \mathbb{R}^{M\times V}$ (M-by-V matrix), and an output function $\mathbf{y}(\mathbf{x};\mathbf{W}) \in \mathbb{R}^{1\times V}$ (V-dimensional row vector) (upright bold lower case letters denote vectors, and upright bold capital letters denote matrices) is expressed as

$$\mathbf{y}(\mathbf{x};\mathbf{W}) = \left(\mathbf{W}^\intercal \boldsymbol{\phi}(\mathbf{x})\right)^\intercal, \tag{3.1}$$

where $\mathbf{W}^\intercal$ is the transpose of the matrix $\mathbf{W}$ (the objective of this chapter is to estimate proper values of $\mathbf{W}$), and $\boldsymbol{\phi}(\mathbf{x}) = [\phi_1(\mathbf{x})\ \phi_2(\mathbf{x})\ \ldots\ \phi_M(\mathbf{x})]^\intercal \in \mathbb{R}^{M\times 1}$ is the transformed input by nonlinear and fixed basis functions. In other words, the output $\mathbf{y}(\mathbf{x};\mathbf{W})$ is a linearly weighted sum of the transformed input $\boldsymbol{\phi}(\mathbf{x})$.

Given a data set of input-target pairs $\left\{\mathbf{x}_i \in \mathbb{R}^{U\times 1}, \mathbf{t}_i \in \mathbb{R}^{1\times V}\right\}_{i=1}^{N}$, where N is the number of training samples, it is assumed that the targets $\mathbf{t}_i$ are samples from the model $\mathbf{y}(\mathbf{x}_i;\mathbf{W})$ with additive noise:

$$\mathbf{t}_i = \mathbf{y}(\mathbf{x}_i;\mathbf{W}) + \boldsymbol{\epsilon}_i, \tag{3.2}$$

where $\mathbf{W} \in \mathbb{R}^{(N+1)\times V}$ is the weight, $\boldsymbol{\epsilon}_i \in \mathbb{R}^{1\times V}$ are independent samples from a Gaussian noise process with mean zero and a covariance matrix $\boldsymbol{\Omega} \in \mathbb{R}^{V\times V}$, and $\boldsymbol{\Omega}$ is decomposed as the diagonal matrix of the variances $\mathbf{D} \in \mathbb{R}^{V\times V}$ and the correlation matrix $\mathbf{R} \in \mathbb{R}^{V\times V}$:

$$\boldsymbol{\Omega} = \mathbf{D}^{\frac{1}{2}}\mathbf{R}\mathbf{D}^{\frac{1}{2}}, \tag{3.3}$$

where $\mathbf{D} = \mathrm{diag}\left(\sigma_1^2, \sigma_2^2, \ldots, \sigma_V^2\right)$, and σ_j^2 is the variance of the j-th dimension's noise.

Because of the ignorance of $\mathbf{R}$ by Thayananthan (2005) and the assumption of independent Gaussian noise, the likelihood of the data set can be given by the product of the Gaussian distributions:

$$p\left(\mathbf{T}|\mathbf{W}, \boldsymbol{\sigma}\right) = \prod_{j=1}^{V} \left(2\pi\sigma_j^2\right)^{-\frac{N}{2}} \exp\left(-\frac{1}{2\sigma_j^2}\left\|\boldsymbol{\tau}_j - \boldsymbol{\Phi}\mathbf{w}_j\right\|^2\right), \tag{3.4}$$

where $\mathbf{T} = \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \vdots \\ \mathbf{t}_N \end{bmatrix} \in \mathbb{R}^{N\times V}$, $\boldsymbol{\sigma} = [\sigma_1\ \sigma_2\ \ldots\ \sigma_V] \in \mathbb{R}_{\geq 0}^{1\times V}$, $\boldsymbol{\tau}_j \in \mathbb{R}^{N\times 1}$ is the j-th column vector of $\mathbf{T}$, $\mathbf{w}_j \in \mathbb{R}^{(N+1)\times 1}$ is the j-th column vector of $\mathbf{W}$, $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1)\ \boldsymbol{\phi}(\mathbf{x}_2)\ \ldots\ \boldsymbol{\phi}(\mathbf{x}_N)]^\intercal \in \mathbb{R}^{N\times(N+1)}$ is a design matrix, $\boldsymbol{\phi}(\mathbf{x}) = [1\ K(\mathbf{x}, \mathbf{x}_1)\ \ldots\ K(\mathbf{x}, \mathbf{x}_N)]^\intercal \in \mathbb{R}^{(N+1)\times 1}$, and $K(\mathbf{x}, \mathbf{x}')$ is a kernel function. For clarity, the implicit conditioning on the input $\mathbf{x}_i, \forall i$ is omitted in Eq. (3.4) and the subsequent expressions.

An assumption to avoid over-fitting in estimation of $\mathbf{W}$ is

$$p\left(\mathbf{W}|\boldsymbol{\alpha}\right) = \prod_{j=1}^{V}\prod_{i=0}^{N} \mathcal{N}\left(0, \alpha_i^{-1}\right). \tag{3.5}$$

This means the prior distribution of $\mathbf{w}_j$ is zero-mean Gaussian with inverse variances $\boldsymbol{\alpha} = [\alpha_0\ \alpha_1\ \ldots\ \alpha_N]^\intercal \in \mathbb{R}_{>0}^{(N+1)\times 1}$, which are $N+1$ hyperparameters (Tipping 2001), and $\mathbf{w}_j$ and $\mathbf{w}_{j'}$ $(j \neq j')$ have the same distribution as $\prod_{i=0}^{N} \mathcal{N}\left(0, \alpha_i^{-1}\right)$.

3.2. *Inference*

By both the Bayes' theorem and the property of $p\left(\mathbf{T}|\mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\sigma}\right) = p\left(\mathbf{T}|\mathbf{W}, \boldsymbol{\sigma}\right)$,[1] the posterior probability distribution function over $\mathbf{W}$ is decomposed as

$$p\left(\mathbf{W}|\mathbf{T}, \boldsymbol{\alpha}, \boldsymbol{\sigma}\right) = \frac{p\left(\mathbf{T}|\mathbf{W}, \boldsymbol{\sigma}\right) p\left(\mathbf{W}|\boldsymbol{\alpha}, \boldsymbol{\sigma}\right)}{p\left(\mathbf{T}|\boldsymbol{\alpha}, \boldsymbol{\sigma}\right)}, \tag{3.6}$$

and it is given by the product of multivariate Gaussian distributions:

$$p\left(\mathbf{W}|\mathbf{T}, \boldsymbol{\alpha}, \boldsymbol{\sigma}\right) = \prod_{j=1}^{V} (2\pi)^{-\frac{N+1}{2}} \left|\boldsymbol{\Sigma}_j\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\mathbf{w}_j - \boldsymbol{\mu}_j\right)^\intercal \boldsymbol{\Sigma}_j^{-1}\left(\mathbf{w}_j - \boldsymbol{\mu}_j\right)\right), \tag{3.7}$$

where the j-th posterior covariance and mean are, respectively:

$$\boldsymbol{\Sigma}_j = \left(\sigma_j^{-2}\boldsymbol{\Phi}^\intercal\boldsymbol{\Phi} + \mathbf{A}\right)^{-1}, \tag{3.8}$$

$$\boldsymbol{\mu}_j = \sigma_j^{-2}\boldsymbol{\Sigma}_j\boldsymbol{\Phi}^\intercal\boldsymbol{\tau}_j, \tag{3.9}$$

[1] In the case that the weight $\mathbf{W}$ is given, its inverse variances $\boldsymbol{\alpha}$ are redundant in the calculation of the conditional probability of the target $\mathbf{T}$.

where $\mathbf{A} = \mathrm{diag}\left(\alpha_0, \alpha_1, \ldots, \alpha_N\right) \in \mathbb{R}^{(N+1)\times(N+1)}$.

In the case of uniform hyperpriors $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma}$, maximising a posteriori $p\left(\boldsymbol{\alpha}, \boldsymbol{\sigma} | \mathbf{T}\right) \propto p\left(\mathbf{T} | \boldsymbol{\alpha}, \boldsymbol{\sigma}\right) p\left(\boldsymbol{\alpha}\right) p\left(\boldsymbol{\sigma}\right)$ is equivalent to maximising the marginal likelihood $p\left(\mathbf{T} | \boldsymbol{\alpha}, \boldsymbol{\sigma}\right)$, which is given by

$$p\left(\mathbf{T} | \boldsymbol{\alpha}, \boldsymbol{\sigma}\right) = \prod_{j=1}^{V} \left(2\pi\right)^{-\frac{N}{2}} \left|\sigma_j^2 \mathbf{I} + \boldsymbol{\Phi} \mathbf{A}^{-1} \boldsymbol{\Phi}^\intercal\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \boldsymbol{\tau}_j^\intercal \left(\sigma_j^2 \mathbf{I} + \boldsymbol{\Phi} \mathbf{A}^{-1} \boldsymbol{\Phi}^\intercal\right)^{-1} \boldsymbol{\tau}_j\right). \tag{3.10}$$

3.3. *Marginal likelihood maximisation*

The same method of accelerating the univariate relevance vector machine (Tipping and Faul 2003) is used to accelerate the existing algorithm.

The log of Eq. (3.10) is an objective function:

$$\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\sigma}) = -\frac{1}{2} \sum_{j=1}^{V} \left(N \log(2\pi) + \log|\mathbf{C}_j| + \boldsymbol{\tau}_j^\intercal \mathbf{C}_j^{-1} \boldsymbol{\tau}_j\right), \tag{3.11}$$

where $\mathbf{C}_j = \sigma_j^2 \mathbf{I} + \boldsymbol{\Phi} \mathbf{A}^{-1} \boldsymbol{\Phi}^\intercal \in \mathbb{R}^{N \times N}$, and by considering the dependence of $\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ on a single hyperparameter $\alpha_i, i \in \{0, 1, \ldots, N\}$, $\mathbf{C}_j$ is decomposed as the following two parts:

$$\begin{aligned} \mathbf{C}_j &= \sigma_j^2 \mathbf{I} + \sum_{m \neq i} \alpha_m^{-1} \boldsymbol{\phi}_m \boldsymbol{\phi}_m^\intercal + \alpha_i^{-1} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\intercal \\ &= \mathbf{C}_{-i,j} + \alpha_i^{-1} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\intercal, \end{aligned} \tag{3.12}$$

where $\mathbf{C}_{-i,j} \in \mathbb{R}^{N \times N}$ is $\mathbf{C}_j$ with the contribution of a basis vector $\boldsymbol{\phi}_i \in \mathbb{R}^{N \times 1}$ removed, and

$$\boldsymbol{\phi}_i = \begin{cases} [1 \ 1 \ \ldots \ 1]^\intercal, & \text{if } i = 0 \\ [K(\mathbf{x}_i, \mathbf{x}_1) \ K(\mathbf{x}_i, \mathbf{x}_2) \ \ldots \ K(\mathbf{x}_i, \mathbf{x}_N)]^\intercal, & \text{otherwise} \end{cases}. \tag{3.13}$$

The determinant and inverse matrix of $\mathbf{C}_j$ are, respectively:

$$|\mathbf{C}_j| = |\mathbf{C}_{-i,j}| \left(1 + \alpha_i^{-1} \boldsymbol{\phi}_i^\intercal \mathbf{C}_{-i,j}^{-1} \boldsymbol{\phi}_i\right), \tag{3.14}$$

by Sylvester's determinant theorem, and

$$\mathbf{C}_j^{-1} = \mathbf{C}_{-i,j}^{-1} - \frac{\mathbf{C}_{-i,j}^{-1} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\intercal \mathbf{C}_{-i,j}^{-1}}{\alpha_i + \boldsymbol{\phi}_i^\intercal \mathbf{C}_{-i,j}^{-1} \boldsymbol{\phi}_i}, \tag{3.15}$$

by Woodbury matrix identity. From these, $\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ in Eq. (3.11) can be decomposed into $\mathcal{L}(\boldsymbol{\alpha}_{-i}, \boldsymbol{\sigma})$, the marginal likelihood with $\boldsymbol{\phi}_i$ excluded, and $\ell(\alpha_i, \boldsymbol{\sigma})$, the isolated marginal likelihood of $\boldsymbol{\phi}_i$:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\sigma}) = & -\frac{1}{2}\sum_{j=1}^{V}\left(N\log(2\pi) + \log|\mathbf{C}_{-i,j}| + \boldsymbol{\tau}_j^\intercal \mathbf{C}_{-i,j}^{-1}\boldsymbol{\tau}_j\right) \\
& -\frac{1}{2}\sum_{j=1}^{V}\left(-\log\alpha_i + \log\left(\alpha_i + \boldsymbol{\phi}_i^\intercal \mathbf{C}_{-i,j}^{-1}\boldsymbol{\phi}_i\right) - \frac{\left(\boldsymbol{\phi}_i^\intercal \mathbf{C}_{-i,j}^{-1}\boldsymbol{\tau}_j\right)^2}{\alpha_i + \boldsymbol{\phi}_i^\intercal \mathbf{C}_{-i,j}^{-1}\boldsymbol{\phi}_i}\right) \\
= & \mathcal{L}(\boldsymbol{\alpha}_{-i}, \boldsymbol{\sigma}) + \frac{1}{2}\sum_{j=1}^{V}\left(\log\alpha_i - \log\left(\alpha_i + s_{i,j}\right) + \frac{q_{i,j}^2}{\alpha_i + s_{i,j}}\right) \\
= & \mathcal{L}(\boldsymbol{\alpha}_{-i}, \boldsymbol{\sigma}) + \ell(\alpha_i, \boldsymbol{\sigma}),
\end{aligned} \tag{3.16}$$

where $s_{i,j}$ and $q_{i,j}$ are defined as, respectively:

$$s_{i,j} \stackrel{\text{def}}{=} \boldsymbol{\phi}_i^\intercal \mathbf{C}_{-i,j}^{-1}\boldsymbol{\phi}_i, \tag{3.17a}$$

$$q_{i,j} \stackrel{\text{def}}{=} \boldsymbol{\phi}_i^\intercal \mathbf{C}_{-i,j}^{-1}\boldsymbol{\tau}_j. \tag{3.17b}$$

To avoid the matrix inversion of $\mathbf{C}_{-i}$ in Eq. (3.17), which requires the time complexity of $O\left(N^3\right)$, $s'_{i,j}$ and $q'_{i,j}$ are computed as, respectively (by the Woodbury matrix identity):[1]

$$\begin{aligned}
s'_{i,j} &= \boldsymbol{\phi}_i^\intercal \mathbf{C}_j^{-1}\boldsymbol{\phi}_i \\
&= \sigma_j^{-2}\boldsymbol{\phi}_i^\intercal\boldsymbol{\phi}_i - \sigma_j^{-4}\boldsymbol{\phi}_i^\intercal\boldsymbol{\Phi}\boldsymbol{\Sigma}_j\boldsymbol{\Phi}^\intercal\boldsymbol{\phi}_i,
\end{aligned} \tag{3.18a}$$

$$\begin{aligned}
q'_{i,j} &= \boldsymbol{\phi}_i^\intercal \mathbf{C}_j^{-1}\boldsymbol{\tau}_j \\
&= \sigma_j^{-2}\boldsymbol{\phi}_i^\intercal\boldsymbol{\tau}_j - \sigma_j^{-4}\boldsymbol{\phi}_i^\intercal\boldsymbol{\Phi}\boldsymbol{\Sigma}_j\boldsymbol{\Phi}^\intercal\boldsymbol{\tau}_j,
\end{aligned} \tag{3.18b}$$

and then $s_{i,j}$ and $q_{i,j}$ in Eq. (3.17) are computed as, respectively:[2]

$$s_{i,j} = \frac{\alpha_i s'_{i,j}}{\alpha_i - s'_{i,j}}, \tag{3.19a}$$

$$q_{i,j} = \frac{\alpha_i q'_{i,j}}{\alpha_i - s'_{i,j}}. \tag{3.19b}$$

$\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ has a unique maximum with respect to α_i when the following equation is satisfied:

$$\frac{\partial\ell(\alpha_i, \boldsymbol{\sigma})}{\partial\alpha_i} = \frac{1}{2}\sum_{j=1}^{V}\left(\frac{1}{\alpha_i} - \frac{1}{\alpha_i + s_{i,j}} - \frac{q_{i,j}^2}{(\alpha_i + s_{i,j})^2}\right) = 0, \tag{3.20}$$

[1] $s'_{i,j} = \sigma_j^{-2}\boldsymbol{\phi}_i^\intercal\boldsymbol{\phi}_i$ and $q'_{i,j} = \sigma_j^{-2}\boldsymbol{\phi}_i^\intercal\boldsymbol{\tau}_j$ when $\alpha_i = \infty, \forall i$.

[2] $s_{i,j} = s'_{i,j}$ and $q_{i,j} = q'_{i,j}$ when $\alpha_i = \infty$.

which is a $(2V-1)$-th order polynomial equation of α_i. This implies that:

- If $\boldsymbol{\phi}_i$ is "in the model" (i.e. $\alpha_i < \infty$) and α_i in Eq. (3.20) has at least one positive real root ($\alpha_i > 0$ as α_i is inverse variance); then, α_i is re-estimated,
- If $\boldsymbol{\phi}_i$ is "in the model" (i.e. $\alpha_i < \infty$) yet α_i in Eq. (3.20) does not have any positive real root; then, $\boldsymbol{\phi}_i$ may be deleted (i.e. α_i is set to be ∞),
- If $\boldsymbol{\phi}_i$ is "out of the model" (i.e. $\alpha_i = \infty$) yet α_i in Eq. (3.20) has at least one positive real root; then, $\boldsymbol{\phi}_i$ may be added (i.e. α_i is set to be a finite value).

In addition, $\dfrac{\partial \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\sigma})}{\partial \sigma_j^2} = 0$ leads to that $\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ has a unique maximum with respect to σ_j^2 when:

$$\sigma_j^2 = \frac{\|\boldsymbol{\tau}_j - \boldsymbol{\Phi}\boldsymbol{\mu}_j\|^2}{N - \sum_{i=1}^{N+1} \gamma_{i,j}}, \tag{3.21}$$

where $\gamma_{i,j} \overset{\text{def}}{=} 1 - \alpha_{(i-1)}\Sigma_{j,ii}$, and $\Sigma_{j,ii}$ is the i-th diagonal element of $\boldsymbol{\Sigma}_j \in \mathbb{R}^{(N+1)\times(N+1)}$.

3.4. *Expectation–maximisation (EM) algorithm*

Algorithm 1, an EM algorithm to maximise the marginal likelihood, starts without any basis vector (i.e. $M = 0$) and selects the basis vector $\boldsymbol{\phi}_i$ which gives the maximum change of the marginal likelihood $\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ of Eq. (3.11) at every iteration.

For efficient computation of the EM algorithm, quantities $\boldsymbol{\Phi} \in \mathbb{R}^{N\times M}$, $\boldsymbol{\Sigma}_j \in \mathbb{R}^{M\times M}$, and $\boldsymbol{\mu}_j \in \mathbb{R}^{M\times 1}$ contain only M ($M \le N+1$ is always satisfied) basis functions that are currently included in the model (i.e. $\boldsymbol{\phi}_i$ satisfying $\alpha_i < \infty$), and the diagonal matrix $\mathbf{A}$ consists of M hyper-parameters of α_i that are currently included in the model (i.e. α_i satisfying $\alpha_i < \infty$). Additionally, Eq. (3.21) is rewritten as

$$\sigma_j^2 = \frac{\|\boldsymbol{\tau}_j - \boldsymbol{\Phi}\boldsymbol{\mu}_j\|^2}{N - \sum_{i=1}^{M} \gamma'_{i,j}}, \tag{3.22}$$

where $\gamma'_{i,j} \overset{\text{def}}{=} 1 - \alpha'_i\Sigma_{j,ii}$, α'_i is the i-th non-infinity value of $\boldsymbol{\alpha}$, and $\Sigma_{j,ii}$ is the i-th diagonal element of $\boldsymbol{\Sigma}_j \in \mathbb{R}^{M\times M}$.

From Eq. (3.11), the change in the marginal likelihood can be written as

$$\begin{aligned} 2\Delta\mathcal{L} &= 2\left(\mathcal{L}(\tilde{\boldsymbol{\alpha}}, \boldsymbol{\sigma}) - \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\sigma})\right) \\ &= \sum_{j=1}^{V} \left(\log \frac{|\mathbf{C}_j|}{|\tilde{\mathbf{C}}_j|} + \boldsymbol{\tau}_j^{\intercal} \left(\mathbf{C}_j^{-1} - \tilde{\mathbf{C}}_j^{-1} \right) \boldsymbol{\tau}_j \right), \end{aligned} \tag{3.23}$$

where updated quantities are denoted by a tilde (e.g., $\tilde{\boldsymbol{\alpha}}$ and $\tilde{\mathbf{C}}_j$). Eq. (3.23) differs according to whether α_i is re-estimated, added, or deleted:

Re-estimation. as $\mathbf{C}_j = \mathbf{C}_{-i,j} + \alpha_i^{-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^{\intercal}$ and $\tilde{\mathbf{C}} = \mathbf{C}_{-i,j} + \tilde{\alpha}_i^{-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^{\intercal}$,

$$2\Delta\mathcal{L}_i = \sum_{j=1}^{V} \left(\frac{q'^2_{i,j}}{s'_{i,j} + \left(\tilde{\alpha}_i^{-1} - \alpha_i^{-1}\right)^{-1}} - \log\left(1 + s'_{i,j}\left(\tilde{\alpha}_i^{-1} - \alpha_i^{-1}\right)\right) \right), \tag{3.24}$$

where $\tilde{\alpha}_i$ is re-estimated α_i,

Algorithm 1: Existing EM algorithm of MRVR.

Input: $\mathbf{T} \in \mathbb{R}^{N \times V}$, and $\boldsymbol{\phi}_i \in \mathbb{R}^{N \times 1}, \forall i = \{0, 1, \ldots, N\}$, where N is the number of training samples, and V is the number of output dimensions
Output: $\boldsymbol{\Sigma}_j \in \mathbb{R}^{M \times M}, \boldsymbol{\mu}_j \in \mathbb{R}^{M \times 1}$, and $\sigma_j, \forall j = \{1, 2, \ldots, V\}$, where M is the number of basis functions in the model

```
// Initialisation
α_i ← ∞, ∀i = {0, 1, ..., N}
for j ← 1 to V do
    t̄_j ← (1/N) Σ_{i=1}^{N} t_{i,j}
    σ_j^2 ← (0.1/(N−1)) Σ_{i=1}^{N} (t_{i,j} − t̄_j)^2
end
convergence←false, n ← 1, M ← 0, where n is the iteration number, and M is the number of basis functions.
while convergence=false do
    // maximisation step
    for i ← 0 to N do
        for j ← 1 to V do
            Update s'_{i,j} and q'_{i,j} using Eq. (3.18), and Update s_{i,j} and q_{i,j} using Eq. (3.19).
        end
        switch the number of positive real roots of Eq. (3.20) do
            case 0 do
                α̃_i ← ∞
            case 1 do
                α̃_i ← the positive real root of Eq. (3.20)
            otherwise do
                α̃_i ← one of the positive real roots of Eq. (3.20), which maximises 2ΔL_i of either i) Eq. (3.24) if α_i < ∞
                  or ii) Eq. (3.25) if α_i = ∞
            end
        end
        if α̃_i < ∞ then
            if α_i < ∞ then z_i ←'re-estimation'
                Update 2ΔL_i using Eq. (3.24).
            else z_i ←'addition'
                Update 2ΔL_i using Eq. (3.25).
            end
        else if α_i < ∞ then z_i ←'deletion'
            Update 2ΔL_i using Eq. (3.26).
        else
            2ΔL_i ← −∞
        end
    end
    i ← arg max_i 2ΔL_i // Select i which gives the greatest increase of the marginal likelihood
    if n ≠ 1 then
        Update σ_j, ∀j using Eq. (3.22).
    end
    switch z_i do
        case 're-estimation' do
            Δ log α ← log (α_i / α̃_i)
            α_i ← α̃_i
            // Check convergence (convergence criteria are the same as those in (Tipping and Faul 2003))
            if |Δ log α| < 0.1 then
                convergence←true
                for i ← 0 to N do
                    if α_i = ∞ then // if φ_i is "out of the model"
                        if α̃_i < ∞ then // if φ_i may be added
                            convergence←false
                            break for loop
                        end
                    end
                end
            end
        case 'addition' do
            α_i ← α̃_i, M ← M + 1
        case 'deletion' do
            α_i ← ∞, M ← M − 1
    end
    // Expectation step
    Sequentially update i) Φ ∈ R^{N×M}, A ∈ R^{M×M}, ii) Σ_j ∈ R^{M×M}, ∀j, and iii) μ_j ∈ R^M, ∀j using Eq. (3.8) and
      Eq. (3.9), where Φ, Σ_j, and μ_j contain only M basis functions that are currently included in the model, and the
      diagonal matrix A consists of M hyperparameters of α_i that are currently included in the model.
    n ← n + 1
end
```

Addition. as $\mathbf{C}_j = \mathbf{C}_{-i,j}$ and $\tilde{\mathbf{C}}_j = \mathbf{C}_{-i,j} + \tilde{\alpha}_i^{-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^\intercal$,

$$2\Delta\mathcal{L}_i = \sum_{j=1}^{V}\left(\frac{q_{i,j}^2}{\tilde{\alpha}_i + s_{i,j}} + \log\frac{\tilde{\alpha}_i}{\tilde{\alpha}_i + s_{i,j}}\right), \tag{3.25}$$

Deletion. as $\mathbf{C}_j = \mathbf{C}_{-i,j} + \alpha_i^{-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^\intercal$ and $\tilde{\mathbf{C}}_j = \mathbf{C}_{-i,j}$,

$$2\Delta\mathcal{L}_i = \sum_{j=1}^{V}\left(\frac{q_{i,j}'^2}{s_{i,j}' - \alpha_i} - \log\left(1 - \frac{s_{i,j}'}{\alpha_i}\right)\right). \tag{3.26}$$

3.5. *Making predictions*

We can predict both the mean of j-th output dimension $y_{*,j}$ and its variance $\sigma_{*,j}^2$ from a new input vector $\mathbf{x}_*$ based on both i) Eq. (3.2), the model specification, and ii) Eq. (3.7), the posterior distribution over the weights, conditioned on the most probable (MP) hyperparameters: $\boldsymbol{\alpha}_{\mathrm{MP}} \in \mathbb{R}_{>0}^{M\times 1}$ and $\boldsymbol{\sigma}_{\mathrm{MP}} \in \mathbb{R}_{\geq 0}^{1\times V}$, obtained from Algorithm 1. Predictive distribution of $t_{*,j}$ is normally distributed as

$$p(t_{*,j}|\mathbf{T}, \boldsymbol{\alpha}_{\mathrm{MP}}, \boldsymbol{\sigma}_{\mathrm{MP}}) = \mathcal{N}\left(t_{*,j}|y_{*,j}, \sigma_{*,j}^2\right), \tag{3.27}$$

with

$$y_{*,j} = \boldsymbol{\phi}(\mathbf{x}_*)^\intercal\boldsymbol{\mu}_j, \tag{3.28}$$

and

$$\sigma_{*,j}^2 = \sigma_{\mathrm{MP},j}^2 + \boldsymbol{\phi}(\mathbf{x}_*)^\intercal\boldsymbol{\Sigma}_j\boldsymbol{\phi}(\mathbf{x}_*), \tag{3.29}$$

where $\boldsymbol{\phi}(\mathbf{x}_*) \in \mathbb{R}^{M\times 1}$ comes from only M basis functions that are included in the model after the EM algorithm, and subscript j refers to the j-th output dimension. The predictive variance $\sigma_{*,j}^2$ comprises the sum of two variance components: the estimated noise on the training data $\sigma_{\mathrm{MP},j}^2$ and that due to the uncertainty in the prediction of the weights $\boldsymbol{\phi}(\mathbf{x}_*)^\intercal\boldsymbol{\Sigma}_j\boldsymbol{\phi}(\mathbf{x}_*)$.

3.6. *Algorithm complexity*

Matrix inversion of $\boldsymbol{\Sigma}_j \in \mathbb{R}^{M\times M}$ in Eq. (3.8) for all $j \in \{1, 2, \ldots, V\}$ determines i) the time complexity of the existing algorithm as $O\left(VM^3\right)$ and ii) the memory complexity as $O\left(VM^2\right)$, where V is the number of output dimensions, and M is the number of basis functions.[1]

[1] The matrix multiplication to calculate $s_{i,j}'$ and $q_{i,j}'$ in Eq. (3.18) for all $i \in \{1, 2, \ldots, N\}$, $j \in \{1, 2, \ldots, V\}$ at the 11-th line of Algorithm 1 has the same time complexity because the matrix multiplication $\boldsymbol{\Phi}\boldsymbol{\Sigma}_j\boldsymbol{\Phi}^\intercal$ is pre-calculated. In other words, the time complexity of the matrix multiplication is $O\left(VM^3\right)$, not $O\left(NVM^3\right)$, because $\boldsymbol{\Phi}\boldsymbol{\Sigma}_j\boldsymbol{\Phi}^\intercal$ is independent of i.

4. Proposed method

4.1. *Model specification*

Given a data set of input-target pairs $\left\{\mathbf{x}_i \in \mathbb{R}^{U\times 1}, \mathbf{t}_i \in \mathbb{R}^{1\times V}\right\}_{i=1}^{N}$, where N is the number of training samples, it is assumed that the targets $\mathbf{t}_i$ are samples from the model $\mathbf{y}(\mathbf{x}_i;\mathbf{W})$ with additive noise:

$$\mathbf{t}_i = \mathbf{y}(\mathbf{x}_i;\mathbf{W}) + \boldsymbol{\epsilon}_i, \tag{4.1}$$

where $\mathbf{W} \in \mathbb{R}^{(N+1)\times V}$ is the weight and $\boldsymbol{\epsilon}_i \in \mathbb{R}^{1\times V}$ are independent samples from a Gaussian noise process with mean zero and a covariance matrix $\boldsymbol{\Omega} \in \mathbb{R}^{V\times V}$.

Eq. (4.1) can be rewritten, using matrix algebra, as

$$\mathbf{T} = \boldsymbol{\Phi}\mathbf{W} + \mathbf{E}, \tag{4.2}$$

where $\mathbf{T} = \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\\ \vdots\\ \mathbf{t}_N\end{bmatrix} \in \mathbb{R}^{N\times V}$ is the target, $\mathbf{E} = \begin{bmatrix}\boldsymbol{\epsilon}_1\\ \boldsymbol{\epsilon}_2\\ \vdots\\ \boldsymbol{\epsilon}_N\end{bmatrix} \in \mathbb{R}^{N\times V}$ is the noise, $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1)\ \boldsymbol{\phi}(\mathbf{x}_2)\ \ldots\ \boldsymbol{\phi}(\mathbf{x}_N)]^\intercal \in \mathbb{R}^{N\times(N+1)}$ is a design matrix, $\boldsymbol{\phi}(\mathbf{x}) = [1\ K(\mathbf{x},\mathbf{x}_1)\ K(\mathbf{x},\mathbf{x}_2)\ \ldots\ K(\mathbf{x},\mathbf{x}_N)]^\intercal \in \mathbb{R}^{(N+1)\times 1}$, and $K(\mathbf{x},\mathbf{x}')$ is a kernel function.

Because of the assumption of independent Gaussian noise, the likelihood of the data set can be given by the matrix Gaussian distribution:

$$p\left(\mathbf{T}|\mathbf{W},\boldsymbol{\Omega}\right) = (2\pi)^{-\frac{VN}{2}} \left|\boldsymbol{\Omega}\right|^{-\frac{N}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\left(\mathbf{T}-\boldsymbol{\Phi}\mathbf{W}\right)^\intercal\left(\mathbf{T}-\boldsymbol{\Phi}\mathbf{W}\right)\right)\right), \tag{4.3}$$

where $\boldsymbol{\Omega} = \frac{\mathbb{E}[\mathbf{E}^\intercal\mathbf{E}]}{N}$, and tr denotes trace.[1] As I assumed, $\mathbf{I} = \frac{\mathbb{E}[\mathbf{E}\mathbf{E}^\intercal]}{\mathrm{tr}(\boldsymbol{\Omega})}$, which means the noise is independent among rows with the same variance, where $\mathbf{I}$ is an $N\times N$ identity matrix. For clarity, the implicit conditioning on the input $\mathbf{x}_i, \forall i$ is omitted in Eq. (4.3) and the subsequent expressions.

An assumption to avoid over-fitting in the estimation of $\mathbf{W}$ is

$$p\left(\mathbf{W}|\boldsymbol{\alpha},\boldsymbol{\Omega}\right) = (2\pi)^{-\frac{V(N+1)}{2}} \left|\boldsymbol{\Omega}\right|^{-\frac{N+1}{2}} \left|\mathbf{A}\right|^{\frac{V}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{W}^\intercal\mathbf{A}\mathbf{W}\right)\right), \tag{4.4}$$

where $\mathbf{A}^{-1} = \mathrm{diag}\left(\alpha_0^{-1},\alpha_1^{-1},\ldots,\alpha_N^{-1}\right) = \frac{\mathbb{E}[\mathbf{W}\mathbf{W}^\intercal]}{\mathrm{tr}(\boldsymbol{\Omega})}$. This means the prior distribution of $\mathbf{W}$ is zero-mean Gaussian with among-row inverse variances $\boldsymbol{\alpha} = [\alpha_0\ \alpha_1\ \ldots\ \alpha_N]^\intercal \in \mathbb{R}_{>0}^{(N+1)\times 1}$, which are $N+1$ hyperparameters (Tipping 2001). Eq. (4.4) implies another assumption: $\boldsymbol{\Omega} = \frac{\mathbb{E}[\mathbf{W}^\intercal\mathbf{W}]}{\mathrm{tr}(\mathbf{A}^{-1})}$. Actually, this is unreasonable because the weight $\mathbf{W}$ has no relationship with the noise $\mathbf{E}$ (i.e. $\mathbf{I} = \frac{\mathbb{E}[\mathbf{W}^\intercal\mathbf{W}]}{\mathrm{tr}(\mathbf{A}^{-1})}$, which means that the weights of different output dimensions are not correlated, is a reasonable assumption), but it is essential for creating a computationally efficient algorithm.

[1] If $\boldsymbol{\Omega} = \mathrm{diag}\left(\sigma_1^2,\sigma_2^2,\ldots,\sigma_V^2\right)$, then Eq. (4.3) will be Eq. (3.4).

4.2. *Inference*

By both the Bayes' theorem and the property of $p(\mathbf{T}|\mathbf{W},\boldsymbol{\alpha},\boldsymbol{\Omega}) = p(\mathbf{T}|\mathbf{W},\boldsymbol{\Omega})$,[1] the posterior probability distribution function over $\mathbf{W}$ is decomposed as

$$p(\mathbf{W}|\mathbf{T},\boldsymbol{\alpha},\boldsymbol{\Omega}) = \frac{p(\mathbf{T}|\mathbf{W},\boldsymbol{\Omega})\, p(\mathbf{W}|\boldsymbol{\alpha},\boldsymbol{\Omega})}{p(\mathbf{T}|\boldsymbol{\alpha},\boldsymbol{\Omega})}, \tag{4.5}$$

and it is given by the matrix Gaussian distribution:[2]

$$p(\mathbf{W}|\mathbf{T},\boldsymbol{\alpha},\boldsymbol{\Omega}) = (2\pi)^{-\frac{V(N+1)}{2}} |\boldsymbol{\Omega}|^{-\frac{N+1}{2}} |\boldsymbol{\Sigma}|^{-\frac{V}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}(\mathbf{W}-\mathbf{M})^\intercal \boldsymbol{\Sigma}^{-1}(\mathbf{W}-\mathbf{M})\right)\right), \tag{4.6}$$

where the posterior covariance and mean are, respectively:

$$\boldsymbol{\Sigma} = \left(\boldsymbol{\Phi}^\intercal\boldsymbol{\Phi} + \mathbf{A}\right)^{-1}, \tag{4.7}$$

$$\mathbf{M} = \boldsymbol{\Sigma}\boldsymbol{\Phi}^\intercal\mathbf{T}. \tag{4.8}$$

In the case of uniform hyperpriors $\boldsymbol{\alpha}$ and $\boldsymbol{\Omega}$, maximising a posteriori $p(\boldsymbol{\alpha},\boldsymbol{\Omega}|\,\mathbf{T}) \propto p(\mathbf{T}|\boldsymbol{\alpha},\boldsymbol{\Omega})\, p(\boldsymbol{\alpha})\, p(\boldsymbol{\Omega})$ is equivalent to maximising the marginal likelihood $p(\mathbf{T}|\boldsymbol{\alpha},\boldsymbol{\Omega})$, which is given by:

$$p(\mathbf{T}|\boldsymbol{\alpha},\boldsymbol{\Omega}) = (2\pi)^{-\frac{VN}{2}} |\boldsymbol{\Omega}|^{-\frac{N}{2}} \left|\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^\intercal\right|^{-\frac{V}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{T}^\intercal\left(\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^\intercal\right)^{-1}\mathbf{T}\right)\right). \tag{4.9}$$

4.3. *Marginal likelihood maximisation*

The same method of accelerating the univariate relevance vector machine (Tipping and Faul 2003) is used to accelerate the proposed algorithm.

The log of Eq. (4.9) is an objective function:

$$\mathcal{L}(\boldsymbol{\alpha},\boldsymbol{\Omega}) = -\frac{1}{2}\left(VN\log(2\pi) + N\log|\boldsymbol{\Omega}| + V\log|\mathbf{C}| + \mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{T}^\intercal\mathbf{C}^{-1}\mathbf{T}\right)\right), \tag{4.10}$$

where $\mathbf{C} = \mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^\intercal \in \mathbb{R}^{N\times N}$, and by considering the dependence of $\mathcal{L}(\boldsymbol{\alpha},\boldsymbol{\Omega})$ on a single hyperparameter $\alpha_i, i \in \{0,1,\ldots,N\}$, $\mathbf{C}$ is decomposed as the following two parts:

$$\begin{aligned}\mathbf{C} &= \mathbf{I} + \sum_{m\neq i}\alpha_m^{-1}\boldsymbol{\phi}_m\boldsymbol{\phi}_m^\intercal + \alpha_i^{-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^\intercal \\ &= \mathbf{C}_{-i} + \alpha_i^{-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^\intercal,\end{aligned} \tag{4.11}$$

where $\mathbf{C}_{-i} \in \mathbb{R}^{N\times N}$ is $\mathbf{C}$ with the contribution of a basis vector $\boldsymbol{\phi}_i \in \mathbb{R}^{N\times 1}$ removed, and

$$\boldsymbol{\phi}_i = \begin{cases} [1\ 1\ \ldots\ 1]^\intercal, & \text{if } i = 0 \\ [K(\mathbf{x}_i,\mathbf{x}_1)\ K(\mathbf{x}_i,\mathbf{x}_2)\ \ldots\ K(\mathbf{x}_i,\mathbf{x}_N)]^\intercal, & \text{otherwise} \end{cases}. \tag{4.12}$$

---

[1] In the case that the weight $\mathbf{W}$ is given, its inverse variances $\boldsymbol{\alpha}$ are redundant in the calculation of the conditional probability of the target $\mathbf{T}$.

[2] The proof is in Appendix.

The determinant and inverse matrix of $\mathbf{C}$ are, respectively:

$$|\mathbf{C}| = |\mathbf{C}_{-i}| \left(1 + \alpha_i^{-1} \boldsymbol{\phi}_i^\intercal \mathbf{C}_{-i}^{-1} \boldsymbol{\phi}_i\right), \tag{4.13}$$

by Sylvester's determinant theorem, and

$$\mathbf{C}^{-1} = \mathbf{C}_{-i}^{-1} - \frac{\mathbf{C}_{-i}^{-1} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\intercal \mathbf{C}_{-i}^{-1}}{\alpha_i + \boldsymbol{\phi}_i^\intercal \mathbf{C}_{-i}^{-1} \boldsymbol{\phi}_i}, \tag{4.14}$$

by Woodbury matrix identity. From these, $\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\Omega})$ in Eq. (4.10) can be decomposed into $\mathcal{L}(\boldsymbol{\alpha}_{-i}, \boldsymbol{\Omega})$, the marginal likelihood with $\boldsymbol{\phi}_i$ excluded, and $\ell(\alpha_i, \boldsymbol{\Omega})$, the isolated marginal likelihood of $\boldsymbol{\phi}_i$:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\Omega}) = &-\frac{1}{2}\left(VN\log(2\pi) + N\log|\boldsymbol{\Omega}| + V\log|\mathbf{C}_{-i}| + \operatorname{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{T}^\intercal\mathbf{C}_{-i}^{-1}\mathbf{T}\right)\right) \\
&-\frac{1}{2}\left(-V\log\alpha_i + V\log\left(\alpha_i + \boldsymbol{\phi}_i^\intercal\mathbf{C}_{-i}^{-1}\boldsymbol{\phi}_i\right) - \operatorname{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{T}^\intercal\frac{\mathbf{C}_{-i}^{-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^\intercal\mathbf{C}_{-i}^{-1}}{\alpha_i + \boldsymbol{\phi}_i^\intercal\mathbf{C}_{-i}^{-1}\boldsymbol{\phi}_i}\mathbf{T}\right)\right) \\
=&\mathcal{L}(\boldsymbol{\alpha}_{-i}, \boldsymbol{\Omega}) + \frac{1}{2}\left(V\log\alpha_i - V\log\left(\alpha_i + s_i\right) + \frac{\operatorname{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{q}_i^\intercal\mathbf{q}_i\right)}{\alpha_i + s_i}\right) \\
=&\mathcal{L}(\boldsymbol{\alpha}_{-i}, \boldsymbol{\Omega}) + \ell(\alpha_i, \boldsymbol{\Omega}),
\end{aligned} \tag{4.15}$$

where s_i and $\mathbf{q}_i \in \mathbb{R}^{1\times V}$ are defined as, respectively:

$$s_i \stackrel{\text{def}}{=} \boldsymbol{\phi}_i^\intercal \mathbf{C}_{-i}^{-1} \boldsymbol{\phi}_i, \tag{4.16a}$$

$$\mathbf{q}_i \stackrel{\text{def}}{=} \boldsymbol{\phi}_i^\intercal \mathbf{C}_{-i}^{-1} \mathbf{T}. \tag{4.16b}$$

To avoid the matrix inversion of $\mathbf{C}_{-i}$ in Eq. (4.16), which requires the time complexity of $O\left(N^3\right)$, s_i' and $\mathbf{q}_i' \in \mathbb{R}^{1\times V}$ are computed as (by the Woodbury matrix identity):[1]

$$\begin{aligned} s_i' &= \boldsymbol{\phi}_i^\intercal \mathbf{C}^{-1} \boldsymbol{\phi}_i \\ &= \boldsymbol{\phi}_i^\intercal \boldsymbol{\phi}_i - \boldsymbol{\phi}_i^\intercal \boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^\intercal \boldsymbol{\phi}_i, \end{aligned} \tag{4.17a}$$

$$\begin{aligned} \mathbf{q}_i' &= \boldsymbol{\phi}_i^\intercal \mathbf{C}^{-1} \mathbf{T} \\ &= \boldsymbol{\phi}_i^\intercal \mathbf{T} - \boldsymbol{\phi}_i^\intercal \boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^\intercal \mathbf{T}, \end{aligned} \tag{4.17b}$$

and then s_i and $\mathbf{q}_i$ in Eq. (4.16) are computed as:[2]

$$s_i = \frac{\alpha_i s_i'}{\alpha_i - s_i'}, \tag{4.18a}$$

$$\mathbf{q}_i = \frac{\alpha_i \mathbf{q}_i'}{\alpha_i - s_i'}. \tag{4.18b}$$

---

[1] $s_i' = \boldsymbol{\phi}_i^\intercal \boldsymbol{\phi}_i$ and $\mathbf{q}_i' = \boldsymbol{\phi}_i^\intercal \mathbf{T}$ when $\alpha_i = \infty, \forall i$.
[2] $s_i = s_i'$ and $\mathbf{q}_i = \mathbf{q}_i'$ when $\alpha_i = \infty$.

$\dfrac{\partial \ell(\alpha_i, \boldsymbol{\Omega})}{\partial \alpha_i} = 0$ leads to that $\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\Omega})$ has a unique maximum with respect to α_i when:

$$\alpha_i = \begin{cases} \dfrac{s_i^2}{\dfrac{\operatorname{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{q}_i^\intercal \mathbf{q}_i\right)}{V} - s_i}, & \text{if } \dfrac{\operatorname{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{q}_i^\intercal \mathbf{q}_i\right)}{V} > s_i \\ \infty, & \text{if } \dfrac{\operatorname{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{q}_i^\intercal \mathbf{q}_i\right)}{V} \leq s_i \end{cases}, \tag{4.19}$$

which implies that:

- If $\boldsymbol{\phi}_i$ is "in the model" (i.e. $\alpha_i < \infty$) and $\dfrac{\operatorname{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{q}_i^\intercal \mathbf{q}_i\right)}{V} > s_i$; then, α_i is re-estimated,
- If $\boldsymbol{\phi}_i$ is "in the model" (i.e. $\alpha_i < \infty$) yet $\dfrac{\operatorname{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{q}_i^\intercal \mathbf{q}_i\right)}{V} \leq s_i$; then, $\boldsymbol{\phi}_i$ may be deleted (i.e. α_i is set to be ∞),
- If $\boldsymbol{\phi}_i$ is "out of the model" (i.e. $\alpha_i = \infty$) yet $\dfrac{\operatorname{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{q}_i^\intercal \mathbf{q}_i\right)}{V} > s_i$; then, $\boldsymbol{\phi}_i$ may be added (i.e. α_i is set to be a finite value).

In addition, $\dfrac{\partial \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\Omega})}{\partial \boldsymbol{\Omega}} = \mathbf{0}$, where $\mathbf{0}$ is a $V \times V$ zero matrix, leads to that $\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\Omega})$ has a unique maximum with respect to $\boldsymbol{\Omega}$ when:

$$\boldsymbol{\Omega} = \frac{\mathbf{T}^\intercal(\mathbf{T} - \boldsymbol{\Phi}\mathbf{M})}{N}. \tag{4.20}$$

4.4. *Expectation–maximisation (EM) algorithm*

Algorithm 2, an EM algorithm to maximise the marginal likelihood, starts without any basis vector (i.e. $M = 0$) and selects the basis vector $\boldsymbol{\phi}_i$ which gives the maximum change of the marginal likelihood $\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\Omega})$ of Eq. (4.10) at every iteration.

For efficient computation of the EM algorithm, quantities $\boldsymbol{\Phi} \in \mathbb{R}^{N \times M}$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{M \times M}$ contain only M ($M \leq N + 1$ is always satisfied) basis functions that are currently included in the model (i.e. $\boldsymbol{\phi}_i$ which satisfies $\alpha_i < \infty$), and the diagonal matrix $\mathbf{A}$ consists of M hyperparameters of α_i that are currently included in the model (i.e. α_i which satisfies $\alpha_i < \infty$).

From Eq. (4.10), the change in the marginal likelihood can be written as

$$\begin{aligned} 2\Delta\mathcal{L} &= 2\left(\mathcal{L}(\tilde{\boldsymbol{\alpha}}, \boldsymbol{\Omega}) - \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\Omega})\right) \\ &= V \log \frac{|\mathbf{C}|}{\left|\tilde{\mathbf{C}}\right|} + \operatorname{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{T}^\intercal\left(\mathbf{C}^{-1} - \tilde{\mathbf{C}}^{-1}\right)\mathbf{T}\right), \end{aligned} \tag{4.21}$$

where updated quantities are denoted by a tilde (e.g., $\tilde{\boldsymbol{\alpha}}$ and $\tilde{\mathbf{C}}$). Eq. (4.21) differs according to whether α_i is re-estimated, added, or deleted:

Re-estimation. as $\mathbf{C} = \mathbf{C}_{-i} + \alpha_i^{-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^\intercal$ and $\tilde{\mathbf{C}} = \mathbf{C}_{-i} + \tilde{\alpha}_i^{-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^\intercal$,

$$2\Delta\mathcal{L}_i = \frac{\operatorname{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{q}_i'^\intercal \mathbf{q}_i'\right)}{s_i' + \left(\tilde{\alpha}_i^{-1} - \alpha_i^{-1}\right)^{-1}} - V\log\left(1 + s_i'\left(\tilde{\alpha}_i^{-1} - \alpha_i^{-1}\right)\right), \tag{4.22}$$

where $\tilde{\alpha}_i$ is re-estimated α_i,

Algorithm 2: Proposed EM algorithm of MRVR.

Input: $\mathbf{T} \in \mathbb{R}^{N \times V}$ and $\boldsymbol{\phi}_i \in \mathbb{R}^{N \times 1}, \forall i = \{0, 1, \ldots, N\}$, where N is the number of training samples, and V is the number of output dimensions

Output: $\boldsymbol{\Sigma} \in \mathbb{R}^{M \times M}$, $\mathbf{M} \in \mathbb{R}^{M \times V}$, and $\boldsymbol{\Omega} \in \mathbb{R}^{V \times V}$, where M is the number of basis functions in the model

// Initilisation

$\alpha_i \leftarrow \infty, \forall i = \{0, 1, \ldots, N\}$

$\bar{\mathbf{t}} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \mathbf{t}_i$, where $\bar{\mathbf{t}} \in \mathbb{R}^{1 \times V}$, and $\mathbf{t}_i \in \mathbb{R}^{1 \times V}$ is the i-th row vector of $\mathbf{T}$.

$\boldsymbol{\Omega} \leftarrow \frac{0.1}{N-1} \sum_{i=1}^{N} (\mathbf{t}_i - \bar{\mathbf{t}})^\intercal (\mathbf{t}_i - \bar{\mathbf{t}})$, where $\boldsymbol{\Omega} \in \mathbb{R}^{V \times V}$ is a covariance matrix.

convergence←false

$n \leftarrow 1$, where n is the iteration number

$M \leftarrow 0$, where M is the number of basis functions.

while *convergence=false* **do**

  // maximisation step

  for $i \leftarrow 0$ *to* N **do**

    Update s_i' and $\mathbf{q}_i'$ using Eq. (4.17).

    Update s_i and $\mathbf{q}_i$ using Eq. (4.18).

    $\theta_i \leftarrow \frac{\mathrm{tr}(\boldsymbol{\Omega}^{-1} \mathbf{q}_i^\intercal \mathbf{q}_i)}{V} - s_i$

    if $\theta_i > 0$ **then**

      if $\alpha_i < \infty$ **then** $z_i \leftarrow$‘re-estimation’

        $\tilde{\alpha}_i \leftarrow \frac{s_i^2}{\theta_i}$

        Update $2\Delta\mathcal{L}_i$ using Eq. (4.22).

      else $z_i \leftarrow$‘addition’

        Update $2\Delta\mathcal{L}_i$ using Eq. (4.23).

      end

    else if $\alpha_i < \infty$ **then** $z_i \leftarrow$‘deletion’

      Update $2\Delta\mathcal{L}_i$ using Eq. (4.24).

    else

      $2\Delta\mathcal{L}_i \leftarrow -\infty$

    end

  end

  $i \leftarrow \arg\max_i 2\Delta\mathcal{L}_i$ // Select i which gives the greatest increase of the marginal likelihood

  switch z_i **do**

    case *‘re-estimation’* **do**

      $\Delta \log \alpha \leftarrow \log \frac{\alpha_i}{\tilde{\alpha}_i}$

      $\alpha_i \leftarrow \tilde{\alpha}_i$

      // Check convergence (convergence criteria are the same as those in (Tipping and Faul 2003))

      if $|\Delta \log \alpha| < 0.1$ **then**

        convergence←true

        for $i \leftarrow 0$ *to* N **do**

          if $\alpha_i = \infty$ **then** // if $\boldsymbol{\phi}_i$ is "out of the model"

            if $\theta_i > 0$ **then** // if $\boldsymbol{\phi}_i$ may be added

              convergence←false

              break for loop

            end

          end

        end

      end

    case *‘addition’* **do**

      $\alpha_i \leftarrow \frac{s_i^2}{\theta_i}$

      $M \leftarrow M + 1$

    case *‘deletion’* **do**

      $\alpha_i \leftarrow \infty$

      $M \leftarrow M - 1$

  end

  if $n \neq 1$ **then**

    Update $\boldsymbol{\Omega}$ using Eq. (4.20).

  end

  // Expectation step

  Sequentially update i) $\boldsymbol{\Phi} \in \mathbb{R}^{N \times M}$, $\mathbf{A} \in \mathbb{R}^{M \times M}$, ii) $\boldsymbol{\Sigma} \in \mathbb{R}^{M \times M}$, and iii) $\mathbf{M} \in \mathbb{R}^{M \times V}$ using Eq. (4.7) and Eq. (4.8), where $\boldsymbol{\Phi}$, $\boldsymbol{\Sigma}$, and $\mathbf{M}$ contain only M basis functions that are currently included in the model, and the diagonal matrix $\mathbf{A}$ consists of M hyperparameters of α_i that are currently included in the model.

  $n \leftarrow n + 1$

end

Addition. as $\mathbf{C} = \mathbf{C}_{-i}$ and $\tilde{\mathbf{C}} = \mathbf{C}_{-i} + \tilde{\alpha}_i^{-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^\intercal$, where $\tilde{\alpha}_i = \dfrac{s_i^2}{\dfrac{\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{q}_i^\intercal\mathbf{q}_i\right)}{V} - s_i}$,

$$2\Delta\mathcal{L}_i = \frac{\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{q}_i'^\intercal\mathbf{q}_i'\right) - Vs_i'}{s_i'} + V\log\frac{Vs_i'}{\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{q}_i'^\intercal\mathbf{q}_i'\right)}, \tag{4.23}$$

Deletion. as $\mathbf{C} = \mathbf{C}_{-i} + \alpha_i^{-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^\intercal$ and $\tilde{\mathbf{C}} = \mathbf{C}_{-i}$,

$$2\Delta\mathcal{L}_i = \frac{\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{q}_i'^\intercal\mathbf{q}_i'\right)}{s_i' - \alpha_i} - V\log\left(1 - \frac{s_i'}{\alpha_i}\right). \tag{4.24}$$

4.5. *Making predictions*

We can predict both a mean vector $\mathbf{y}_* \in \mathbb{R}^{1\times V}$ and a covariance matrix $\boldsymbol{\Omega}_* \in \mathbb{R}^{V\times V}$ from a new input vector $\mathbf{x}_* \in \mathbb{R}^{U\times 1}$ based on both i) Eq. (4.2), the model specification, and ii) Eq. (4.6), the posterior distribution over the weights, conditioned on the most probable (MP) hyperparameters: $\boldsymbol{\alpha}_{\mathrm{MP}} \in \mathbb{R}_{>0}^{M\times 1}$ and $\boldsymbol{\Omega}_{\mathrm{MP}} \in \mathbb{R}^{V\times V}$, obtained from Algorithm 2. Predictive distribution of $\mathbf{t}_*$ is jointly normally distributed as

$$p(\mathbf{t}_*|\mathbf{T}, \boldsymbol{\alpha}_{\mathrm{MP}}, \boldsymbol{\Omega}_{\mathrm{MP}}) = \mathcal{N}(\mathbf{t}_*|\mathbf{y}_*, \boldsymbol{\Omega}_*), \tag{4.25}$$

with

$$\mathbf{y}_* = \boldsymbol{\phi}(\mathbf{x}_*)^\intercal\mathbf{M}, \tag{4.26}$$

and

$$\boldsymbol{\Omega}_* = \boldsymbol{\Omega}_{\mathrm{MP}}\left(1 + \boldsymbol{\phi}(\mathbf{x}_*)^\intercal\boldsymbol{\Sigma}\boldsymbol{\phi}(\mathbf{x}_*)\right), \tag{4.27}$$ [1]

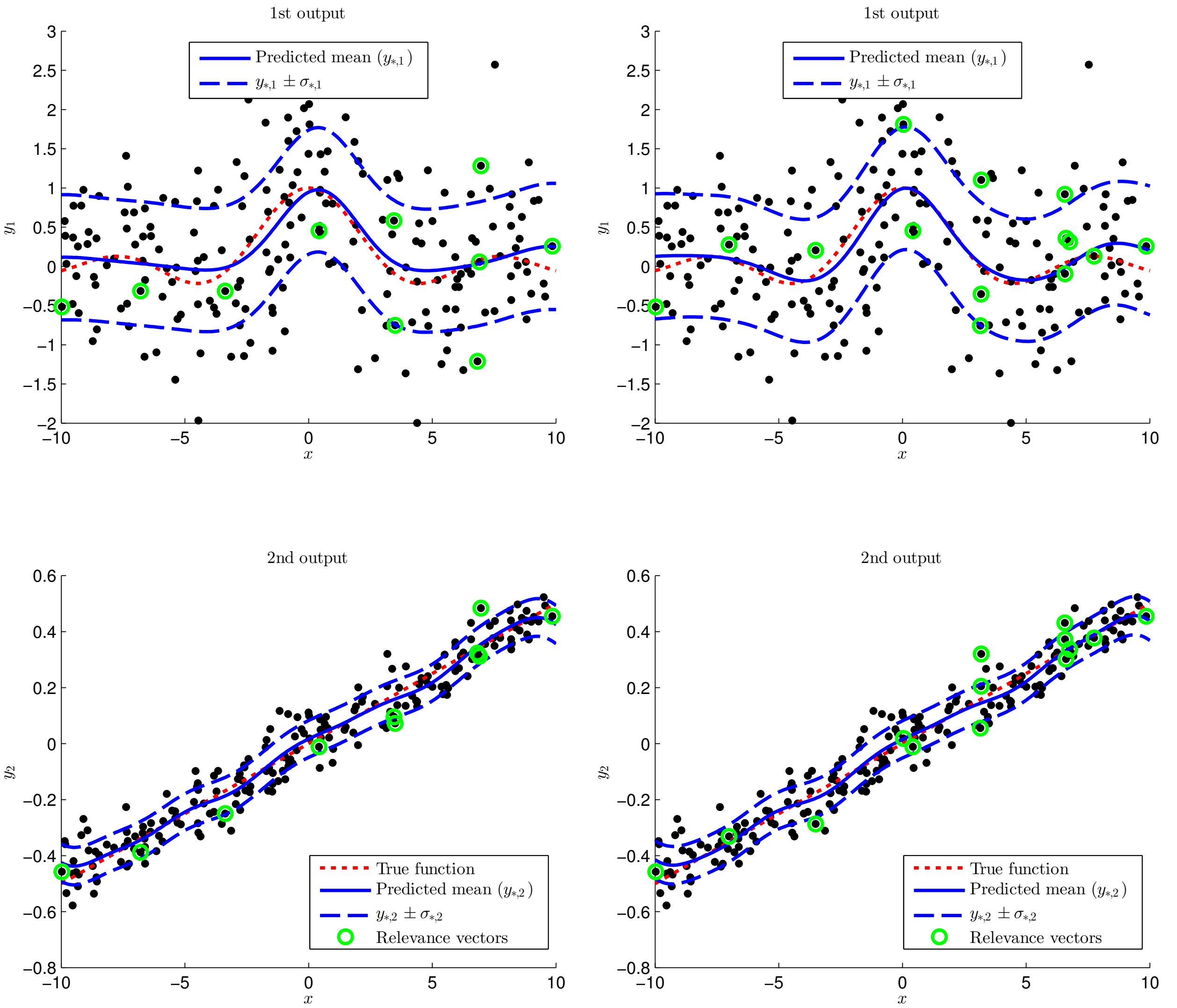
where $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^{M\times 1}$ comes from only M basis functions that are included in the model after the EM algorithm. The predictive covariance matrix $\boldsymbol{\Omega}_*$ comprises the two components: the estimated noise on the training data $\boldsymbol{\Omega}_{\mathrm{MP}}$ and that due to the uncertainty in the prediction of the weights $\boldsymbol{\Omega}_{\mathrm{MP}}\boldsymbol{\phi}(\mathbf{x}_*)^\intercal\boldsymbol{\Sigma}\boldsymbol{\phi}(\mathbf{x}_*)$, where $\boldsymbol{\phi}(\mathbf{x}_*)^\intercal\boldsymbol{\Sigma}\boldsymbol{\phi}(\mathbf{x}_*) \in \mathbb{R}_{\geq 0}$ by the fact that a covariance matrix is always positive semidefinite. They share $\boldsymbol{\Omega}_{\mathrm{MP}}$ by the assumption of $\boldsymbol{\Omega} = \frac{\mathbb{E}[\mathbf{E}^\intercal\mathbf{E}]}{N} = \frac{\mathbb{E}[\mathbf{W}^\intercal\mathbf{W}]}{\mathrm{tr}(\mathbf{A}^{-1})}$ in Section 4.1.

4.6. *Algorithm complexity*

Matrix inversion of $\boldsymbol{\Omega} \in \mathbb{R}^{V\times V}$ in Eq. (4.19) and that of the $M \times M$ matrix in Eq. (4.7) determine i) the time complexity of the proposed algorithm as $O\left(V^3 + M^3\right)$ and ii) the memory complexity as $O\left(V^2 + M^2\right)$, where V is the number of output dimensions, and M is the number of basis functions.[2]

[1] Eq. (4.27) is obtained by the property that the covariance between two elements $W_{i,j}$ and $W_{i',j'}$ is the covariance between the rows i and i', i.e. $\boldsymbol{\Sigma}$, multiplied by the covariance between the columns j and j', i.e. $\boldsymbol{\Omega}_{\mathrm{MP}}$ (Arnold 1981, p. 311).

[2] The matrix multiplication to calculate s_i' and $\mathbf{q}_i'$ in Eq. (4.17) for all $i \in \{1, 2, \ldots, N\}$ at the 9-th line of Algorithm 2 does not influence the time complexity because the matrix multiplication $\boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^\intercal$ is pre-calculated. In other words, the time complexity of the matrix multiplication is $O\left(M^3\right)$, not $O\left(NM^3\right)$, because $\boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^\intercal$ is independent of i.



(a) Existing method ($\sigma_{*,j}^2$ is equal to Eq. (3.29))

(b) Proposed method ($\sigma_{*,j}^2$ is equal to the j-th diagonal element of $\mathbf{\Omega}_*$ in Eq. (4.27))

Figure 1.: An example of MRVR (when $U = 1, V = 2, N = 200$, and the Gaussian kernel $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\lambda^2}\right)$ with a free parameter $\lambda = 1.6$ is used)

5. Experimental results

5.1. *An example of MRVR*

Fig. 1 shows an example of the MRVR results obtained using the two methods when the true function of each output dimension is the sinc function and the linear function, respectively. Fig. 1a and Fig. 1b show slightly different results although the same training samples are used.

5.2. *Comparisons of the performance*

The two methods are compared in terms of i) running time (computation time in INTEL® Core™ i5-3470 CPU and MATLAB® R2013b), ii) the estimation accuracy of the noise covariance matrix, iii) root-mean-square error (RMSE) between true functions and predicted mean values, and iv) the number of relevance vectors (RVs), where RVs are those training vectors associated with the remaining non-zero weights (i.e. the number of basis functions M is equal to the number of RVs).

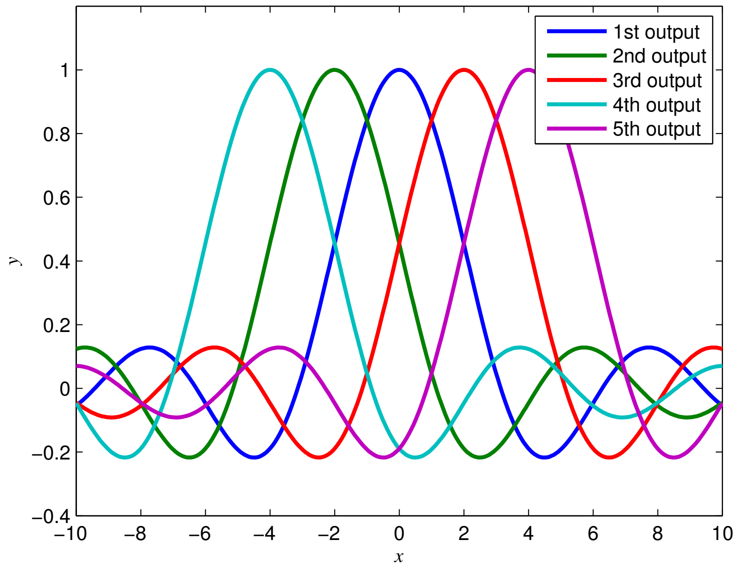


Figure 2.: True functions of MC simulations

To measure the estimation accuracy of the noise covariance matrix $\mathbf{\Omega}$, entropy loss $L_1\left(\mathbf{\Omega}, \hat{\mathbf{\Omega}}\right)$ and quadratic loss $L_2\left(\mathbf{\Omega}, \hat{\mathbf{\Omega}}\right)$ are used (each of these is 0 when $\hat{\mathbf{\Omega}} = \mathbf{\Omega}$ and is positive when $\hat{\mathbf{\Omega}} \neq \mathbf{\Omega}$) (Anderson 1984, pp. 273–274):

$$L_1\left(\mathbf{\Omega}, \hat{\mathbf{\Omega}}\right) = \mathrm{tr}\left(\hat{\mathbf{\Omega}}\mathbf{\Omega}^{-1}\right) - \log\left|\hat{\mathbf{\Omega}}\mathbf{\Omega}^{-1}\right| - V, \tag{5.1}$$

$$L_2\left(\mathbf{\Omega}, \hat{\mathbf{\Omega}}\right) = \mathrm{tr}\left(\left(\hat{\mathbf{\Omega}}\mathbf{\Omega}^{-1} - \mathbf{I}\right)^2\right), \tag{5.2}$$

where the estimated $V \times V$ covariance matrix of the noise $\hat{\mathbf{\Omega}}$ is $\mathbf{\Omega}_{\mathrm{MP}}$ in the case of the proposed method. In the case of the existing method, $\hat{\mathbf{\Omega}}$ can be obtained using both i) the estimated standard deviation of the noise $\hat{\mathbf{D}} = \mathrm{diag}(\sigma_{\mathrm{MP},1}, \sigma_{\mathrm{MP},2}, \ldots, \sigma_{\mathrm{MP},V})$ in Section 3.5 and ii) the estimated correlation matrix of the noise $\hat{\mathbf{R}}$:

$$\hat{\mathbf{\Omega}} = \hat{\mathbf{D}}\hat{\mathbf{R}}\hat{\mathbf{D}}, \tag{5.3}$$

where $\hat{\mathbf{R}} = \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{\Omega}}\tilde{\mathbf{D}}^{-1}$, $\tilde{\mathbf{\Omega}} = \dfrac{(\mathbf{T} - \mathbf{\Phi}\tilde{\mathbf{M}})^{\intercal}(\mathbf{T} - \mathbf{\Phi}\tilde{\mathbf{M}})}{N-1}$, $\tilde{\mathbf{M}} = [\boldsymbol{\mu}_1 \ \boldsymbol{\mu}_2 \ \ldots \ \boldsymbol{\mu}_V] \in \mathbb{R}^{M \times V}$, and $\tilde{\mathbf{D}} = \sqrt{\mathrm{diag}(\tilde{\mathbf{\Omega}})}$.

Monte Carlo (MC) simulations with random covariance matrices of the noise were conducted for the performance comparisons. The noise from the random covariance matrix is added to the true functions in Fig. 2 (each output has a sinc function with a translation in the x-axis), and the two methods of MRVR with the Gaussian kernel were performed using the same training samples for a fair comparison.

Unpaired two-sample t-tests may be used to compare the two methods to determine whether the performance difference is fundamental or whether it is due to random fluctuations (Simon 2013, pp. 631–635), but the normality assumption of the performance measures (i.e. running time, entropy loss, quadratic loss, RMSE, and the number of RVs) of the two methods must be checked. The Jarque–Bera tests $JB = \frac{n}{6}\left(S^2 + \frac{1}{4}(K-3)^2\right)$ with the number of observations $n = 101$ and

Table 1.: The number of rejections of the null hypothesis of the Jarque–Bera test

	Running time	Entropy loss	Quadratic loss	RMSE	The number of RVs
Existing method	30	30	30	4	16
Proposed method	30	28	29	3	14

Table 2.: The difference in median values of running time (seconds)

	$N = 50$	$N = 100$	$N = 150$	$N = 200$	$N = 250$	$N = 300$
$V = 1$	**0.04** (3.6×10^{-10})	**0.19** (2.1×10^{-9})	**0.40** (4.9×10^{-9})	**1.02** (3.2×10^{-9})	**3.11** (1.8×10^{-20})	**5.95** (7.1×10^{-18})
$V = 2$	**0.20** (4.8×10^{-25})	**0.76** (1.5×10^{-22})	**2.44** (9.0×10^{-26})	**5.95** (2.1×10^{-25})	**17.20** (1.3×10^{-30})	**26.17** (1.0×10^{-31})
$V = 3$	**0.33** (3.2×10^{-26})	**1.70** (3.8×10^{-32})	**5.27** (5.4×10^{-29})	**10.37** (1.1×10^{-31})	**37.74** (4.7×10^{-34})	**64.99** (2.2×10^{-34})
$V = 4$	**0.74** (2.2×10^{-33})	**3.60** (1.3×10^{-32})	**9.69** (2.1×10^{-33})	**24.51** (2.4×10^{-33})	**57.14** (2.1×10^{-34})	**112.18** (1.2×10^{-34})
$V = 5$	**1.05** (1.1×10^{-33})	**5.25** (5.4×10^{-34})	**13.01** (6.3×10^{-34})	**34.09** (2.2×10^{-34})	**92.96** (1.2×10^{-34})	**176.24** (1.2×10^{-34})

Table 3.: The difference in median values of entropy loss

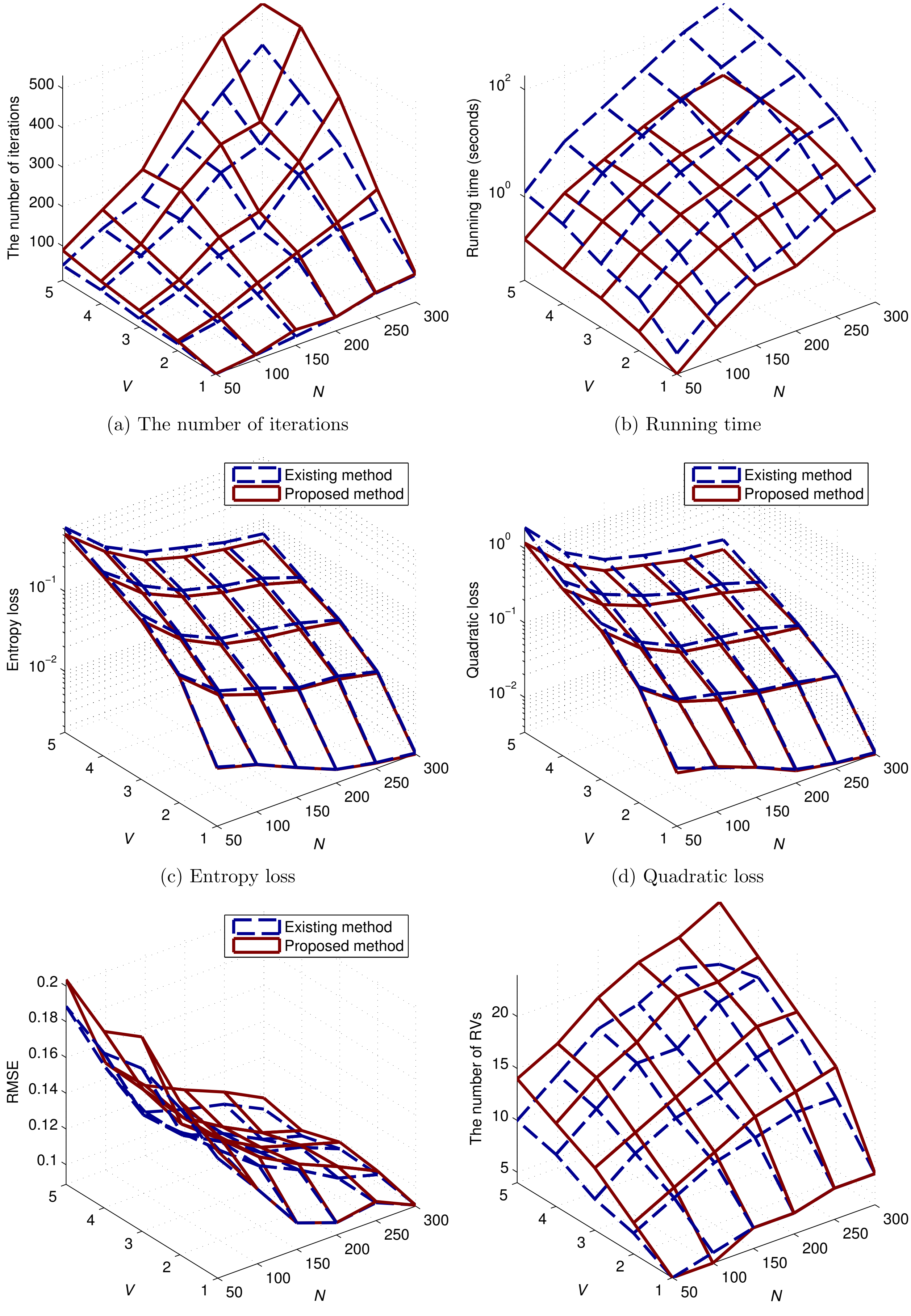
	$N = 50$	$N = 100$	$N = 150$	$N = 200$	$N = 250$	$N = 300$
$V = 1$	3.4×10^{-4} (0.9904)	8.0×10^{-5} (0.9176)	2.2×10^{-5} (0.9962)	1.7×10^{-5} (0.9520)	3.7×10^{-5} (0.8454)	5.7×10^{-5} (0.9808)
$V = 2$	3.9×10^{-3} (0.4942)	3.3×10^{-3} (0.5964)	3.4×10^{-3} (0.1132)	6.8×10^{-4} (0.3915)	1.1×10^{-3} (0.7434)	4.0×10^{-4} (0.6234)
$V = 3$	3.2×10^{-2} (0.0883)	8.6×10^{-3} (0.2469)	$\mathbf{6.8 \times 10^{-3}}$ (0.0192)	$\mathbf{8.0 \times 10^{-3}}$ (0.0069)	$\mathbf{5.1 \times 10^{-3}}$ (0.0119)	$\mathbf{1.4 \times 10^{-3}}$ (0.0412)
$V = 4$	3.9×10^{-2} (0.1137)	$\mathbf{3.0 \times 10^{-2}}$ (0.0030)	$\mathbf{1.5 \times 10^{-2}}$ (0.0010)	$\mathbf{1.1 \times 10^{-2}}$ (0.0042)	$\mathbf{9.3 \times 10^{-3}}$ (0.0063)	$\mathbf{4.4 \times 10^{-3}}$ (0.0058)
$V = 5$	$\mathbf{1.1 \times 10^{-1}}$ (0.0231)	$\mathbf{3.0 \times 10^{-2}}$ (0.0015)	$\mathbf{2.9 \times 10^{-2}}$ (0.0000)	$\mathbf{2.5 \times 10^{-2}}$ (0.0008)	$\mathbf{1.4 \times 10^{-2}}$ (0.0005)	$\mathbf{1.2 \times 10^{-2}}$ (0.0018)

a 5% significance level for 30 cases ($V = \{1, 2, 3, 4, 5\}$ and $N = \{50, 100, 150, 200, 250, 300\}$) were conducted, in which the null hypothesis was that the data of the performance measures came from a normal distribution. Table 1 shows the number of rejections of the null hypothesis. Consequently, the t-test can yield misleading results in the case that the null hypothesis is rejected.

Instead of the t-test, two-sided Wilcoxon rank sum tests, whose null hypothesis is that two populations have equal median values, were used for the comparisons as they have greater efficiency than the t-test on non-normal distributions and are nearly as efficient as the t-test on normal distributions (Montgomery 2013, Chapter 10).

Fig. 3 shows the median values of the performance measures of both methods with various V and N values. Entropy loss, quadratic loss, and RMSE decrease as N increases: the greater the number of training samples, the more accurate the estimation. In contrast, the number of RVs, the number of iterations of each EM algorithm (the same tolerance value of 0.1 for checking the convergence of each EM algorithm was used as in 43th line of Algorithm 1 and 30th line of Algorithm 2), and the running time (only for learning without prediction) of each EM algorithm increase as N increases: the greater the number of training samples, the greater the computational burden.

Tables 2–6 show i) the difference in the median values of the performance measures, where each median value is obtained from 101 MC simulations, and then the median value of the proposed method is subtracted from that of the existing method (i.e. positive difference values mean that the proposed method is better than the existing method, while negative difference values mean the



(a) The number of iterations

(b) Running time

(c) Entropy loss

(d) Quadratic loss

(e) RMSE

(f) The number of RVs

Figure 3.: Median values of MC simulations (the number of simulations is 101)

Table 4.: The difference in median values of quadratic loss

	$N=50$	$N=100$	$N=150$	$N=200$	$N=250$	$N=300$
$V=1$	2.8×10^{-3} (0.9981)	-5.7×10^{-4} (0.9233)	4.9×10^{-5} (0.9981)	2.3×10^{-4} (0.9405)	4.4×10^{-5} (0.8605)	2.0×10^{-4} (0.9770)
$V=2$	5.5×10^{-3} (0.5080)	3.5×10^{-3} (0.4526)	7.0×10^{-3} (0.0915)	2.2×10^{-3} (0.3616)	1.6×10^{-3} (0.6824)	-1.9×10^{-4} (0.5782)
$V=3$	$\mathbf{8.5 \times 10^{-2}}$ (0.0180)	2.5×10^{-2} (0.1154)	$\mathbf{1.4 \times 10^{-2}}$ (0.0109)	$\mathbf{1.4 \times 10^{-2}}$ (0.0029)	$\mathbf{1.3 \times 10^{-2}}$ (0.0051)	$\mathbf{2.7 \times 10^{-3}}$ (0.0263)
$V=4$	$\mathbf{1.7 \times 10^{-1}}$ (0.0382)	$\mathbf{8.6 \times 10^{-2}}$ (0.0005)	$\mathbf{5.4 \times 10^{-2}}$ (0.0001)	$\mathbf{2.4 \times 10^{-2}}$ (0.0007)	$\mathbf{2.9 \times 10^{-2}}$ (0.0019)	$\mathbf{1.5 \times 10^{-2}}$ (0.0013)
$V=5$	$\mathbf{7.3 \times 10^{-1}}$ (0.0058)	$\mathbf{1.8 \times 10^{-1}}$ (0.0000)	$\mathbf{8.5 \times 10^{-2}}$ (0.0000)	$\mathbf{5.5 \times 10^{-2}}$ (0.0001)	$\mathbf{4.8 \times 10^{-2}}$ (0.0000)	$\mathbf{3.7 \times 10^{-2}}$ (0.0004)

Table 5.: The difference in median values of RMSE

	$N=50$	$N=100$	$N=150$	$N=200$	$N=250$	$N=300$
$V=1$	-0.0038 (0.956)	-0.0000 (0.985)	0.0000 (0.967)	-0.0001 (1.000)	-0.0007 (0.965)	0.0001 (0.987)
$V=2$	-0.0035 (0.898)	-0.0076 (0.544)	-0.0051 (0.562)	-0.0033 (0.758)	-0.0064 (0.646)	-0.0020 (0.977)
$V=3$	-0.0177 (0.182)	-0.0049 (0.299)	-0.0012 (0.408)	-0.0030 (0.661)	-0.0013 (0.546)	-0.0037 (0.565)
$V=4$	**-0.0121** (0.041)	**-0.0142** (0.034)	**-0.0101** (0.017)	-0.0037 (0.092)	-0.0038 (0.115)	-0.0029 (0.293)
$V=5$	**-0.0150** (0.008)	**-0.0026** (0.013)	**-0.0149** (0.001)	**-0.0110** (0.015)	**-0.0068** (0.025)	-0.0062 (0.072)

Table 6.: The difference in median values of the number of RVs

	$N=50$	$N=100$	$N=150$	$N=200$	$N=250$	$N=300$
$V=1$	0 (8.8×10^{-1})	1 (9.0×10^{-1})	0 (8.4×10^{-1})	0 (9.6×10^{-1})	0 (8.9×10^{-1})	0 (9.9×10^{-1})
$V=2$	-1 (5.2×10^{-2})	**-1** (4.8×10^{-4})	**-1** (3.2×10^{-3})	**-2** (5.1×10^{-4})	**-2** (1.5×10^{-3})	**-3** (2.2×10^{-3})
$V=3$	**-3** (3.6×10^{-7})	**-2** (3.9×10^{-5})	**-2** (1.5×10^{-5})	**-3** (3.4×10^{-7})	**-3** (3.2×10^{-6})	**-2** (2.6×10^{-6})
$V=4$	**-3** (2.2×10^{-8})	**-3** (4.8×10^{-6})	**-2** (8.6×10^{-5})	**-5** (2.4×10^{-9})	**-2** (2.2×10^{-7})	**-2** (1.8×10^{-4})
$V=5$	**-4** (3.9×10^{-15})	**-3** (3.9×10^{-10})	**-3** (3.9×10^{-8})	**-4** (9.0×10^{-10})	**-3** (6.9×10^{-7})	**-6** (3.2×10^{-7})

opposite) and ii) the p-values of the Wilcoxon rank sum tests, which appear inside the brackets. The p-value is interpreted as the probability that a difference in the median values would be obtained given that the population medians of two methods are equivalent, i.e. the p-value is not equal to the probability that the population medians are equivalent (Simon 2013, p. 635). Note that statistically significant difference values are marked in bold (p-value < 0.05).

The proposed method is faster than the existing method as shown in Table 2 (all differences are statistically significant). In particular, the time difference is amplified as V or N increases. This is because the time complexity of the proposed method $O\left(V^3 + M^3\right)$ is less than that of the existing method $O\left(VM^3\right)$ ($O\left(V^3 + M^3\right) < O\left(VM^3\right)$ is satisfied since $V < M$ is satisfied in most applications). Note that even when the number of input dimensions U changes, the size of the design matrix $\mathbf{\Phi}$ does not change. Hence, U does not influence the time complexity of both the methods.

Furthermore, the proposed method achieves higher accuracy in estimating the covariance matrix of the noise $\mathbf{\Omega}$ than the existing method as shown in Table 3 and Table 4. This is because the proposed method considers the correlation matrix of the noise as Eq. (4.3), but the existing method does not as Eq. (4.3).

However, the proposed method is worse than the existing one in terms of i) the accuracy in predicting the mean values as shown in Table 5 (in particular, the RMSE increases in the region of high V and low N) and ii) the number of RVs as shown in Table 6. This is because the proposed method has the assumption of the weight $\mathbf{\Omega} = \frac{\mathbb{E}[\mathbf{W}^\intercal \mathbf{W}]}{\mathrm{tr}(\mathbf{A}^{-1})}$, which behaves as the constraint of the weight. Consequently, the mean values tend to deviate from the true functions, and the number of RVs increases.

The MATLAB codes of the experiment have been uploaded on `http://www.mathworks.com/matlabcentral/fileexchange/49131` to avoid any potential ambiguity of both the methods.

6. Conclusion

A new algorithm of MRVR has been proposed. It is more efficient in computing the weight $\mathbf{W}$ and more accurate in estimating the covariance matrix of the noise $\mathbf{\Omega}$ than the existing algorithm. Its computational efficiency and accuracy can be attributed to the different model specifications: the existing method expresses the likelihood of the training data as the product of the Gaussian distributions in Eq. (3.4), but the proposed one expresses it as the matrix Gaussian distribution in Eq. (4.3).

However, the proposed method has drawbacks of lower accuracy in estimating the mean of the weight $\mathbf{M}$ in Eq. (4.8) and higher number of RVs than the existing method. These disadvantages are caused by the assumption $\mathbf{\Omega} = \frac{\mathbb{E}[\mathbf{W}^\intercal \mathbf{W}]}{\mathrm{tr}(\mathbf{A}^{-1})}$, which means the weight $\mathbf{W}$ is related to the noise $\mathbf{E}$ in Eq. (4.2), but it was indispensable to make MRVR faster.

References

Alvarez, M. and Lawrence, N.D., Sparse convolved Gaussian processes for multi-output regression. In *Proceedings of the Advances in Neural Information Processing Systems*, pp. 57--64, 2009.

Anderson, T.W., *An introduction to multivariate statistical analysis* (2 edn), 1984, Wiley.

Arnold, S.F., *The theory of linear models and multivariate analysis*, 1981, Wiley.

Ben-Shimon, D. and Shmilovici, A., Accelerating the relevance vector machine via data partitioning. *Foundations of Computing and Decision Sciences*, 2006, **31**, 27--41.

Bonilla, E.V., Chai, K.M.A. and Williams, C.K.I., Multi-task Gaussian process prediction. In *Proceedings of the Advances in Neural Information Processing Systems*, pp. 153--160, 2007.

Boyle, P. and Frean, M.R., Dependent Gaussian Processes.. In *Proceedings of the Advances in Neural Information Processing Systems*, pp. 217--224, 2004.

Catanzaro, B., Sundaram, N. and Keutzer, K., Fast support vector machine training and classification on graphics processors. In *Proceedings of the 25th international conference on Machine learning*, pp. 104--111, 2008.

Chang, C.C. and Lin, C.J., LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011, **2**, 27:1--27.

Chu, W., Keerthi, S.S. and Ong, C.J., Bayesian support vector regression using a unified loss function. *IEEE Transactions on Neural Networks*, 2004, **15**, 29--44.

Cortes, C. and Vapnik, V., Support-vector networks. *Machine learning*, 1995, **20**, 273--297.

Gao, J.B., Gunn, S.R., Harris, C.J. and Brown, M., A probabilistic framework for SVM regression and error bar estimation. *Machine Learning*, 2002, **46**, 71--89.

Gibbs, M., Bayesian Gaussian processes for classification and regression. PhD thesis, University of Cambridge, 1997.

Gramacy, R.B., Niemi, J. and Weiss, R.M., Massively parallel approximate Gaussian process regression. *SIAM/ASA Journal on Uncertainty Quantification*, 2014, **2**, 564--584.

Guo, G. and Zhang, J.S., Reducing examples to accelerate support vector regression. *Pattern Recognition Letters*, 2007, **28**, 2173--2183.

Montgomery, D.C., *Applied Statistics and Probability for Engineers* (6 edn), 2013, Wiley.
Pérez-Cruz, F., Camps-Valls, G., Soria-Olivas, E., Pérez-Ruixo, J.J., Figueiras-Vidal, A.R. and Artés-Rodríguez, A., Multi-dimensional function approximation and regression estimation. In *Proceedings of the International Conference on Artificial Neural Networks*, pp. 757--762, 2002.
Schölkopf, B., Smola, A.J., Williamson, R.C. and Bartlett, P.L., New support vector algorithms. *Neural computation*, 2000, **12**, 1207--1245.
Seeger, M., Williams, C. and Lawrence, N., Fast forward selection to speed up sparse Gaussian process regression. In *Proceedings of the Workshop on Artificial Intelligence and Statistics 9*, 2003.
Shen, Y., Ng, A. and Seeger, M., Fast Gaussian process regression using kd-trees. In *Proceedings of the 19th Annual Conference on Neural Information Processing Systems*, 2006.
Simon, D., *Evolutionary optimization algorithms*, 2013, Wiley.
Srinivasan, B.V., Qi, H. and Duraiswami, R., GPUML: Graphical processors for speeding up kernel machines. In *Proceedings of the Workshop on High Performance Analytics - Algorithms, Implementations, and Applications*, 2010.
Thayananthan, A., Template-based pose estimation and tracking of 3D hand motion. PhD thesis, University of Cambridge, 2005.
Thayananthan, A., Navaratnam, R., Stenger, B., Torr, P.H. and Cipolla, R., Pose estimation and tracking using multivariate regression. *Pattern Recognition Letters*, 2008, **29**, 1302--1310.
Tipping, M.E., Sparse Bayesian learning and the relevance vector machine. *The journal of machine learning research*, 2001, **1**, 211--244.
Tipping, M.E. and Faul, A.C., Fast marginal likelihood maximisation for sparse Bayesian models. In *Proceedings of the ninth international workshop on artificial intelligence and statistics*, Vol. 1, 2003.
Tuia, D., Verrelst, J., Alonso, L., Pérez-Cruz, F. and Camps-Valls, G., Multioutput support vector regression for remote sensing biophysical parameter estimation. *IEEE Geoscience and Remote Sensing Letters*, 2011, **8**, 804--808.
Vapnik, V., *The nature of statistical learning theory*, 2000, Springer.
Vazquez, E. and Walter, E., Multi-output support vector regression. In *Proceedings of the 13th IFAC Symposium on System Identification*, pp. 1820--1825, 2003.
Yang, D., Liang, G., Jenkins, D.D., Peterson, G.D. and Li, H., High performance relevance vector machine on GPUs. In *Proceedings of the Symposium on Application Accelerators in High-Performance Computing*, 2010.

Appendix: proof of Eq. (4.6) and Eq. (4.9)

$$
\begin{aligned}
& p\left(\mathbf{W}|\mathbf{T},\boldsymbol{\alpha},\boldsymbol{\Omega}\right) p\left(\mathbf{T}|\boldsymbol{\alpha},\boldsymbol{\Omega}\right)\\
=& p\left(\mathbf{T}|\mathbf{W},\boldsymbol{\Omega}\right) p\left(\mathbf{W}|\boldsymbol{\alpha},\boldsymbol{\Omega}\right)\\
=& \rho \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\left(\mathbf{T}-\boldsymbol{\Phi}\mathbf{W}\right)^{\intercal}\left(\mathbf{T}-\boldsymbol{\Phi}\mathbf{W}\right)\right)\right)\exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{W}^{\intercal}\mathbf{A}\mathbf{W}\right)\right),\\
& \text{where } \rho=(2\pi)^{-\frac{VN}{2}}\left|\boldsymbol{\Omega}\right|^{-\frac{N}{2}}(2\pi)^{-\frac{V(N+1)}{2}}\left|\boldsymbol{\Omega}\right|^{-\frac{N+1}{2}}\left|\mathbf{A}\right|^{\frac{V}{2}}\\
=& \rho \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\left(\left(\mathbf{T}^{\intercal}-\mathbf{W}^{\intercal}\boldsymbol{\Phi}^{\intercal}\right)\left(\mathbf{T}-\boldsymbol{\Phi}\mathbf{W}\right)+\mathbf{W}^{\intercal}\mathbf{A}\mathbf{W}\right)\right)\right)\\
=& \rho \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\left(\mathbf{T}^{\intercal}\mathbf{T}-\mathbf{T}^{\intercal}\boldsymbol{\Phi}\mathbf{W}-\mathbf{W}^{\intercal}\boldsymbol{\Phi}^{\intercal}\mathbf{T}+\mathbf{W}^{\intercal}\boldsymbol{\Sigma}^{-1}\mathbf{W}\right)\right)\right),\ \text{where } \boldsymbol{\Sigma}=\left(\boldsymbol{\Phi}^{\intercal}\boldsymbol{\Phi}+\mathbf{A}\right)^{-1}\\
=& \rho \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\left(\mathbf{T}^{\intercal}\mathbf{T}-\mathbf{T}^{\intercal}\boldsymbol{\Phi}\mathbf{W}+\mathbf{W}^{\intercal}\boldsymbol{\Sigma}^{-1}\left(\mathbf{W}-\mathbf{M}\right)\right)\right)\right),\text{where } \mathbf{M}=\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\intercal}\mathbf{T}\\
=& \rho \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\left(\mathbf{T}^{\intercal}\mathbf{T}-\mathbf{T}^{\intercal}\boldsymbol{\Phi}\mathbf{W}+\mathbf{W}^{\intercal}\boldsymbol{\Sigma}^{-1}\left(\mathbf{W}-\mathbf{M}\right)-\mathbf{M}^{\intercal}\boldsymbol{\Sigma}^{-1}\left(\mathbf{W}-\mathbf{M}\right)+\mathbf{M}^{\intercal}\boldsymbol{\Sigma}^{-1}\left(\mathbf{W}-\mathbf{M}\right)\right)\right)\right)\\
=& \rho \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\left(\left(\mathbf{W}-\mathbf{M}\right)^{\intercal}\boldsymbol{\Sigma}^{-1}\left(\mathbf{W}-\mathbf{M}\right)+\mathbf{M}^{\intercal}\boldsymbol{\Sigma}^{-1}\left(\mathbf{W}-\mathbf{M}\right)+\mathbf{T}^{\intercal}\mathbf{T}-\mathbf{T}^{\intercal}\boldsymbol{\Phi}\mathbf{W}\right)\right)\right)\\
=& \rho \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\left(\left(\mathbf{W}-\mathbf{M}\right)^{\intercal}\boldsymbol{\Sigma}^{-1}\left(\mathbf{W}-\mathbf{M}\right)+\mathbf{T}^{\intercal}\boldsymbol{\Phi}\left(\mathbf{W}-\mathbf{M}\right)+\mathbf{T}^{\intercal}\mathbf{T}-\mathbf{T}^{\intercal}\boldsymbol{\Phi}\mathbf{W}\right)\right)\right),\\
& \text{since } \boldsymbol{\Sigma}=\boldsymbol{\Sigma}^{\intercal}\\
=& \rho \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\left(\left(\mathbf{W}-\mathbf{M}\right)^{\intercal}\boldsymbol{\Sigma}^{-1}\left(\mathbf{W}-\mathbf{M}\right)+\mathbf{T}^{\intercal}\left(\mathbf{T}-\boldsymbol{\Phi}\mathbf{M}\right)\right)\right)\right)\\
=& \rho \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\left(\left(\mathbf{W}-\mathbf{M}\right)^{\intercal}\boldsymbol{\Sigma}^{-1}\left(\mathbf{W}-\mathbf{M}\right)+\mathbf{T}^{\intercal}\left(\mathbf{I}-\boldsymbol{\Phi}\left(\boldsymbol{\Phi}^{\intercal}\boldsymbol{\Phi}+\mathbf{A}\right)^{-1}\boldsymbol{\Phi}^{\intercal}\right)\mathbf{T}\right)\right)\right)\\
=& \rho \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\left(\left(\mathbf{W}-\mathbf{M}\right)^{\intercal}\boldsymbol{\Sigma}^{-1}\left(\mathbf{W}-\mathbf{M}\right)+\mathbf{T}^{\intercal}\left(\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^{\intercal}\right)^{-1}\mathbf{T}\right)\right)\right),\\
& \text{by the Woodbury matrix identity}\\
=& (2\pi)^{-\frac{V(N+1)}{2}}\left|\boldsymbol{\Omega}\right|^{-\frac{N+1}{2}}\left|\boldsymbol{\Sigma}\right|^{-\frac{V}{2}}\exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\left(\mathbf{W}-\mathbf{M}\right)^{\intercal}\boldsymbol{\Sigma}^{-1}\left(\mathbf{W}-\mathbf{M}\right)\right)\right)\\
& (2\pi)^{-\frac{VN}{2}}\left|\boldsymbol{\Omega}\right|^{-\frac{N}{2}}\left|\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^{\intercal}\right|^{-\frac{V}{2}}\exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Omega}^{-1}\mathbf{T}^{\intercal}\left(\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^{\intercal}\right)^{-1}\mathbf{T}\right)\right)
\end{aligned}
$$

Chapter 2. Review of online portfolio selection: Performance comparison with transaction costs including market impact costs

1. Introduction

Online (sequential) portfolio selection (OPS) aims to maximise the portfolio's expected terminal wealth in the long run and differs from single-period (Markowitz 1959) or multi-period (Dantzig and Infanger 1993, Ben-Tal *et al.* 2000, Li and Ng 2000, Steinbach 2001) mean-variance portfolio selection (MVPS). To be specific, OPS makes minimal statistical assumptions about the behaviour of the stock market (e.g. stationarity and ergodicity), but MVPS assumes a log-normal distribution of stock returns (Markowitz 1959). Furthermore, OPS aims for higher expected terminal wealth without considering the variance (risk), but MVPS considers trade-off between the mean (expected wealth) and variance (risk).

Both OPS and multi-period MVPS rebalance a portfolio periodically before an investor obtains the final reward. The differences between them are i) the uncertainty of input data for portfolio rebalancing and ii) portfolio rebalancing frequency: OPS uses deterministic data (e.g. historical stock returns), but multi-period MVPS uses stochastic data (e.g. mean and covariance of stock returns' distribution). Besides, OPS rebalances a portfolio at the same frequency as the input data, but multi-period MVPS rebalances a portfolio at a lower frequency than the input data.

OPS differs from portfolio optimisation using a risk measure based either on Value-at-Risk (VaR) or on conditional VaR (CVaR) although neither of them nor OPS assumes any specific distribution of stock returns (Gaivoronski and Pflug 2005). However, OPS uses stock return time series spanning between the first day of investment and the current day, whereas portfolio optimisation using either VaR or CVaR uses the finite samples in the left tail of historical stock returns spanning before the first day of investment.

OPS directly optimises a portfolio in terms of the long-term investment without forecasting (Li and Hoi 2014), and it differs from the previous studies of prediction-based portfolio selection (Freitas *et al.* 2009, Otranto 2010, Brown and Smith 2011, Ferreira and Santa-Clara 2011, Gârleanu and Pedersen 2013, DeMiguel *et al.* 2014, Palczewski *et al.* 2015), which i) forecasts the expected values or covariance matrix of stock returns[1] and ii) uses the mean-variance optimisation.[2] Therefore, OPS neither suffers from the difficulty of the prediction nor uses in-sample and out-of-sample tests.

However, all existing OPS methods (Blum and Kalai 1999, Györfi and Vajda 2008, Kozat and Singer 2011, Bean and Singer 2012, Györfi and Walk 2012, Tunc *et al.* 2013, Das *et al.* 2013, 2014) have not considered market impact costs by limited liquidity although they have considered proportional transaction costs. Therefore, the aim of this chapter is to develop a new model of transaction cost factor (TCF; this will be explained in Section 5) by considering the limit order

[1] Dynamic portfolio selection with transaction costs prevents too much trading by using multi-period prediction in the time horizon from $t+1$ to $t+h$ (Brown and Smith 2011), and from $t+1$ to ∞ (Gârleanu and Pedersen 2013), where t is the current period. Both the methods show better backtesting results than their benchmarks with single-period prediction of $t+1$.

[2] Brown and Smith (2011), Palczewski *et al.* (2015) used risk-averse utility functions instead of the mean-variance portfolio. In addition, DeMiguel *et al.* (2014) constructed an arbitrage (zero-cost) portfolio, creating a zero net value, as well as the mean-variance portfolio.

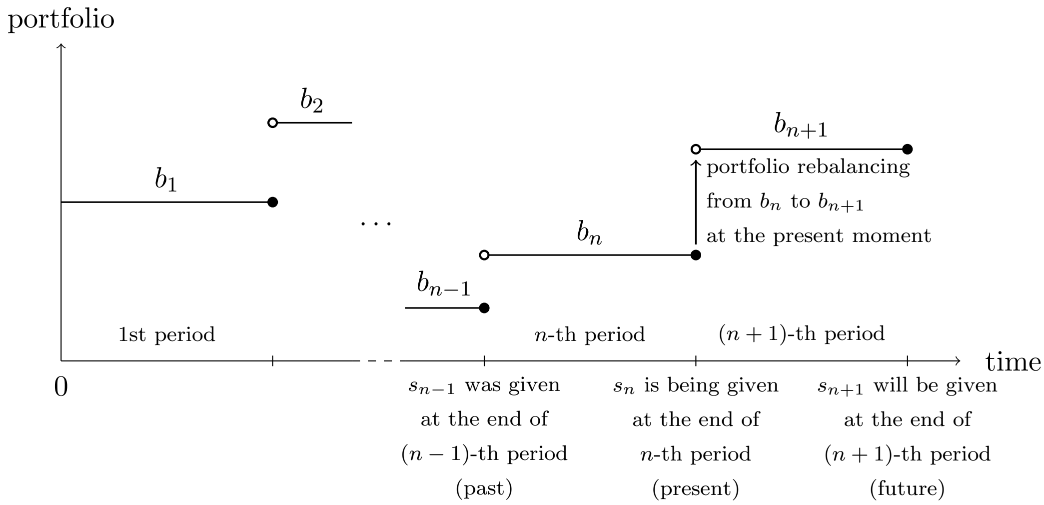


Figure 1. Timeline of trading. A portfolio is rebalanced at the end of every period, where an asset's price is given at the end of every period. The end of n-th period is the present moment for rebalancing.

books (LOBs) of stocks of which a portfolio consists. All the previous methods of OPS (these will be reviewed in Section 3) assumed that an investor can buy or sell any quantities of stocks, which in turn makes the OPS methods impractical: they made investment strategies of rebalancing a portfolio under the ideal assumption of unlimited liquidity of assets, and their numerical results overestimated the performance of OPS. However, the new model will overcome these problems by using LOB data, which reflects the liquidity of assets.

The contributions of this chapter are:

- in Section 5.2 and Section 5.3, to propose the new TCF model which reflects LOBs as well as a fixed percentage fee;
- in Section 8, to present the backtesting results of a comparison among OPS methods (including the proposed method) by using the real-world data (historical NASDAQ LOB data).

The rest of this chapter is organised as follows: Section 2 lists mathematical notations. Section 3 describes the existing methods of OPS. Section 4 includes the motivation of this chapter and the review of the mathematical formulation of market impact costs (MICs) as a function of order size. Section 5 reviews a TCF model without MICs and proposes the new TCF model with MICs. Section 6 reviews a log-optimal portfolio, one of the OPS methods, which is the basis of the proposed method. Section 7 describes the proposed OPS method. Section 8 demonstrates the performance of OPS methods including the proposed method by computer simulations (backtesting). Section 9 gives the conclusion.

2. Mathematical setup

The following notations are used in this chapter:

- $\boldsymbol{b}_n = \left[b_n^{(1)}\ b_n^{(2)}\ \dots\ b_n^{(d)}\right]^{\mathrm{T}}$ is a portfolio vector of d risky assets (there is no risk-free asset in the portfolio) at the n-th period (trading occurs in a fixed interval such as a day or a week at the end of every period as shown in Figure 1), where $n \in \mathbb{Z}_{\geq 1}$, $b_n^{(j)} \in \mathbb{R}_{\geq 0}$ (i.e. neither short selling nor buying stocks on margin is permitted), and $\sum_{j=1}^{d} b_n^{(j)} = 1$ (i.e. $b_n^{(j)}$ is the proportion of a portfolio invested in asset $j \in \{1, 2, \dots, d\}$ at the n-th period). Hence, $\boldsymbol{b}_n \in \Delta^{d-1}$, where $\Delta^{d-1} = \left\{ \left[b^{(1)}\ b^{(2)}\ \dots\ b^{(d)}\right]^{\mathrm{T}} \in \mathbb{R}^d_{\geq 0} \,\middle|\, \sum_{j=1}^{d} b^{(j)} = 1 \right\}$ is the standard $(d-1)$-simplex.
- $\boldsymbol{b}_1 = [1/d\ 1/d\ \dots\ 1/d]^{\mathrm{T}}$ is an initial portfolio vector.

- $s_n^{(j)}$ is the price of asset j at the end of the n-th period (see Figure 1).
- $x_n^{(j)} = \dfrac{s_n^{(j)}}{s_{n-1}^{(j)}}$ is a price relative of asset j at the end of the n-th period.
- $\boldsymbol{x}_n = \left[x_n^{(1)}\ x_n^{(2)}\ \ldots\ x_n^{(d)}\right]^{\mathrm{T}} \in \mathbb{R}_{>0}^d$ is a market vector at the end of the n-th period.
- $\boldsymbol{x}_{i:j} = \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{x}_{i+1} \\ \vdots \\ \boldsymbol{x}_j \end{bmatrix} \in \mathbb{R}_{>0}^{d(j-i+1)}$ is the array of the market vectors, where $i \leq j$.

3. Review of the existing methods of online portfolio selection

The two categories of OPS, classified according to whether considering transaction costs (TCs) or not, are reviewed in the following subsections.

3.1. *Online portfolio selection without transaction costs*[1]

The most basic benchmark of OPS is a best constant rebalanced portfolio (BCRP), whose portfolio vector is

$$\boldsymbol{b}^* = \underset{\boldsymbol{b} \in \Delta^{d-1}}{\arg\max} \prod_{i=1}^{T} \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle, \tag{1}$$

where T is the last period of trading, and $\langle \cdot, \cdot \rangle$ denotes the inner product. This is a hindsight strategy that can only be calculated with complete market vectors from the first period to the last period in the future $\boldsymbol{x}_{1:T}$; hence, BCRP is unimplementable as an investment strategy but available only for benchmark.

A universal portfolio (UP) is the beginning of OPS and performs asymptotically as well as BCRP under the assumption that stock returns are a stationary ergodic time series (Cover 1991). The portfolio vector of UP of the next period $n+1$ (n is the current period as shown in Figure 1) is

$$\boldsymbol{b}_{n+1} = \frac{\int_{\Delta^{d-1}} \boldsymbol{b} S_n(\boldsymbol{b}, \boldsymbol{x}_{1:n}) f(\boldsymbol{b}) d\boldsymbol{b}}{\int_{\Delta^{d-1}} S_n(\boldsymbol{b}, \boldsymbol{x}_{1:n}) f(\boldsymbol{b}) d\boldsymbol{b}},^{2} \tag{2}$$

where $S_n(\boldsymbol{b}, \boldsymbol{x}_{1:n}) = S_0 \prod_{i=1}^{n} \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle$ is wealth at the end of the n-th period with an initial wealth S_0, and $f(\cdot)$ is the probability density function (PDF) of the Dirichlet distribution with the d-dimensional concentration parameter vector $[1/2\ 1/2\ \ldots\ 1/2]^{\mathrm{T}}$. The UP strategy is a follow-the-winner approach according to Li and Hoi (2014) as it increases the relative weights $S_n(\boldsymbol{b}, \boldsymbol{x}_{1:n})$ of more successful assets in the past. In addition, Cover and Ordentlich (1996) extended UP (Cover 1991) to UP with side information, which uses additional information concerning the stock market (e.g. a series of trading signals).

A non-parametric (i.e. the distribution of the market vector is unknown) kernel-based sequential log-optimal investment strategy[3] guarantees an optimal asymptotic growth rate of capital under minimal assumptions on the behaviour of the market (i.e. daily price relatives $\boldsymbol{x}_i$ are K-th order

---

[1] A detailed survey of OPS without TCs was carried out by Li and Hoi (2014), and Das (2014, pp. 22–29).
[2] The mathematical proof of (2) is more easily explained in (Cover and Thomas 2006, Chapter 16) than (Cover 1991).
[3] Log-optimal portfolio is explained in Section 6.

stationary Markov processes) (Györfi *et al.* 2006). Its portfolio vector of the next period is

$$\boldsymbol{b}_{n+1} = \frac{\sum_{k=1}^{K}\sum_{l=1}^{L} \boldsymbol{h}^{(k,l)}\left(\boldsymbol{x}_{1:n}\right) S_n\left(\boldsymbol{h}^{(k,l)}\left(\boldsymbol{x}_{1:n}\right), \boldsymbol{x}_{1:n}\right)}{\sum_{k=1}^{K}\sum_{l=1}^{L} S_n\left(\boldsymbol{h}^{(k,l)}\left(\boldsymbol{x}_{1:n}\right), \boldsymbol{x}_{1:n}\right)}, \tag{3}$$

where $S_n\left(\boldsymbol{h}^{(k,l)}\left(\boldsymbol{x}_{1:n}\right), \boldsymbol{x}_{1:n}\right) = S_0 \prod_{i=1}^{n} \left\langle \boldsymbol{h}^{(k,l)}\left(\boldsymbol{x}_{1:n}\right), \boldsymbol{x}_i \right\rangle$ is wealth at the end of the n-th period with an initial wealth S_0 and expert $\boldsymbol{h}^{(k,l)}(\cdot)$: i.e. the higher wealth $S_n\left(\boldsymbol{h}^{(k,l)}(\cdot), \cdot\right)$; the greater weight on $\boldsymbol{b}_{n+1}$. The portfolio vector of the expert $\boldsymbol{h}^{(k,l)}(\cdot)$ from the series of past and current market vectors $\boldsymbol{x}_{1:n}$ is

$$\boldsymbol{h}^{(k,l)}\left(\boldsymbol{x}_{1:n}\right) = \begin{cases} \boldsymbol{b}_1, & \text{if } J_n = \varnothing \\ \underset{\boldsymbol{b}\in\Delta^{d-1}}{\arg\max} \sum_{i\in J_n} \ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle, {}^{1} & \text{otherwise} \end{cases}, \tag{4}$$

where J_n is the locations of matches:

$$J_n = \left\{k \le i \le n-1 : \|\boldsymbol{x}_{i-k+1:i} - \boldsymbol{x}_{n-k+1:n}\| \le l/c\right\}. \tag{5}$$

This strategy is classified as a pattern-matching-based approach by Li and Hoi (2014) as it finds the matching in Euclidean space between the past market vectors $\boldsymbol{x}_{i-k+1:i}$ and the most recent market vectors $\boldsymbol{x}_{n-k+1:n}$. According to their numerical results, it outperformed UP (Cover 1991) although its performance (i.e. the terminal wealth) varies by the choice of the three free parameters: K, L, and c in (3) and (5).

The computation time to calculate $\boldsymbol{h}^{(k,l)}(\cdot)$ in (4) was decreased by transforming the constrained (i.e. $\boldsymbol{b} \in \Delta^{d-1}$ in (4)) nonlinear (i.e. the log function in (4)) programming to the constrained quadratic programming (Györfi *et al.* 2007):

$$\bar{\boldsymbol{h}}^{(k,l)}\left(\boldsymbol{x}_{1:n}\right) = \underset{\boldsymbol{b}\in\Delta^{d-1}}{\arg\max} \sum_{i\in J_n} \left(\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle - 1 - \frac{1}{2}\left(\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle - 1\right)^2\right), \ J_n \neq \varnothing, \tag{6}$$

(i.e. the second-order Taylor expansion of the log function was used). Moreover, the computation time of the optimisation problem of (6) was reduced further by Györfi *et al.* (2007) by rewriting (6) as

$$\bar{\boldsymbol{h}}^{(k,l)}\left(\boldsymbol{x}_{1:n}\right) = \underset{\boldsymbol{b}\in\Delta^{d-1}}{\arg\max} \left(\langle \boldsymbol{b}, \boldsymbol{m}\rangle - \langle \boldsymbol{b}, \boldsymbol{\Sigma}\boldsymbol{b}\rangle\right), {}^{2} \ J_n \neq \varnothing, \tag{7}$$

where $\boldsymbol{m}$ is the d-dimensional column vector ($\boldsymbol{1}$ denotes the all-ones column vector):

$$\boldsymbol{m} = \sum_{i\in J_n} (\boldsymbol{x}_{i+1} - \boldsymbol{1}),$$

[1] $\arg\min_{\boldsymbol{b}\in\Delta^{d-1}} \left(-\sum_{i\in J_n} \ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle\right)$ is a convex optimisation problem (proof is in Appendix B), which means any local solution is guaranteed to be a global solution.

[2] $\arg\min_{\boldsymbol{b}\in\Delta^{d-1}} \left(\langle \boldsymbol{b}, \boldsymbol{\Sigma}\boldsymbol{b}\rangle - \langle \boldsymbol{b}, \boldsymbol{m}\rangle\right)$ in (7) is a convex optimisation problem (technically, a quadratic optimisation problem); thus, any local solution is guaranteed to be a global solution.

and $\boldsymbol{\Sigma}$ is the $d \times d$ matrix:

$$\boldsymbol{\Sigma} = \frac{1}{2} \sum_{i \in J_n} (\boldsymbol{x}_{i+1} - \boldsymbol{1})(\boldsymbol{x}_{i+1} - \boldsymbol{1})^{\mathrm{T}}.$$

As a result, if we calculate $\boldsymbol{m}$ and $\boldsymbol{\Sigma}$ beforehand, the complexity of the optimisation problem of (7) does not depend on the number of matches $|J_n|$ (i.e. the size of $\boldsymbol{m}$ and $\boldsymbol{\Sigma}$ is fixed even if $|J_n|$ in (7) changes), while that of (6) does.

Ottucsák and Vajda (2007) showed a relationship between the log-optimal portfolio strategy (Györfi *et al.* 2006) and the single-period MVPS by rewriting (6) as the equation form of the single-period MVPS:

$$\begin{aligned}
\bar{\boldsymbol{h}}^{(k,l)}(\boldsymbol{x}_{1:n}) &= \operatorname*{arg\,max}_{\boldsymbol{b} \in \Delta^{d-1}} \sum_{i \in J_n} \left(\langle \boldsymbol{b}, \boldsymbol{x}_{i+1} \rangle - 1 - \frac{1}{2} \left(\langle \boldsymbol{b}, \boldsymbol{x}_{i+1} \rangle - 1 \right)^2 \right), \ J_n \neq \varnothing, \\
&\approx \operatorname*{arg\,max}_{\boldsymbol{b} \in \Delta^{d-1}} \left(2E_{n+1}(\boldsymbol{b}) - \frac{1}{2} E_{n+1}(\boldsymbol{b})^2 - \frac{3}{2} \right) \\
&\approx \operatorname*{arg\,max}_{\boldsymbol{b} \in \Delta^{d-1}} \left[2E_{n+1}(\boldsymbol{b}) - \frac{1}{2} \left(E_{n+1}(\boldsymbol{b})^2 - \left(E_{n+1}(\boldsymbol{b})\right)^2 \right) - \frac{1}{2} \left(E_{n+1}(\boldsymbol{b})\right)^2 \right] \\
&\approx \operatorname*{arg\,max}_{\boldsymbol{b} \in \Delta^{d-1}} \left[E_{n+1}(\boldsymbol{b}) \left(4 - E_{n+1}(\boldsymbol{b})\right) - V_{n+1}(\boldsymbol{b}) \right] \\
&\approx \operatorname*{arg\,max}_{\boldsymbol{b} \in \Delta^{d-1}} \left(\frac{E_{n+1}(\boldsymbol{b})}{\lambda_{n+1}} - V_{n+1}(\boldsymbol{b}) \right)
\end{aligned} \tag{8}$$

where $E_{n+1}(\boldsymbol{b}) \stackrel{\text{def}}{=} \mathbb{E}\left[\langle \boldsymbol{b}, \boldsymbol{X}_{n+1} \rangle | \boldsymbol{X}_n = \boldsymbol{x}_n, \boldsymbol{X}_{n-1} = \boldsymbol{x}_{n-1}, \ldots, \boldsymbol{X}_1 = \boldsymbol{x}_1\right]$, the conditional expected value of the dot product between a portfolio vector $\boldsymbol{b}$ and the price relative at the end of the $(n+1)$-th period $\boldsymbol{x}_{n+1}$ (see Figure 1), is approximately equal to

$$E_{n+1}(\boldsymbol{b}) \approx \frac{1}{|J_n|} \sum_{i \in J_n} \langle \boldsymbol{b}, \boldsymbol{x}_{i+1} \rangle, \tag{9}$$

$E_{n+1}(\boldsymbol{b})^2 \stackrel{\text{def}}{=} \mathbb{E}\left[\langle \boldsymbol{b}, \boldsymbol{X}_{n+1} \rangle^2 | \boldsymbol{X}_n = \boldsymbol{x}_n, \boldsymbol{X}_{n-1} = \boldsymbol{x}_{n-1}, \ldots, \boldsymbol{X}_1 = \boldsymbol{x}_1\right]$, the conditional second-order moment of the dot product, is approximately equal to

$$E_{n+1}(\boldsymbol{b})^2 \approx \frac{1}{|J_n|} \sum_{i \in J_n} \langle \boldsymbol{b}, \boldsymbol{x}_{i+1} \rangle^2, \tag{10}$$

$V_{n+1}(\boldsymbol{b}) \stackrel{\text{def}}{=} \operatorname{Var}\left(\langle \boldsymbol{b}, \boldsymbol{X}_{n+1} \rangle | \boldsymbol{X}_n = \boldsymbol{x}_n, \boldsymbol{X}_{n-1} = \boldsymbol{x}_{n-1}, \ldots, \boldsymbol{X}_1 = \boldsymbol{x}_1\right)$, the conditional variance of the dot product at the end of the $(n+1)$-th period, is approximately equal to

$$\begin{aligned}
V_{n+1}(\boldsymbol{b}) =& E_{n+1}(\boldsymbol{b})^2 - \left(E_{n+1}(\boldsymbol{b})\right)^2 \\
\approx& \frac{1}{|J_n|} \sum_{i \in J_n} \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle^2 - \left(\frac{1}{|J_n|} \sum_{i \in J_n} \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle \right)^2,
\end{aligned} \tag{11}$$

and λ_{n+1}, the coefficient of risk aversion at the end of the $(n+1)$-th period, is defined as

$$\lambda_{n+1} \stackrel{\text{def}}{=} \frac{1}{4 - E_{n+1}(\boldsymbol{b})}, \tag{12}$$

under the assumption that the market vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots$ are realisations of random vectors $\boldsymbol{X}_1, \boldsymbol{X}_2, \dots$ drawn from a d-dimensional vector-valued stationary and ergodic process $\{\boldsymbol{X}_i, i \in \mathbb{Z}\}$. I.e. the log-optimal strategy (Györfi *et al.* 2006) dynamically adjusts the coefficient of risk aversion λ_n while the single-period MPVS uses a constant value of the coefficient. As a result, the log-optimal portfolio strategy (Györfi *et al.* 2006) outperformed the single-period MVPS in growth rate (the terminal wealth in the long run) in their numerical experiments.

Györfi *et al.* (2008b) also assumed that daily price relatives are K-th order stationary Markov process as the same as (Györfi *et al.* 2006) and made a nearest-neighbour(NN)-based investment strategy with two free parameters: K and L. The difference between the two strategies is only the way to find the locations of matching J_n in (5): instead of the threshold of the distance l/c in (5), the criterion of NN in Euclidean space is used as

$$J_n = \left\{ k \leq i \leq n-1 \;\middle|\; \boldsymbol{x}_{i-k+1:i} \text{ is among the } \hat{l} \text{ NNs of } \boldsymbol{x}_{n-k+1:n} \text{ in } \boldsymbol{x}_{1:k}, \boldsymbol{x}_{2:k+1}, \dots, \boldsymbol{x}_{n-k:n-1} \right\}, \tag{13}$$

where $\hat{l} = \lfloor p_l n \rfloor$, and $p_l = 0.02 + 0.5\frac{l-1}{L-1}$, specified by Györfi *et al.* (2008a). Their experimental results showed that the NN-based method (Györfi *et al.* 2008b) outperforms the kernel-based method (Györfi *et al.* 2006) in terms of the terminal wealth and the robustness of choosing suitable free parameters of K and L. Furthermore, as the terminal wealth of the NN-based method with the order of Markov process $k = 1$ was the highest among $k = \{1, 2, \dots, 5\}$ for all of $l = \{1, 2, \dots, 10\}$, we can infer that the market vector $\boldsymbol{x}_n$ is a first-order Markov process rather than multiple-order.

Horváth and Urbán (2012) defined the sets of possible portfolio vectors of OPS with short selling or leverage (i.e. borrowing money). In order to allow short selling, the original constraints of the proportion of a portfolio invested in asset j, $b^{(j)} \in [0, 1]$ and $\sum_{j=1}^{d} b^{(j)} = 1$, are replaced with $b^{(j)} \in [-1, 1]$ and $\sum_{j=1}^{d} b^{(j)+} = 1$, where $x^+ \stackrel{\text{def}}{=} \max(0, x)$. In order to allow borrowing money, the original constraints are replaced with $b^{(j)} \in [0, \infty)$ and $\sum_{j=1}^{d} b^{(j)} = B$, where B is the maximum investable amount. The three sets (short selling only, leverage only, and both) are applicable to any OPS method without TCs.

3.2. *Online portfolio selection with transaction costs*

UP with TCs by Blum and Kalai (1999) is a trivial extension of UP (Cover 1991). Instead of accumulated wealth without TCs S_n in (2), that with TCs N_n (net wealth N_n is defined in Section 5) is used to calculate the portfolio vector of the next period $n + 1$:

$$\boldsymbol{b}_{n+1} = \frac{\int_{\Delta^{d-1}} \boldsymbol{b} N_n(\boldsymbol{b}, \boldsymbol{x}_{1:n}) f(\boldsymbol{b}) d\boldsymbol{b}}{\int_{\Delta^{d-1}} N_n(\boldsymbol{b}, \boldsymbol{x}_{1:n}) f(\boldsymbol{b}) d\boldsymbol{b}}, \tag{14}$$

where $N_n(\boldsymbol{b}, \boldsymbol{x}_{1:n}) = S_0 \prod_{i=1}^{n} (\langle \boldsymbol{b}, \boldsymbol{x}_i \rangle - C_i)$ is wealth at the end of the n-th period with an initial wealth S_0 and the TC at the end of the i-th period C_i.

Györfi and Vajda (2008) i) extended the investment strategy by Györfi *et al.* (2006) by considering TCs and ii) simplified the assumption of the market from the multiple-order Markov process to a first-order Markov process. They suggested an implementable but suboptimal algorithm with one free parameter L:

$$\boldsymbol{b}_{n+1} = \frac{\sum_{l=1}^{L} \boldsymbol{h}^{(l)}(\boldsymbol{x}_{1:n}) S_n\left(\boldsymbol{h}^{(l)}(\boldsymbol{x}_{1:n}), \boldsymbol{x}_{1:n}\right)}{\sum_{l=1}^{L} S_n\left(\boldsymbol{h}^{(l)}(\boldsymbol{x}_{1:n}), \boldsymbol{x}_{1:n}\right)}, \tag{15}$$

where $S_n\left(\boldsymbol{h}^{(l)}\left(\boldsymbol{x}_{1:n}\right), \boldsymbol{x}_{1:n}\right) = S_0 \prod_{i=1}^{n}\left\langle \boldsymbol{h}^{(l)}\left(\boldsymbol{x}_{1:n}\right), \boldsymbol{x}_i\right\rangle$ is wealth at the end of the n-th period with an initial wealth S_0.[1] The portfolio vector of the expert $\boldsymbol{h}^{(l)}(\cdot)$ from the series of past and current market vectors $\boldsymbol{x}_{1:n}$ is

$$\boldsymbol{h}^{(l)}\left(\boldsymbol{x}_{1:n}\right) = \begin{cases} \boldsymbol{b}_1, & \text{if } J_n = \varnothing \\ \underset{\boldsymbol{b}\in\Delta^{d-1}}{\arg\max} \sum_{i\in J_n} \left(\ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle + \ln w_n\right), & \text{otherwise} \end{cases}, \tag{16}$$

where J_n is the locations of matches between past market vector $\boldsymbol{x}_i$ and current one $\boldsymbol{x}_n$:

$$J_n = \left\{1 \le i \le n-1 \;\middle|\; \|\boldsymbol{x}_i - \boldsymbol{x}_n\| \le \sqrt{0.0001dl}\right\}. \tag{17}$$

I.e. the difference between (4) and (16) is only the addition of $\ln w_n$ (transaction cost factor w_n will be explained in Section 5) that considers TCs.[2]

Kozat and Singer (2011) proposed a sequential universal portfolio whose achieved wealth is asymptotically as large as the wealth achieved by the best semi-constant rebalanced portfolio: a semi-constant rebalanced portfolio rebalances its portfolio only on selected instants to reduce TCs, while a constant rebalanced portfolio rebalances it at every period. Their portfolio vector of the next period is

$$\boldsymbol{b}_{n+1} = \sum_{i=0}^{n}\sum_{j=1}^{d} \sigma_n(i,j)\Big(F_{\mathrm{tr}}(s_{n+1}=i|s_n=i)\boldsymbol{e}_j + F_{\mathrm{tr}}(s_{n+1}=n+1|s_n=i)\boldsymbol{b}_1\Big), \tag{18}$$

where $\boldsymbol{e}_j \stackrel{\text{def}}{=} [0 \;\ldots\; 0\;1\;0 \;\ldots\; 0]^{\mathrm{T}}$ is a vector of all zeros except a single one at location j, and the weight of asset j at state i is defined as

$$\sigma_n(i,j) \stackrel{\text{def}}{=} \frac{W_n(\boldsymbol{x}_n, i, j)}{\sum\limits_{k=0}^{n}\sum\limits_{l=1}^{d} W_n(\boldsymbol{x}_n, k, l)}. \tag{19}$$

Hence, this strategy can be classified as the follow-the-winner approach as the weight $\sigma_n(i,j)$ is proportional to the weighted wealth of asset j at state i at the end of the n-th period $W_n(\boldsymbol{x}_n, i, j)$, $0 \le i \le n$. This weighted wealth is calculated from the previous (i.e. $(n-1)$-th period) weighted wealth $W_{n-1}(\boldsymbol{x}_{n-1}, i, j)$, $0 \le i \le n-1$ (see Figure 2):

$$W_n(\boldsymbol{x}_n, i, j) \stackrel{\text{def}}{=} \begin{cases} W_{n-1}(\boldsymbol{x}_{n-1}, i, j) F_{\mathrm{tr}}(s_n = i|s_{n-1}=i) x_n^{(j)}, & \text{if } 0 \le i \le n-1 \\ \sum\limits_{k=0}^{n-1}\left(\sum\limits_{l=1}^{d} W_{n-1}(\boldsymbol{x}_{n-1}, k, l)\right) F_{\mathrm{tr}}(s_n = n|s_{n-1}=k) b_1^{(j)} x_n^{(j)}, & \text{if } i = n \end{cases}. \tag{20}$$

$F_{\mathrm{tr}}(s_n = i|s_{n-1} = i)$ in both (18) and (20) is the horizontal path weight (i.e. there is no rebalancing to $\boldsymbol{b}_1$ as shown in Figure 2) from state i at the end of the $(n-1)$-th period to the same state i at

[1] Györfi and Vajda (2008) compared between i) aggregation with wealth (i.e. the expert in (16) makes the portfolio selection and pays TC individually) and ii) aggregation with portfolio (i.e. the aggregated portfolio b_{n+1} in (15) pays TC), which implies that the former uses equations (16) and (17) while the latter uses equations (15), (16), and (17). Their numerical experiments showed that the latter outperformed the former.

[2] Györfi and Vajda (2008) also introduced an optimal but unimplementable algorithm which solves a theoretical dynamic programming problem.

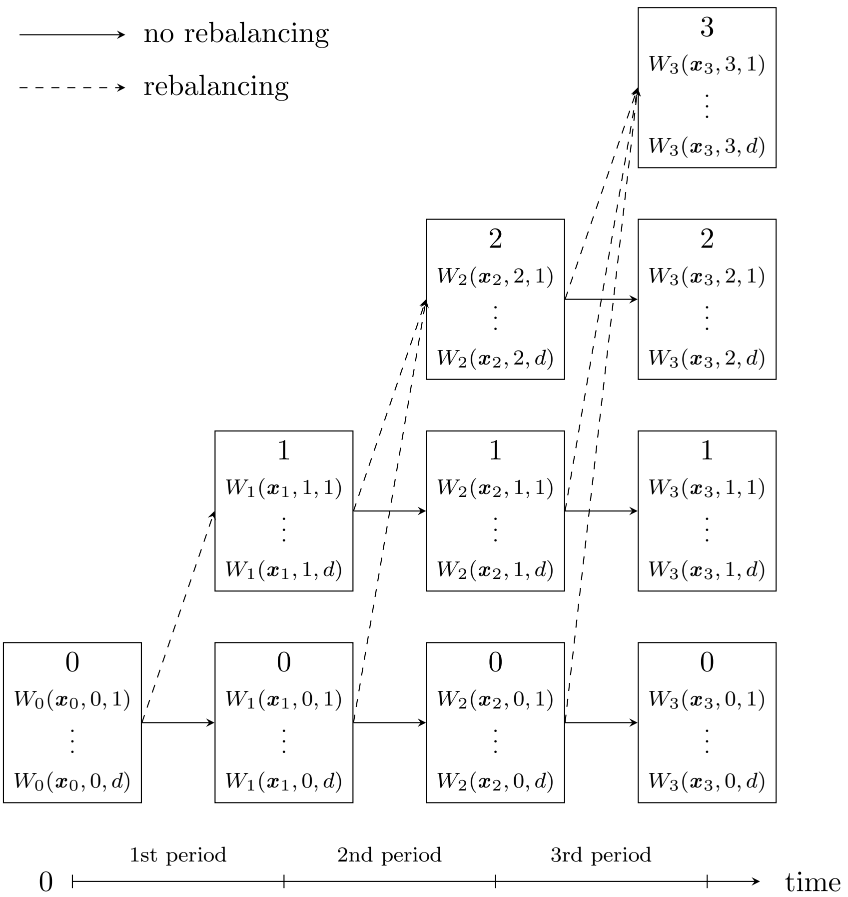


Figure 2. The rebalancing diagram of universal semi-constant rebalanced portfolio for $T = 3$ (a similar diagram is in (Kozat and Singer 2011, p. 300)). Each box represents a state, where each number in the box is the time of the last rebalancing instant, and it includes accumulated wealth $W_n(\boldsymbol{x}_n, i, j)$ for state $i \in \{0, 1, \ldots, n\}$ and for all assets $j = \{1, 2, \ldots, d\}$. A portfolio is rebalanced at the end of the n-th period by using the accumulated wealth $W_n(\boldsymbol{x}_n, i, j), \forall i, j$, where $x_0^{(j)} = 1, \forall j$.

the end of the n-th period (Kozat and Singer 2008):

$$F_{\mathrm{tr}}(s_n = i | s_{n-1} = i) \stackrel{\mathrm{def}}{=} \frac{n - 1 - i + \frac{1}{2}}{n - 1 - i + 1}, \tag{21}$$

and $F_{\mathrm{tr}}(s_n = n | s_{n-1} = i)$ in both (18) and (20) is the upward path weight (i.e. there is a rebalancing to $\boldsymbol{b}_1$ as shown in Figure 2) from state i at the end of the $(n-1)$-th period to the state n (i.e. the highest state in each period in Figure 2) at the end of the n-th period (Kozat and Singer 2008):

$$F_{\mathrm{tr}}(s_n = n | s_{n-1} = i) \stackrel{\mathrm{def}}{=} \frac{\frac{1}{2}}{n - 1 - i + 1}. \tag{22}$$

Algorithm 3 in Appendix A describes the investment strategy by Kozat and Singer (2011) in detail.

Bean and Singer (2012) combined UP with side information (Cover and Ordentlich 1996) and UP with TCs (Blum and Kalai 1999). Also, they employed factor graphs and a sum-product algorithm to derive computationally more efficient implementations of the combined UP. UP with side information under TCs (Bean and Singer 2012) achieves equal or greater wealth than UP with side information (Cover and Ordentlich 1996) under all simulated fixed percentage commissions. However, the Bean and Singer's method underperforms that of Blum and Kalai (1999).

Györfi and Walk (2012) extended the theoretical and unimplementable dynamic programming

Table 1. A 5-level limit order book of Microsoft Corporation, traded on NASDAQ, on 21 Jun 2012 at 16:00:00 (downloaded from `https://lobsterdata.com/info/DataSamples.php`). Bid-ask spread is USD 0.01, and midpoint price is USD 30.135.

	Level	Price (USD)	Volume (shares)
Asks	5	30.18	110,006
	4	30.17	86,886
	3	30.16	65,399
	2	30.15	80,663
	1	30.14	16,600
Bids	-1	30.13	-50,426
	-2	30.12	-83,306
	-3	30.11	-8,506
	-4	30.10	-43,838
	-5	30.09	-167,371

algorithm developed by Györfi and Vajda (2008) to two data-driven algorithms of the log-optimal investment, based on a partitioning-based portfolio selection rule and K-nearest-neighbour-based rule. However, both are still unimplementable (i.e. the algorithms cannot be transformed into a computer program) for rebalancing a portfolio.

The numerical results of Ormos and Urbán (2013) show that the implementable log-optimal strategy (Györfi and Vajda 2008) generates greater alpha values (excess return) of the CAPM or Fama–French model than a buy-and-hold strategy, even in the presence of proportional TCs (TC rate was set to 0.1% both for sale and purchase). Therefore, the log-optimal portfolio strategy shows some kind of market inefficiency, in the sense that the first-order Markov model is better than random stock selection.

A threshold rebalanced portfolio trades stocks only if a fraction of the portfolio is below a lower threshold or over an upper threshold to reduce TCs (Tunc *et al.* 2013). However, this approach is only available in two-stock markets, although Tunc *et al.* mentioned the possibility of its extension to markets having more than two stocks. Furthermore, their assumption that the prices of the two stocks follow two independent geometric Brownian motions does not consider the correlation between the two stocks.

Das *et al.* (2013) assumed that the portfolio vector in the current period is replicated in the next period (i.e. $\boldsymbol{x}_n = \boldsymbol{x}_{n+1}$) and added the transaction penalty term $\alpha\|\boldsymbol{b}_{n+1} - \boldsymbol{b}_n\|_1$, where $\|\cdot\|_1$ denotes the L^1 norm. The disadvantage of this method is that the parameter α which controls the amount of trading should be properly chosen, whereas Györfi and Vajda (2008)'s method does not require the user's care. Besides, Das *et al.* (2014) added a group sparsity term to the Das *et al.* (2013)'s method to increase portfolio weights on a few top performing sectors and beat the market. However, the additional term is also controlled by a user parameter. Hence, the performance of this OPS algorithm depends on both the user parameters: the transaction penalty parameter and the group sparsity parameter.

4. Market impact costs

Market impact costs (MICs, also called price impact costs) can be generated by an investor who trades on an asset, pushing the price up when buying the asset and pushing it down while selling (Damodaran 2012, Chapter 5).

4.1. *Limit order book*

An LOB (Table 1 is an example) is defined as the current set of active limit orders that is sent to, and maintained by, a security exchange or a security dealer (Levy and Post 2005, p. 68).

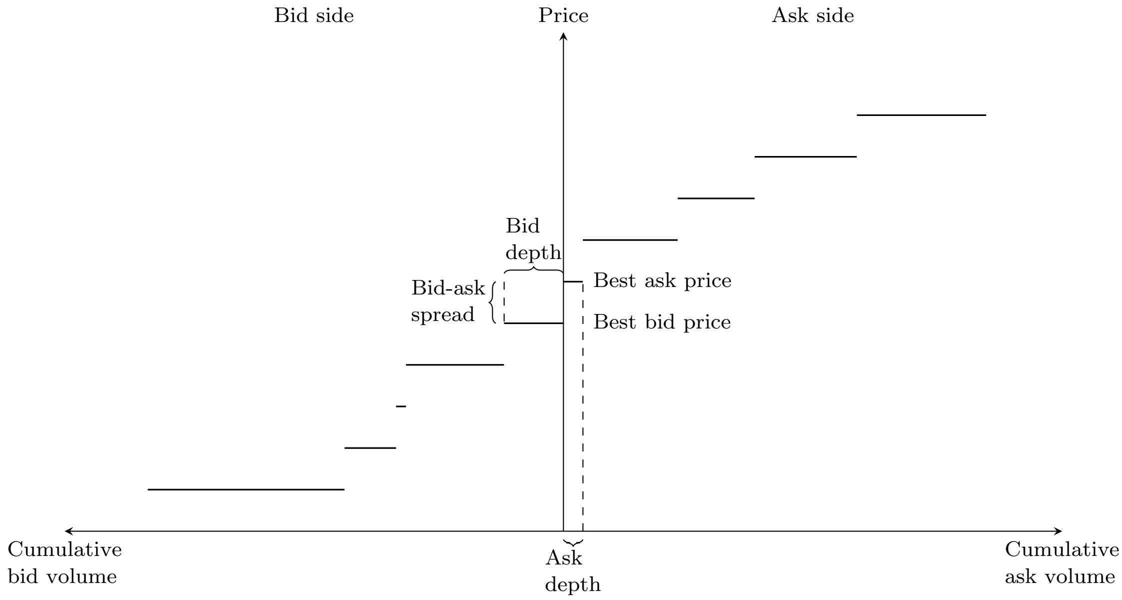


Figure 3. The liquidity of the limit order book in Table 1 (a similar diagram is in (Pristas 2007, p. 22)). The horizontal axis represents the cumulative volumes of the bid (ask) side on the left (right) hand side. The vertical axis shows the quoted prices.

4.2. *Market impact costs of online portfolio selection*

All the previous studies of OPS in Section 3 did not consider MICs but assumed that one can buy or sell any quantities of stock at its closing price. However, this is impracticable due to bid-ask spread and the finite depth of an LOB (Figure 3 shows both of them).

The bid-ask spread causes MICs.[1] The closing price is the last price at which a stock is traded on a day, and it is either the lowest ask price (if a buyer buys) or the highest bid price (if a seller sells). Therefore, it is possible either i) that the purchase price is greater than the closing price or ii) that the sale price is less than the closing price, which in turn will make the terminal wealth of OPS less than the ideal case of zero bid-ask spread. In other words, the gap between the closing price and the purchase (or sale) price occurs in every period and generates TCs whenever OPS rebalances a portfolio.

Furthermore, if the ask or bid depth (see Figure 3) is shallow, MICs increase as the size (in dollars) of a portfolio increases. If an OPS algorithm intends to trade for the amount greater than the depth, the order will be executed for the amount of the depth at the best quoted price (i.e. the lowest ask price or the highest bid price), and then the remaining part of the order will be executed at the next prices of an LOB, which in turn will increase MICs and decrease the terminal wealth of OPS.

Consequently, LOBs (e.g. Table 1) as well as fixed percentage TCs (e.g. brokerage commissions and transaction taxes) should be considered to make OPS strategies practical.

4.3. *Market impact costs as a function of order size in a limit order book*

MICs that occur when rebalancing a portfolio can be written as a function of order volumes and prices in LOBs. The average MIC as a function of order size q is defined as (Olsson 2005,

[1] MICs include the bid-ask spread costs in this chapter although they are separate in (Damodaran 2012).

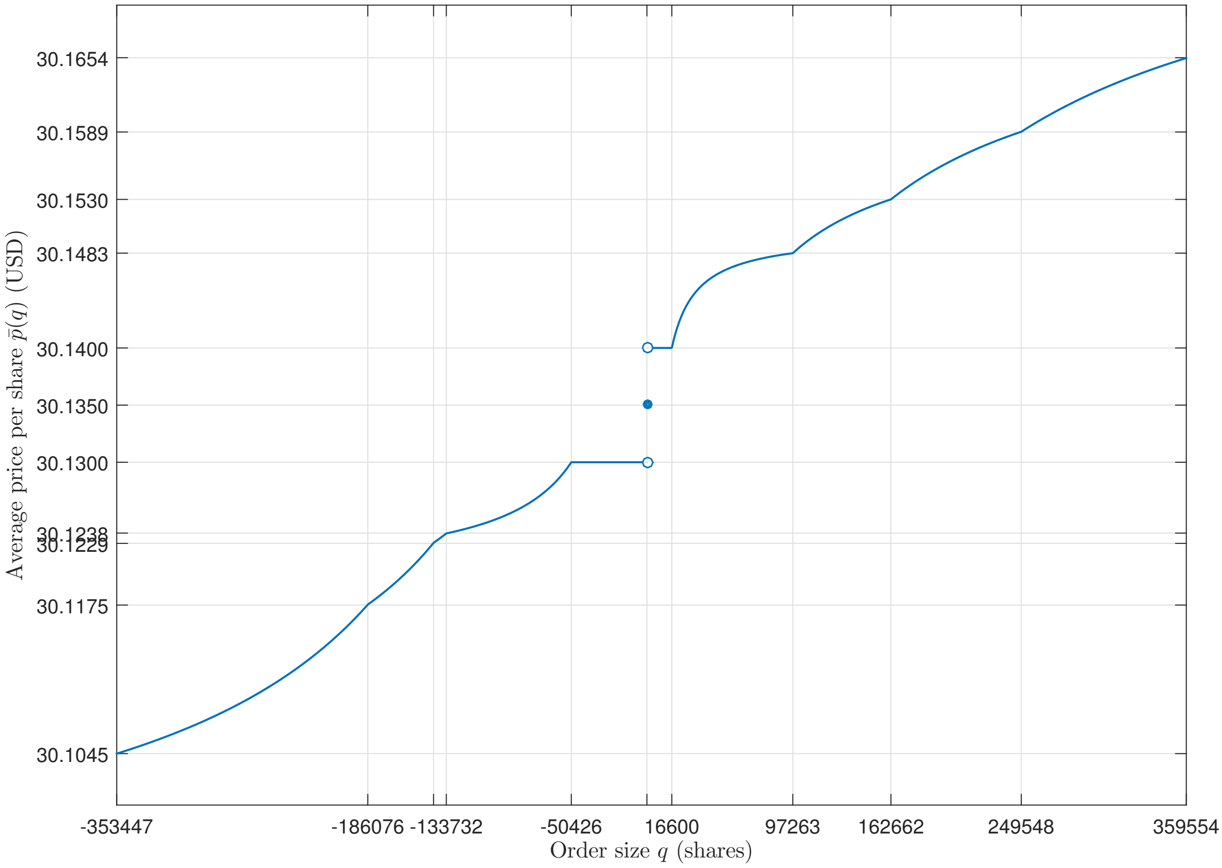


Figure 4. Average price per share $\bar{p}(q)$ for order size q from the LOB data in Table 1. Positive (negative) q means buying (selling) stocks.

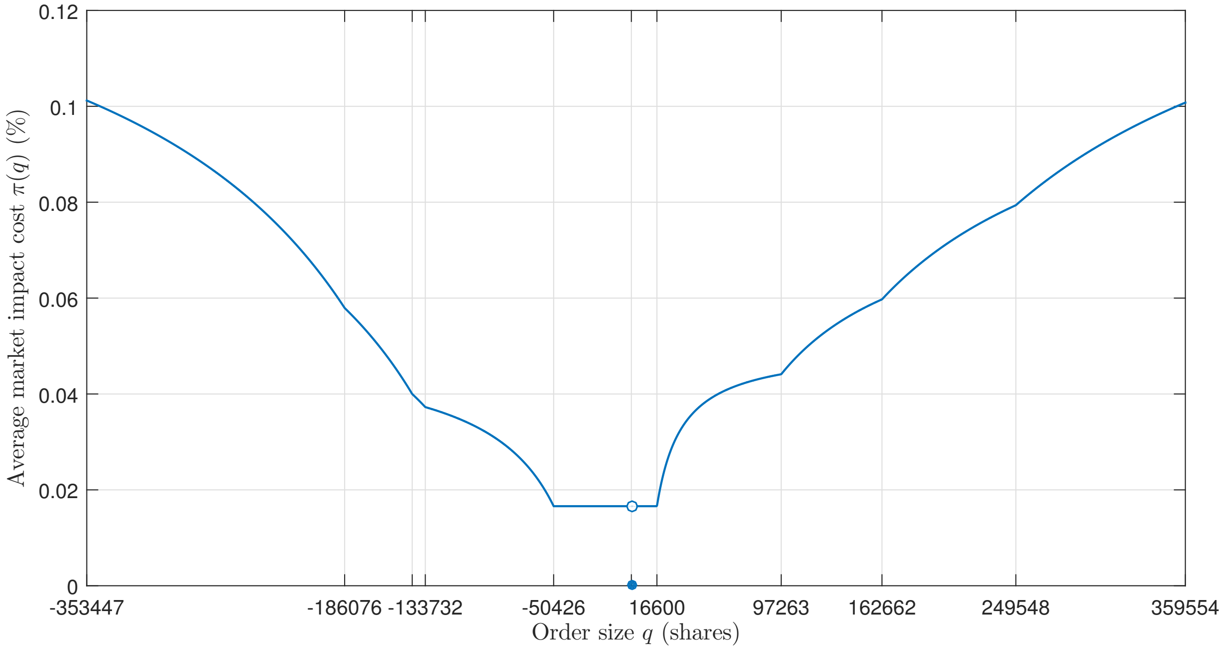


Figure 5. Average market impact cost $\pi(q)$ for order size q from the LOB data in Table 1. Positive (negative) q means buying (selling) stocks.

Chapter 2.3)

$$\begin{aligned}&\pi(q, M, P_1, P_2, \ldots, P_{-1}, P_{-2}, \ldots, V_1, V_2, \ldots, V_{-1}, V_{-2}, \ldots) \\ &\stackrel{\text{def}}{=} \frac{|\bar{p}(q, M, P_1, P_2, \ldots, P_{-1}, P_{-2}, \ldots, V_1, V_2, \ldots, V_{-1}, V_{-2}, \ldots) - M|}{M},\end{aligned} \tag{23}$$

where $M = \frac{P_{-1}+P_1}{2}$ is the midpoint between the best bid and ask price, called mid price. The average price per share for the order size q is defined as (Olsson 2005, Chapter 2.3)

$$\begin{aligned}&\bar{p}(q, M, P_1, P_2, \ldots, P_{-1}, P_{-2}, \ldots, V_1, V_2, \ldots, V_{-1}, V_{-2}, \ldots) \\ &\stackrel{\text{def}}{=} \begin{cases} \dfrac{\sum\limits_{i=k+1}^{-1} P_i V_i + P_k \left(q - \sum\limits_{i=k+1}^{-1} V_i \right)}{q}, & \text{if } q < V_{-1} \\ P_{-1}, & \text{if } V_{-1} \le q < 0 \\ M, & \text{if } q = 0 \\ P_1, & \text{if } 0 < q \le V_1 \\ \dfrac{\sum\limits_{i=1}^{k-1} P_i V_i + P_k \left(q - \sum\limits_{i=1}^{k-1} V_i \right)}{q}, & \text{if } V_1 < q \end{cases},\end{aligned} \tag{24}$$

where positive (negative) q means buying (selling) stocks, P_i and V_i with positive (negative) i are the quoted ask (bid) price and volume at level i, respectively (i.e. P_i and V_i correspond to the price and volume column of Table 1, respectively, and $V_i \ge 0, V_{-i} \le 0, \forall i \in \mathbb{Z}_{\ge 1}$), and the highest (lowest) trading level k when $q > V_1$ ($q < V_{-1}$) is

$$k = \left\{ x \in \mathbb{Z}_{\ge 2} \,\middle|\, \sum_{i=1}^{x-1} V_i < q \le \sum_{i=1}^{x} V_i \right\}, \tag{25a}$$

$$\left(k = \left\{ x \in \mathbb{Z}_{\le -2} \,\middle|\, \sum_{i=x}^{-1} V_i \le q < \sum_{i=x+1}^{-1} V_i \right\} \right). \tag{25b}$$

I.e. k represents the level in the order book where the q-th share would be executed (Figure 4 and 5 show $\bar{p}(q)$ and $\pi(q)$ from the LOB data in Table 1).

5. Transaction cost factor

The net wealth at the end of the n-th period N_n is defined as (Györfi and Vajda 2008)

$$N_n \stackrel{\text{def}}{=} S_n - C_n, \tag{26}$$

where S_n is the growth wealth at the end of the n-th period, and C_n is TC at the end of the n-th period. The current growth wealth S_n can be calculated from the previous net wealth N_{n-1}:

$$S_n = N_{n-1} \sum_{j=1}^{d} b_n^{(j)} x_n^{(j)} = N_{n-1} \langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle, \tag{27}$$

where $x_n^{(j)} = \frac{s_n^{(j)}}{s_{n-1}^{(j)}} = \frac{M_n^{(j)}}{M_{n-1}^{(j)}}$ (i.e. the price of asset j at the end of the n-th period $s_n^{(j)}$ is the mid price of asset j at the end of the n-th period $M_n^{(j)}$).

The transaction cost factor (TCF) at the end of the n-th period (i.e. the present moment; see Figure 1) is defined as (Györfi and Vajda 2008)

$$w_n \stackrel{\text{def}}{=} \frac{N_n}{S_n}, \tag{28}$$

where $w_n \in (0, 1]$ and $w_0 \stackrel{\text{def}}{=} 1$ (the subscript 0 denotes time 0; see Figure 1). Let us calculate w_n when rebalancing from the current portfolio vector $\boldsymbol{b}_n$ to the next portfolio vector $\boldsymbol{b}_{n+1}$ (see Figure 1) for the following three cases: Section 5.1 describes an existing TCF model, while Section 5.2 and 5.3 describe a new TCF model proposed in this chapter.

5.1. *Transaction cost factor with proportional costs but no market impact costs*

If there are no MICs, only mid prices are used to calculate TCF. Let c_p and c_s denote the rate of proportional TCs when purchasing and selling stocks, respectively, where $c_p, c_s \in [0, 1) : c_p + c_s > 0$. At the present moment (see Figure 1), there are $b_n^{(j)} x_n^{(j)} N_{n-1}$ dollars of asset j before rebalancing, by (27), while there are $b_{n+1}^{(j)} N_n$ dollars of asset j after rebalancing. If $b_{n+1}^{(j)} N_n > b_n^{(j)} x_n^{(j)} N_{n-1}$, then we have to purchase asset j for $b_{n+1}^{(j)} N_n - b_n^{(j)} x_n^{(j)} N_{n-1}$ dollars, and $c_p \left(b_{n+1}^{(j)} N_n - b_n^{(j)} x_n^{(j)} N_{n-1} \right)$ is the TC of asset j. On the other hand, if $b_n^{(j)} x_n^{(j)} N_{n-1} > b_{n+1}^{(j)} N_n$, then we have to sell asset j for $b_n^{(j)} x_n^{(j)} N_{n-1} - b_{n+1}^{(j)} N_n$ dollars, and $c_s \left(b_n^{(j)} x_n^{(j)} N_{n-1} - b_{n+1}^{(j)} N_n \right)$ is TC of asset j.

The growth wealth S_n consists of the sum of the net wealth N_n and the TCs of all assets in the following self-financing way (Györfi and Vajda 2008):

$$S_n = N_n + c_p \sum_{j=1}^{d} \left(b_{n+1}^{(j)} N_n - b_n^{(j)} x_n^{(j)} N_{n-1} \right)^+ + c_s \sum_{j=1}^{d} \left(b_n^{(j)} x_n^{(j)} N_{n-1} - b_{n+1}^{(j)} N_n \right)^+, \tag{29}$$

where $x^+ \stackrel{\text{def}}{=} \max(0, x)$. By dividing both sides of (29) by S_n and by referring to (27) and (28), Equation (29) can be rewritten as (Györfi and Vajda 2008)

$$1 = w_n + c_p \sum_{j=1}^{d} \left(b_{n+1}^{(j)} w_n - \frac{b_n^{(j)} x_n^{(j)}}{\langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle} \right)^+ + c_s \sum_{j=1}^{d} \left(\frac{b_n^{(j)} x_n^{(j)}}{\langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle} - b_{n+1}^{(j)} w_n \right)^+. \tag{30}$$

Additionally, by using the property of $(a - b)^+ = a - b + (b - a)^+$, Equation (30) can be simplified as (Ormos and Urbán 2013)

$$w_n = 1 - \frac{c_p + c_s}{1 + c_p} \sum_{j=1}^{d} \left(\frac{b_n^{(j)} x_n^{(j)}}{\langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle} - b_{n+1}^{(j)} w_n \right)^+, \tag{31}$$

which is solvable by using a root-finding algorithm, where $w_n = w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_n)$ is an unknown variable (c_p and c_s are omitted for notational simplicity).[1]

TCF w_n in (31) is a quasi-concave but not concave function of $\boldsymbol{b}_{n+1} \in \Delta^{d-1}$ (proof is in Appendix C) as shown in Figure 6(d) and 6(f), where its convexity depends on the TC rates, c_p and c_s: i.e. the greater $\frac{c_p+c_s}{1+c_p}$; the greater $\frac{\partial^2 w_n}{\left(\partial b_{n+1}^{(j)}\right)^2}$, by (C8). Meanwhile, 1D plots in 6(b) look like piecewise linear functions since $\frac{\partial^2 w_n}{\left(\partial b_{n+1}^{(j)}\right)^2} \approx 0$ when $\frac{c_p+c_s}{1+c_p}$ is tiny.

LEMMA 5.1 *If $g : \mathbb{R}^d \to \mathbb{R}$ is quasi-concave (quasi-convex) and $h : \mathbb{R} \to \mathbb{R}$ is nondecreasing, then $f = h \circ g$ is quasi-concave (quasi-convex) (Boyd and Vandenberghe 2004, p. 102).*

LEMMA 5.2 *The sum of quasi-concave (quasi-convex) functions is not necessarily quasi-concave (quasi-convex) (Sydsæter et al. 2010, p. 95).*

THEOREM 5.3 *$\sum_{i\in J_n} (\ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle + \ln w_n)$ in (16) is not necessarily a quasi-concave function of $\boldsymbol{b} \in \Delta^{d-1}$ even if w_n is quasi-concave.*

Proof. $\sum_{i\in J_n} \ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle$ is quasi-concave both i) by the proof in Appendix B that $\sum_{i\in J_n} \ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle$ is concave and ii) by the fact that every concave (convex) function is quasi-concave (quasi-convex). In addition, $\ln w_n$ is quasi-concave but not concave because w_n is quasi-concave but not concave by lemma 5.1. As a result, $\sum_{i\in J_n} (\ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle + \ln w_n) = \sum_{i\in J_n} \ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle + |J_n| \ln w_n$ in (16) is not a necessarily quasi-concave function of $\boldsymbol{b} \in \Delta^{d-1}$ by lemma 5.2. □

Therefore, a local solution of (16) with TCF w_n in (31) is not guaranteed to be a global solution. Györfi and Vajda (2008, p. 112) also mentioned that no global optimality of (16) is guaranteed although they did not give any proof.

5.2. *Transaction cost factor with market impact costs but no proportional costs*

This subsection is to propose a new TCF model by considering market impact costs but not considering proportional costs: if there are no proportional costs, the growth wealth S_n consists of the sum of the net wealth N_n and the MICs:

$$S_n = N_n + \sum_{j=1}^{d} \left(\bar{p}\left(q_n^{(j)}\right) - M_n^{(j)}\right) q_n^{(j)}, \tag{33}$$

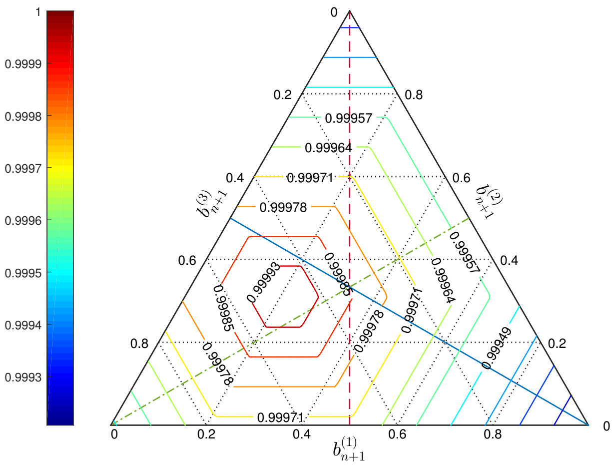
where $\bar{p}(\cdot)$ is the average price function in (24) (for notational simplicity, some input variables of (24), i.e. $M, P_1, P_2, \ldots, P_{-1}, P_{-2}, \ldots, V_1, V_2, \ldots, V_{-1}, V_{-2}, \ldots$ of each asset j, are omitted in (33) and the subsequent expressions), $M_n^{(j)}$ is the mid price of asset j, $q_n^{(j)}$ is an unknown order size of asset j, and subscript n denotes the end of the n-th period.

When the investor either buys or sells every asset j according to the portfolio vector of the

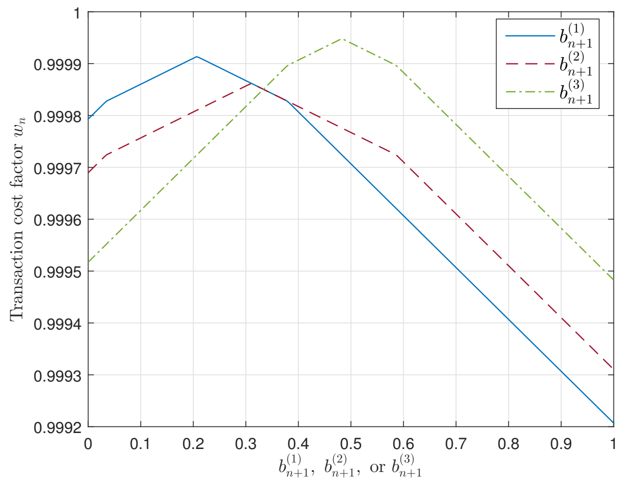
[1] Another form of TCF is defined as (Borodin and El-Yaniv 1998, pp. 299–300)

$$w_n = 1 - \frac{c_p + c_s}{1 + c_p} \sum_{j=1}^{d} \left(\frac{b_n^{(j)} x_n^{(j)}}{\langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle} - b_{n+1}^{(j)} \right)^+, \tag{32}$$

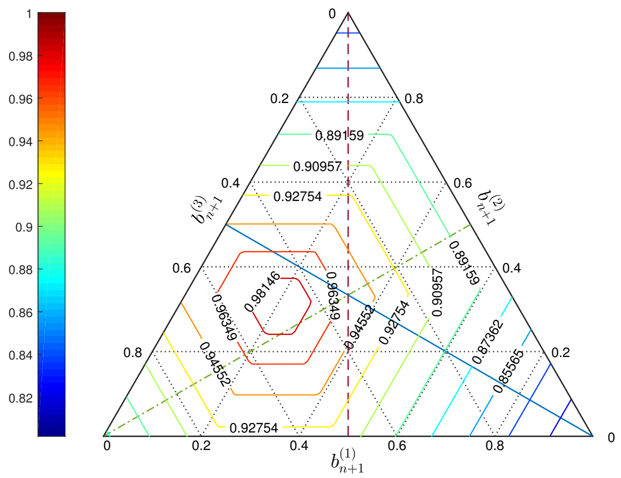
which has less computational burden than (31), but it is error-prone if $\frac{c_p+c_s}{1+c_p}$ is large. Meanwhile, Borodin and El-Yaniv (1998, pp. 299–300) and Borodin *et al.* (2004, pp. 590–591) made the same typo of substituting $w_n = 1 - \frac{c_p + c_s}{1 + c_p} \sum_{j=1}^{d} \left(\frac{b_n^{(j)} x_n^{(j)}}{\langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle} - b_n^{(j)} \right)^+$ for (32).



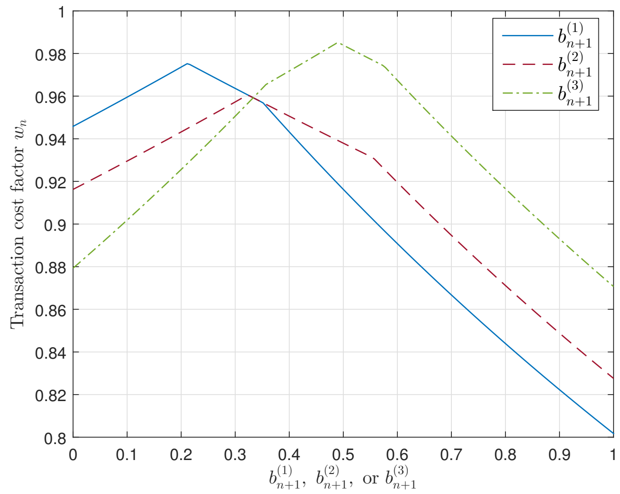
(a) Ternary contour plot of transaction cost factor when $c_s = 0.1\%$.



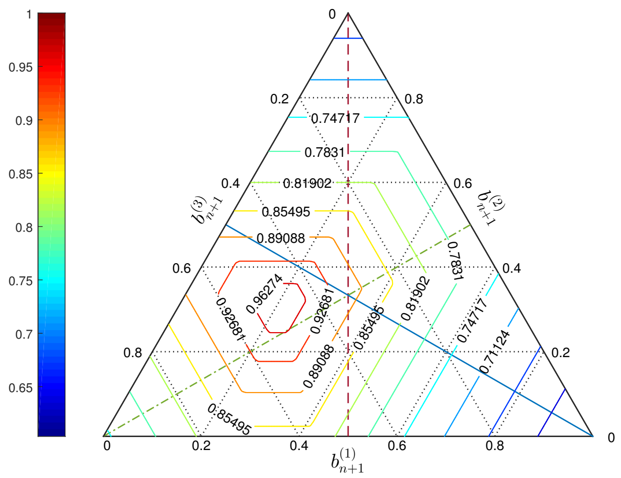
(b) 1D plots of transaction cost factor when $c_s = 0.1\%$ (each line matches each straight line in the left plot).



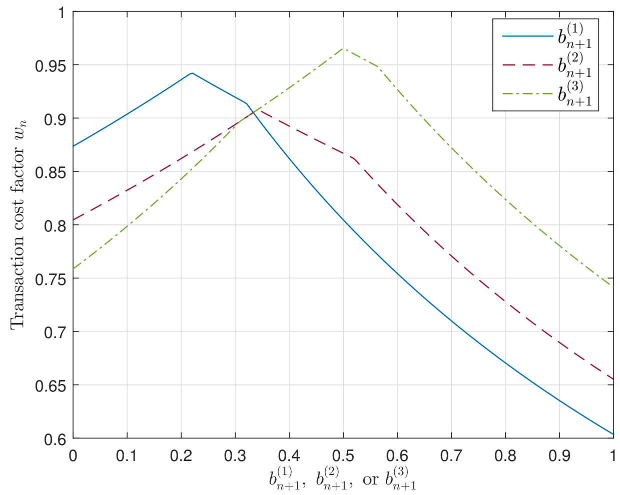
(c) Ternary contour plot of transaction cost factor when $c_s = 25\%$.



(d) 1D plots of transaction cost factor when $c_s = 25\%$ (each line matches each straight line in the left plot).



(e) Ternary contour plot of transaction cost factor when $c_s = 50\%$.



(f) 1D plots of transaction cost factor when $c_s = 50\%$ (each line matches each straight line in the left plot).

Figure 6. Ternary contour plots and 1D plots of transaction cost factor $w_n = w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_n)$ in (31) with the variable $\boldsymbol{b}_{n+1}$ and the fixed values: $\boldsymbol{b}_n = [1/3\ 1/3\ 1/3]^{\mathrm{T}}$, $\boldsymbol{x}_n = [0.6\ 0.9\ 1.4]^{\mathrm{T}}$, and $c_p = 0$.

$(n+1)$-th period $\boldsymbol{b}_{n+1}$, the following equation is satisfied:

$$b_{n+1}^{(j)} N_n - b_n^{(j)} x_n^{(j)} N_{n-1} = M_n^{(j)} q_n^{(j)}, \tag{34}$$

where $b_n^{(j)} x_n^{(j)} N_{n-1}$ is the amount of dollars of asset j before rebalancing a portfolio, and $b_{n+1}^{(j)} N_n$ is that after rebalancing. Hence, equation (34) means that the purchase (if left-hand side of (34) is positive) or sale (if left-hand side of (34) is negative) amount in dollars of asset j equals the product between $M_n^{(j)}$, the mid price of asset j, and $q_n^{(j)}$, an unknown order size of asset j, where positive (negative) $q_n^{(j)}$ means the purchase (sale) of asset j.[1]

Equation (33) can be rewritten, by (28) and (34), as

$$w_n = 1 - \frac{\sum_{j=1}^{d} \left(\bar{p}\left(q_n^{(j)}\right) - M_n^{(j)} \right) q_n^{(j)}}{S_n}, \tag{35}$$

where

$$q_n^{(j)} = \frac{b_{n+1}^{(j)} S_n w_n - b_n^{(j)} x_n^{(j)} N_{n-1}}{M_n^{(j)}}. \tag{36}$$

Equations (35) and (36) are solvable by using a root-finding algorithm, where $w_n = w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_n, N_{n-1})$ is an unknown variable. TCF w_n in (35) is calculated in the following order: i) $N_0 = S_0$, ii) S_1 by using (27), iii) w_1 by using (35), iv) $N_1 = w_1 S_1$, v) S_2 by using (27), and vi) continued calculations.

TCF w_n in (35) looks like a piecewise linear concave function as shown in Figure 7(b), 7(d), and 7(f).[2] Also, the greater net wealth at the end of the previous period N_{n-1}; the less w_n and more non-differentiable points in the 1D plots. This is because $|k|$ in (25), the absolute value of the executed level in LOB, increases as N_{n-1} increases.

5.3. *Transaction cost factor with both proportional and market impact costs*

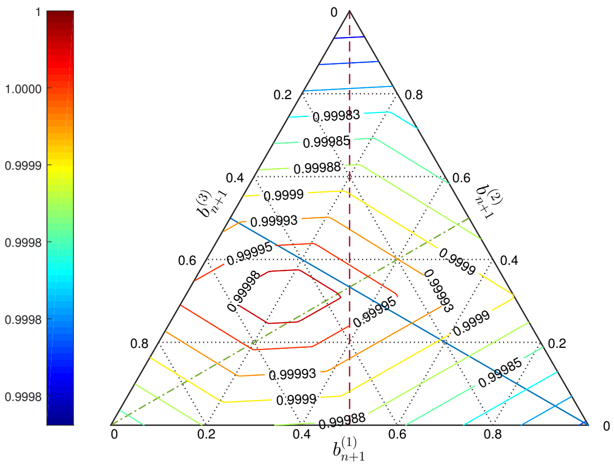
If both proportional TCs and MICs are considered, the growth wealth S_n consists of the sum of the net wealth N_n, the MICs in (33), and the proportional TCs for buying and selling assets:

$$S_n = N_n + \sum_{j=1}^{d} \left(\bar{p}\left(q_n^{(j)}\right) - M_n^{(j)} \right) q_n^{(j)} + c_p \sum_{j=1}^{d} \left(\bar{p}\left(q_n^{(j)}\right) q_n^{(j)} \right)^+ + c_s \sum_{j=1}^{d} \left(-\bar{p}\left(q_n^{(j)}\right) q_n^{(j)} \right)^+ .^{3} \tag{37}$$

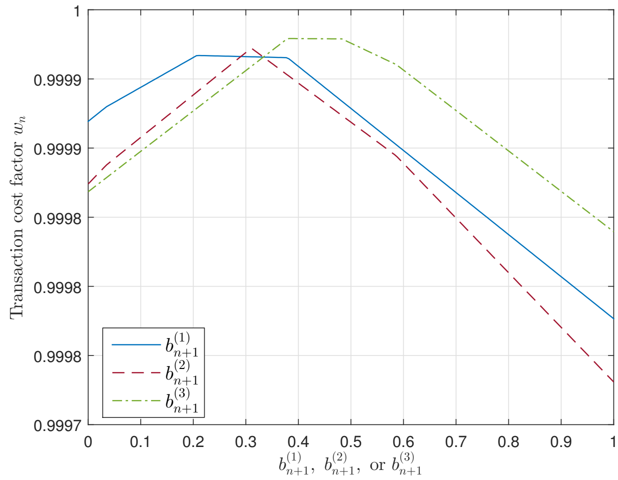
[1] The reason why $M_n^{(j)}$ in (34) cannot be replaced with $\bar{p}\left(q_n^{(j)}\right)$ is that $b_{n+1}^{(j)} N_n$ denotes the asset j's portion of the net wealth, excluding MICs, at the end of the n-th period. Also, $M_n^{(j)}$ excludes MIC, while $\bar{p}\left(q_n^{(j)}\right)$ includes MIC.

[2] Mathematical proof of the convexity of w_n in (35) is not given in this chapter due to the complex equations: (24), (35), and (36). However, numerically approximate values of second-order partial derivative at differentiable points of the nine 1D plots in Figure 7(b), 7(d), and 7(f) are all positive when using the central difference $\frac{\partial^2 f(x)}{\partial x^2} \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$. This implies that w_n in (35) is not concave.

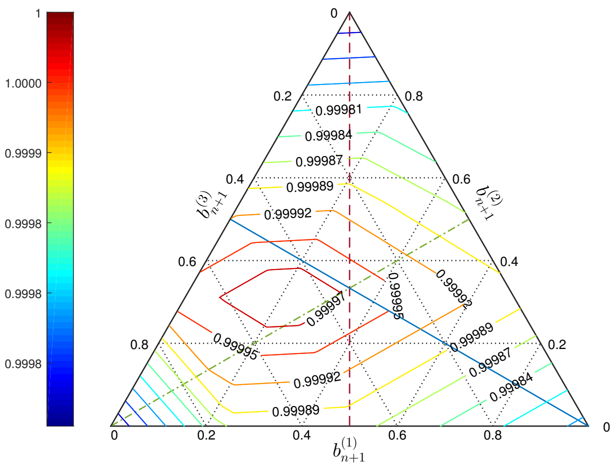
[3] The reason why the proportional TCs in (37) cannot be replaced with $c_p \sum_{j=1}^{d} \left(M_n^{(j)} q_n^{(j)} \right)^+ + c_s \sum_{j=1}^{d} \left(-M_n^{(j)} q_n^{(j)} \right)^+$ is that proportional TCs are determined by the actual traded price $\bar{p}\left(q_n^{(j)}\right)$, not by the mid price $M_n^{(j)}$.



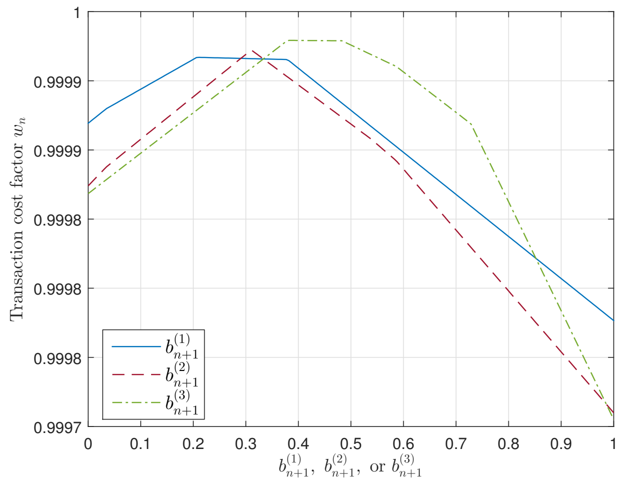
(a) Ternary contour plot of transaction cost factor when N_{n-1} is USD 10^4.



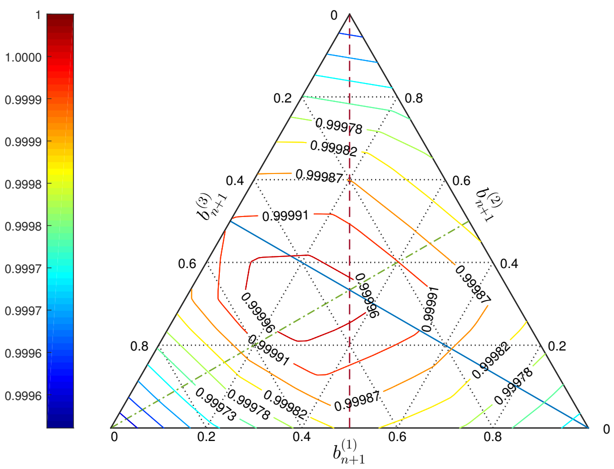
(b) 1D plots of transaction cost factor when N_{n-1} is USD 10^4 (each line matches each straight line in the left plot).



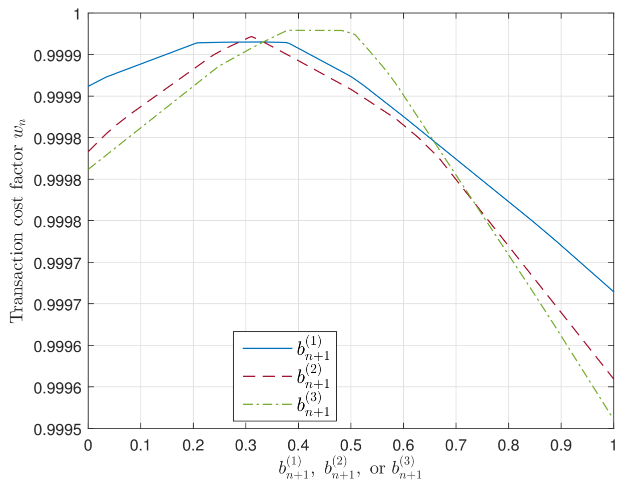
(c) Ternary contour plot of transaction cost factor when N_{n-1} is USD 10^5.



(d) 1D plots of transaction cost factor when N_{n-1} is USD 10^5 (each line matches each straight line in the left plot).



(e) Ternary contour plot of transaction cost factor when N_{n-1} is USD 10^6.



(f) 1D plots of transaction cost factor when N_{n-1} is USD 10^6 (each line matches each straight line in the left plot).

Figure 7. Ternary contour plots and 1D plots of transaction cost factor $w_n = w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_n, N_{n-1})$ in (35) with the variable $\boldsymbol{b}_{n+1}$ and the fixed values: $\boldsymbol{b}_n = [1/3\ 1/3\ 1/3]^{\mathrm{T}}$ and $\boldsymbol{x}_n = [0.6\ 0.9\ 1.4]^{\mathrm{T}}$. 10-level limit order book data of AAPL ($b^{(1)}$), AMZN ($b^{(2)}$), and GOOG ($b^{(3)}$) on 21 Jun 2012 at 16:00:00 was used.

This can be simplified, by the property of $a^+ = a + (-a)^+$, as

$$S_n = N_n + \sum_{j=1}^{d} \left((1-c_s)\bar{p}\left(q_n^{(j)}\right) - M_n^{(j)} \right) q_n^{(j)} + (c_p + c_s) \sum_{j=1}^{d} \left(\bar{p}\left(q_n^{(j)}\right) q_n^{(j)} \right)^+, \tag{38}$$

and can be rewritten, by (28), as

$$w_n = 1 - \frac{\sum_{j=1}^{d} \left((1-c_s)\bar{p}\left(q_n^{(j)}\right) - M_n^{(j)} \right) q_n^{(j)} + (c_p + c_s) \sum_{j=1}^{d} \left(\bar{p}\left(q_n^{(j)}\right) q_n^{(j)} \right)^+}{S_n}, \tag{39}$$

where $q_n^{(j)}$ is the same as (36). Equation (39) and (36) are also solvable by using a root-finding algorithm, where $w_n = w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_n, N_{n-1})$ is an unknown variable (c_p and c_s are omitted for notational simplicity).

The ternary contour plots and 1D plots of w_n in (39) are different from those of w_n in (31) when comparing between Figure {8(a), 8(b)} and Figure {6(a), 6(b)} due to the additional consideration of MICs. In contrast, they are similar when comparing between Figure {8(c), 8(d), 8(e), 8(f)} and Figure {6(c), 6(d), 6(e), 6(f)} because proportional TCs are much greater than MICs when c_p or c_s is large.

THEOREM 5.4 *The quasi-concavity of function $f : \mathbb{R}^n \to \mathbb{R}$ is equivalent to the fact that f is unimodal (i.e. single-peaked) (Simchi-Levi et al. 2014, p. 18).*

TCF w_n in (39) is a unimodal function of $\boldsymbol{b}_{n+1}$ (TCF w_n in (31) and that in (35) are also unimodal), as shown in Figure 8(a), 8(c), and 8(e), because w_n strictly decreases as $\boldsymbol{b}_{n+1}$ goes away from the maximum point:

$$\boldsymbol{b}^\star_{n+1} \stackrel{\text{def}}{=} \underset{\boldsymbol{b} \in \Delta^{d-1}}{\arg\max}\, w(\boldsymbol{b}_n, \boldsymbol{b}, \boldsymbol{x}_n, N_{n-1}) = \frac{\boldsymbol{b}_n \odot \boldsymbol{x}_n}{\langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle}, \tag{40}$$

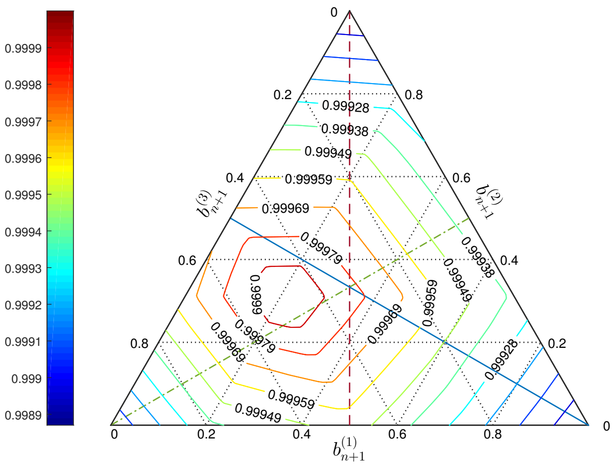
where $\odot$ denotes element-wise multiplication of vectors.[1] Hence, w_n in (39) is quasi-concave by theorem 5.4,[2] but $\sum_{i \in J_n} (\ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1} \rangle + \ln w_n)$ in (16) with w_n in (39) is not a necessarily quasi-concave function of $\boldsymbol{b} \in \Delta^{d-1}$ by theorem 5.3.

6. Log-optimal portfolio

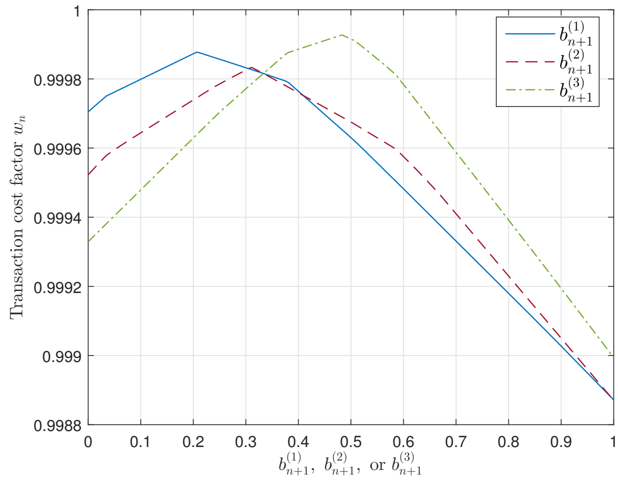
The log-optimal portfolio (or the log-utility approach) is one of OPS strategies and is the basis of the proposed method described in Section 7.

[1] The mathematical proof of the unimodality is not provided in this chapter.

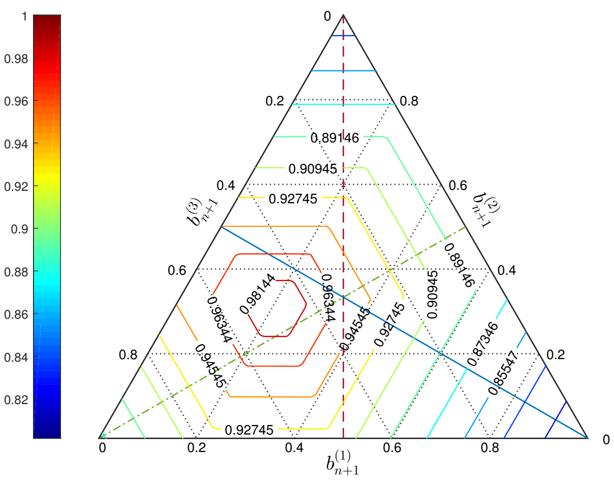
[2] Mathematical proof of the convexity of w_n in (39) is not given in this chapter due to the complex equations: (24), (25), (39), and (36). However, numerically approximate values of second-order partial derivative at differentiable points of the nine 1D plots in Figure 8(b), 8(d), and 8(f) are all positive when using the central difference $\frac{\partial^2 f(x)}{\partial x^2} \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$. This implies that w_n in (39) is not concave.



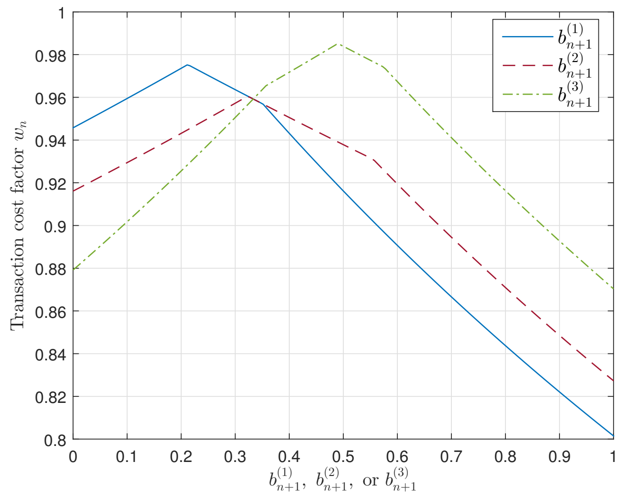
(a) Ternary contour plot of transaction cost factor when $c_s = 0.1\%$.



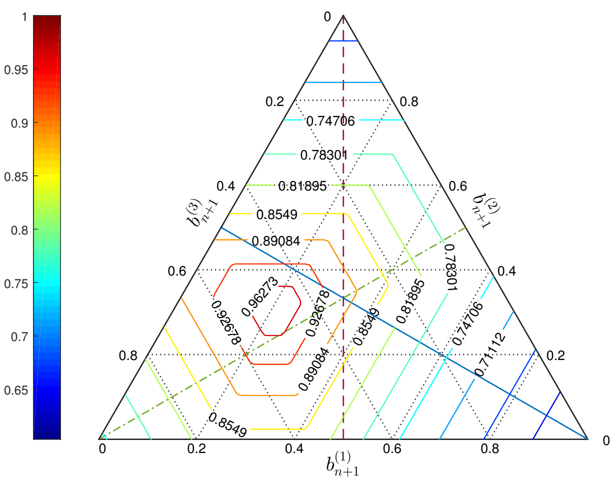
(b) 1D plots of transaction cost factor when $c_s = 0.1\%$ (each line matches each straight line in the left plot).



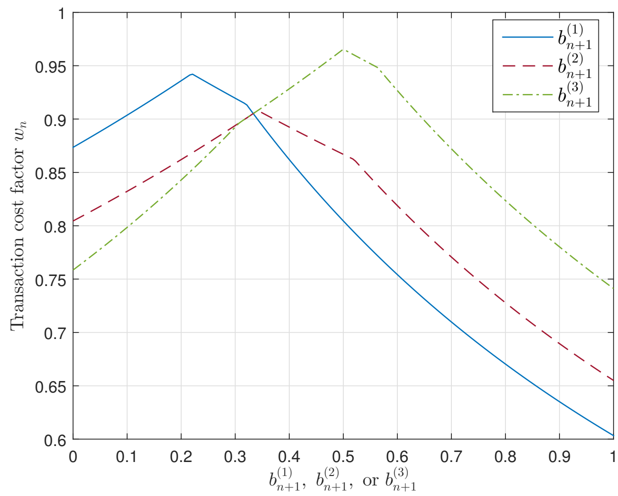
(c) Ternary contour plot of transaction cost factor when $c_s = 25\%$.



(d) 1D plots of transaction cost factor when $c_s = 25\%$ (each line matches each straight line in the left plot).



(e) Ternary contour plot of transaction cost factor when $c_s = 50\%$.



(f) 1D plots of transaction cost factor when $c_s = 50\%$ (each line matches each straight line in the left plot).

Figure 8. Ternary contour plots and 1D plots of transaction cost factor $w_n = w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_n, N_{n-1})$ in (39) with the variable $\boldsymbol{b}_{n+1}$ and the fixed values: $\boldsymbol{b}_n = [1/3\ 1/3\ 1/3]^{\mathrm{T}}$, $\boldsymbol{x}_n = [0.6\ 0.9\ 1.4]^{\mathrm{T}}$, $c_p = 0$, and N_{n-1} is USD 10^6. 10-level limit order book data of AAPL ($b^{(1)}$), AMZN ($b^{(2)}$), and GOOG ($b^{(3)}$) on 21 Jun 2012 at 16:00:00 was used.

6.1. *Objective of log-optimal investment strategy*[1]

Assume that the present moment is time 0 (do not see Figure 1). The wealth of a portfolio of a single asset at the end of the n-th period (in the future) with an initial wealth S_0 is

$$S_n = S_0 \prod_{i=1}^{n} X_i, \tag{41}$$

where X_i is a random variable of the price relative at the end of the i-th period (i.e. $X_i = \frac{S_i}{S_{i-1}}$, where S_i is a random variable of an asset's price at the end of the i-th period). Taking the log of both the sides gives

$$\ln S_n = \ln S_0 + \sum_{i=1}^{n} \ln X_i, \tag{42}$$

and this can be rewritten as

$$\ln \left(\frac{S_n}{S_0} \right)^{1/n} = \frac{1}{n} \sum_{i=1}^{n} \ln X_i. \tag{43}$$

As n becomes extremely large, the right-hand side of (43) converges to

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \ln X_i = \mathbb{E}[\ln X_i], \tag{44}$$

under the assumption that the random variable X_i is independent and identically distributed, and the wealth S_n converges to

$$\lim_{n \to \infty} S_n = S_0 e^{mn}, \tag{45}$$

where $m \stackrel{\text{def}}{=} \mathbb{E}[\ln X_i]$ is a growth rate. Consequently, for large n (i.e. long-term investment), the wealth S_n grows (roughly) exponentially with mn; therefore, maximising the growth rate m is critical for the long-term investment. In other words, the log is an appropriate utility function for the long-term investment.

6.2. *log-optimal portfolio with transaction costs*

The growth wealth at the end of the next period (i.e. $n+1$; see Figure 1), from (27) and (28), is

$$S_{n+1} = w_n S_n \langle \boldsymbol{b}_{n+1}, \boldsymbol{X}_{n+1} \rangle, \tag{46}$$

where $w_n = w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_n)$ is TCF in (31), the deterministic value S_n is the growth wealth at the present moment, the d-dimensional variable $\boldsymbol{b}_{n+1}$ is the portfolio vector of the next period, and the d-dimensional multivariate random variable $\boldsymbol{X}_{n+1}$ is a set of possible market vectors in the next period.

Our aim is to choose an appropriate portfolio vector $\boldsymbol{b}_{n+1}$ to maximise the conditional expected value of the log (the log behaves as a utility function for the long-term investment as mentioned in

[1] The whole of this subsection is a paraphrase of (Luenberger 1998, pp. 419–421).

Section 6.1) of the growth wealth at the end of the next period, given the observed market vectors $\boldsymbol{x}_n, \boldsymbol{x}_{n-1}, \dots, \boldsymbol{x}_1$:

$$\begin{aligned}\boldsymbol{b}_{n+1} &= \underset{\boldsymbol{b}\in\Delta^{d-1}}{\arg\max}\, \mathbb{E}[\ln S_{n+1}] \\ &= \underset{\boldsymbol{b}\in\Delta^{d-1}}{\arg\max}\, \mathbb{E}[\ln \left(w(\boldsymbol{b}_n, \boldsymbol{b}, \boldsymbol{x}_n) S_n \langle \boldsymbol{b}, \boldsymbol{X}_{n+1}\rangle\right)] \\ &= \underset{\boldsymbol{b}\in\Delta^{d-1}}{\arg\max} \Big(\ln w(\boldsymbol{b}_n, \boldsymbol{b}, \boldsymbol{x}_n) + \mathbb{E}[\ln\langle \boldsymbol{b}, \boldsymbol{X}_{n+1}\rangle | \boldsymbol{X}_n = \boldsymbol{x}_n, \boldsymbol{X}_{n-1} = \boldsymbol{x}_{n-1}, \dots, \boldsymbol{X}_1 = \boldsymbol{x}_1] \Big),\end{aligned} \tag{47}$$

and this can be rewritten as

$$\boldsymbol{b}_{n+1} = \underset{\boldsymbol{b}\in\Delta^{d-1}}{\arg\max} \Big(\ln w(\boldsymbol{b}_n, \boldsymbol{b}, \boldsymbol{x}_n) + \mathbb{E}[\ln\langle \boldsymbol{b}, \boldsymbol{X}_{n+1}\rangle | \boldsymbol{X}_n = \boldsymbol{x}_n] \Big), \tag{48}$$

under the assumption that the market process $\{\boldsymbol{X}_i\}$ is a homogeneous first-order Markov process. This result is the same as (Györfi and Vajda 2008, equation (5)) although they derived (48) in a different way. Finally, the implementable algorithm to obtain $\boldsymbol{b}_{n+1}$ in (48) employs Equation (16) and (17).

7. Proposed method of online portfolio selection

7.1. *Assumptions for simplicity*

I assume i) that assets are arbitrarily divisible (i.e. $q_n^{(j)} \in \mathbb{R}, \forall j, n \geq 1$) to avoid mixed-integer nonlinear programming[1] and ii) that the market impact of trading by the proposed OPS method is transitory, not persistent: i.e. portfolio rebalancing at the present moment does not affect the price at the end of the next period s_{n+1} (see Figure 1).[2]

Only market orders, an order to buy or sell a specific number of shares q at the best price available when an investor places his or her order, are submitted when the proposed strategy rebalances a portfolio in order to avoid the risk of non-execution. In addition, the proposed method does not split a large market order into smaller market orders in order to avoid potential liquidity risk in the future:[3] i.e. it rebalances a portfolio by using current LOBs (which are obvious) rather than by using LOBs in the future (which are unknown).

The proposed method ignores hidden limit orders (HLOs), invisible in limit order books. If a market order is executed against a hidden order, the trader submitting the market order may receive an unexpected price improvement (Bauwens *et al.* 2007, p. 115): i.e. if HLOs are in a limit order book, a trader may buy stocks at a cheaper price than expected, and he or she may sell stocks at a superior price than expected. In other words, $\tilde{\pi}(q) \leq \pi(q), \forall q \in \mathbb{R}$, where $\tilde{\pi}(q)$ is the average market impact cost with HLOs for order size q, and $\pi(q)$ is that without HLOs in (23). The additional liquidity from HLOs can be quantified by hidden volume rate, the total volume of trades against hidden orders divided by the total volume of all trades. The mean of this value between 2 Jan 2014 and 31 Dec 2015 is {13.30%, 13.33%} in the case of stocks, not exchange traded products, traded on {NYSE, NASDAQ}, respectively.[4] Therefore, the difference between $\tilde{\pi}(q)$ and $\pi(q)$ may not be significant.

[1] Softwares of mixed-integer nonlinear programming are listed in (Bussieck and Vigerske 2011).

[2] Dynamic trading strategy by Gârleanu and Pedersen (2013) allows for both the cases: transitory and persistent market impact.

[3] Expected overall market impact costs can be minimized by the split of a large market order into a number of smaller consecutive market orders under a specified resilience rate (Alfonsi *et al.* 2008).

[4] This value is the average of daily hidden volume rates, downloaded from the homepage of U.S. Securities and Exchange Commission (URL: http://www.sec.gov/opa/data/market-structure/marketstructuredata-by-exchange.html).

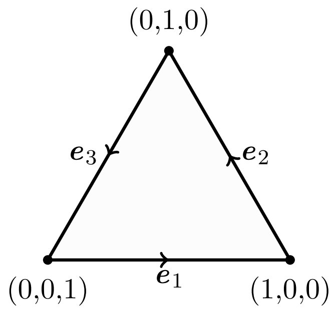


Figure 9. Ternary axes. Each vertex has the barycentric coordinate of $\boldsymbol{b} \in \Delta^2$, and each unit vector $\boldsymbol{e}_j$ corresponds to the direction of the partial derivative with respect to $b^{(j)}$.

Computation time to calculate $\boldsymbol{b}_{n+1}$ between receiving real-time data of LOBs from a stock exchange and sending market orders to rebalance a portfolio is ignored. To be specific, the price and volume in LOBs change during the computation as LOBs are continuously updated by other investors. Therefore, the proposed method actually uses the past LOBs, not the present, to calculate $\boldsymbol{b}_{n+1}$. In other words, it employs TCF w_n in (39) delayed for the computation time.

7.2. *Details of the proposed method*

The proposed method is based on the log-optimal portfolio with TCs in (48), as the same as (Györfi and Vajda 2008). The difference between the existing (Györfi and Vajda 2008) and proposed method is w_n. I.e. $w_n = w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_n, N_{n-1})$ in (39) is employed for the proposed method, whereas $w_n = w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_n)$ in (31) is employed for the existing method. As a result, the portfolio vector of the next period $\boldsymbol{b}_{n+1}$ of the proposed method depends on the net wealth of the previous period N_{n-1}, while that of the existing method does not (the proposed method has less tendency of portfolio rebalancing as N_{n-1} increases, while the existing method does not).

Consequently, the proposed method uses the following equations: (15), (16), (17), (24), (25), (36), and (39), where (16) is a constrained (i.e. $\boldsymbol{b}_{n+1} \in \Delta^{d-1}$) nonlinear optimisation problem (Algorithm 1 describes the difference between the existing and proposed method). However, Equation (16) is not a quasi-convex optimisation problem for both the cases, (31) and (39), which means a local solution is not guaranteed to be a global solution, as mentioned in Section 5.1 and 5.3.

7.3. *Local vs. global optimisation*[1]

Let us find which is the appropriate optimisation algorithm between local and global optimisation to maximise $\sum_{i \in J_n} (\ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle + \ln w_n)$ in (16) with respect to $\boldsymbol{b} \in \Delta^{d-1}$. The summation part is equal to the sum of the following two equations:

$$f(\boldsymbol{b}) = \sum_{i \in J_n} \ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle, \tag{49}$$

and

$$g(\boldsymbol{b}) = |J_n| \ln w_n, \tag{50}$$

where w_n is either $w(\boldsymbol{b}_n, \boldsymbol{b}, \boldsymbol{x}_n)$ in (31) or $w(\boldsymbol{b}_n, \boldsymbol{b}, \boldsymbol{x}_n, N_{n-1})$ in (39), in the case of $|J_n| \geq 1$ (if $|J_n| = \varnothing$, the optimisation is not performed as the nine and tenth lines of Algorithm 1).

[1] Neither Györfi and Vajda (2008) nor Ormos and Urbán (2013) explained how to solve the optimisation problem of (16).

Algorithm 1: Difference between the existing (Györfi and Vajda 2008) and proposed method.

Input: c_p, c_s, S_0, $\boldsymbol{b}_1$, and limit order book data between time 0 and the T-th period, where T is the last period of trading.

Output: terminal wealth S_T.

`// Initialisation`

$L \leftarrow 5$ `// The number of experts` L `is less than 10, specified in (Györfi and Vajda 2008, p. 112), in order to reduce the computation time.`

$N_0 \leftarrow S_0$

`// A loop to update` $\boldsymbol{b}_{n+1}$

for $n \leftarrow 1$ *to* T **do**

    for $j \leftarrow 1$ *to* d **do**

        $x_n^{(j)} \leftarrow \frac{M_n^{(j)}}{M_{n-1}^{(j)}}$ `// where` $M_n^{(j)}$ `is the mid price of asset` j `at the end of the` n`-th period.`

    end

    for $l \leftarrow 1$ *to* L **do**

        $J_n = \left\{ 1 \le i \le n-1 \;\middle|\; \|\boldsymbol{x}_i - \boldsymbol{x}_n\| \le \sqrt{0.001 d \frac{l}{L}} \right\}$

        if $J_n = \varnothing$ **then**

            $\boldsymbol{h}^{(l)} \leftarrow \boldsymbol{b}_{n+1}^{\star}$ `//` $\boldsymbol{b}_{n+1}^{\star}$ `in (40) is employed instead of` $\boldsymbol{b}_1$ `in order not to suffer large transaction costs.`

        else

            switch method **do**

                case existing **do**

                    $\boldsymbol{h}^{(l)} \leftarrow \mathop{\arg\max}_{\boldsymbol{b} \in \Delta^{d-1}} \sum_{i \in J_n} \left(\ln \langle \boldsymbol{b}, \boldsymbol{x}_{i+1} \rangle + \ln w(\boldsymbol{b}_n, \boldsymbol{b}, \boldsymbol{x}_n) \right)$ `// by using (31).`

                case proposed **do**

                    $\boldsymbol{h}^{(l)} \leftarrow \mathop{\arg\max}_{\boldsymbol{b} \in \Delta^{d-1}} \sum_{i \in J_n} \left(\ln \langle \boldsymbol{b}, \boldsymbol{x}_{i+1} \rangle + \ln w(\boldsymbol{b}_n, \boldsymbol{b}, \boldsymbol{x}_n, N_{n-1}) \right)$ `// by using (39).`

            end

        end

        $S_{\text{past}}^{(l)} \leftarrow S_0 \prod_{i=1}^{n} \left\langle \boldsymbol{h}^{(l)}, \boldsymbol{x}_i \right\rangle$

    end

    $\boldsymbol{b}_{n+1} \leftarrow \frac{\sum_{l=1}^{L} \boldsymbol{h}^{(l)} S_{\text{past}}^{(l)}}{\sum_{l=1}^{L} S_{\text{past}}^{(l)}}$

    $S_n \leftarrow N_{n-1} \langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle$

    if unlimited liquidity **then**

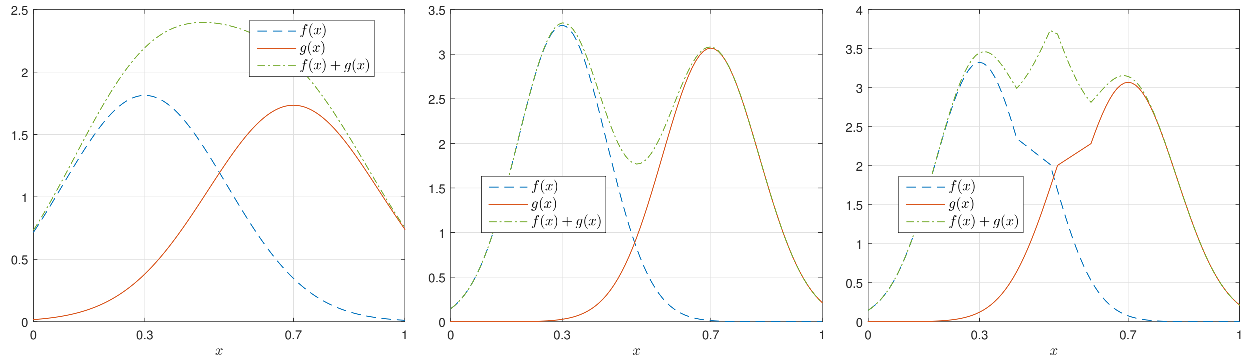
        $N_n \leftarrow S_n w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_n)$ `// by using (31).`

    else

        $N_n \leftarrow S_n w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_n, N_{n-1})$ `// by using (39).`

    end

end



(a) $f(x)+g(x)$ is unimodal, when $f(x)$ is PDF of $\mathcal{N}(0.3, 0.22^2)$, and $g(x)$ is PDF of $\mathcal{N}(0.7, 0.23^2)$.

(b) $f(x)+g(x)$ is bimodal, when $f(x)$ is PDF of $\mathcal{N}(0.3, 0.12^2)$, and $g(x)$ is PDF of $\mathcal{N}(0.7, 0.13^2)$.

(c) $f(x)+g(x)$ is trimodal, when $f(x)$ and $g(x)$ are modified functions of Figure 10(b).

Figure 10. The sum of two unimodal functions is not necessarily unimodal.

$f(\boldsymbol{b})$ is either (i) a constant, (ii) unimodal function of $\boldsymbol{b} \in \Delta^{d-1}$, or (iii) strictly increasing (decreasing) function of $\boldsymbol{b}$ if its first partial derivative (Figure 9 shows the direction of the partial derivative when $d=3$)

$$
\begin{aligned}
\frac{\partial f(\boldsymbol{b})}{\partial \boldsymbol{b}} &= \left[\frac{\partial f(\boldsymbol{b})}{\partial b^{(1)}} \; \frac{\partial f(\boldsymbol{b})}{\partial b^{(2)}} \; \cdots \; \frac{\partial f(\boldsymbol{b})}{\partial b^{(d)}}\right]^{\mathrm{T}} \\
&= \left[\sum_{i \in J_n} \frac{x_{i+1}^{(1)} - x_{i+1}^{(d)}}{\langle \boldsymbol{b}, \boldsymbol{x}_{i+1} \rangle} \; \sum_{i \in J_n} \frac{x_{i+1}^{(2)} - x_{i+1}^{(1)}}{\langle \boldsymbol{b}, \boldsymbol{x}_{i+1} \rangle} \; \cdots \; \sum_{i \in J_n} \frac{x_{i+1}^{(d)} - x_{i+1}^{(d-1)}}{\langle \boldsymbol{b}, \boldsymbol{x}_{i+1} \rangle}\right]^{\mathrm{T}},
\end{aligned} \tag{51}
$$

is, respectively:

(i) $\boldsymbol{0}, \forall \boldsymbol{b} \in \Delta^{d-1}$, where $\boldsymbol{0}$ is a zero vector (a sufficient condition that makes $f(\boldsymbol{b})$ a constant is $\left[x_{i+1}^{(1)} \; x_{i+1}^{(2)} \; \dots \; x_{i+1}^{(d)}\right]^{\mathrm{T}} = \left[x_{i+1}^{(d)} \; x_{i+1}^{(1)} \; \dots \; x_{i+1}^{(d-1)}\right]^{\mathrm{T}}, \forall i \in J_n$),

(ii) $\boldsymbol{0}$ at a single point (necessary conditions that make $f(\boldsymbol{b})$ a unimodal function are $\exists i \in J_n : \left[x_{i+1}^{(1)} \; x_{i+1}^{(2)} \; \dots \; x_{i+1}^{(d)}\right]^{\mathrm{T}} \neq \left[x_{i+1}^{(d)} \; x_{i+1}^{(1)} \; \dots \; x_{i+1}^{(d-1)}\right]^{\mathrm{T}}$, and $|J_n| \geq 2$),

(iii) $\neg\boldsymbol{0}, \forall \boldsymbol{b} \in \Delta^{d-1}$ (a necessary condition that makes $f(\boldsymbol{b})$ a strictly increasing (decreasing) function is $\exists i \in J_n : \left[x_{i+1}^{(1)} \; x_{i+1}^{(2)} \; \dots \; x_{i+1}^{(d)}\right]^{\mathrm{T}} \neq \left[x_{i+1}^{(d)} \; x_{i+1}^{(1)} \; \dots \; x_{i+1}^{(d-1)}\right]^{\mathrm{T}}$).

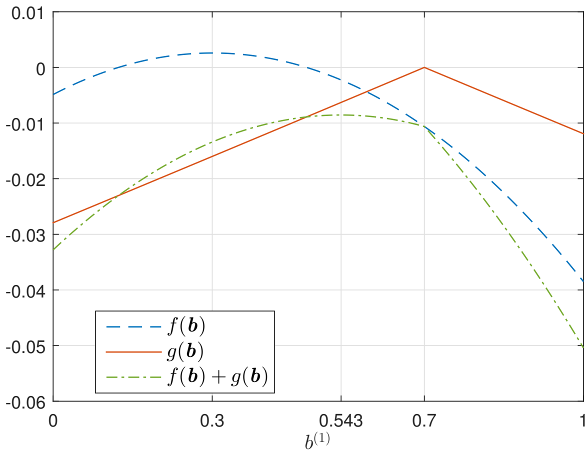
However, $f(\boldsymbol{b})$ cannot be multimodal since it is a concave function of $\boldsymbol{b}$ (see Appendix B).

$g(\boldsymbol{b})$ is a unimodal function of $\boldsymbol{b} \in \Delta^{d-1}$ for both the cases of $w_n = w(\boldsymbol{b}_n, \boldsymbol{b}, \boldsymbol{x}_n)$ in (31) and $w_n = w(\boldsymbol{b}_n, \boldsymbol{b}, \boldsymbol{x}_n, N_{n-1})$ in (39). This is because $w(\boldsymbol{b}_n, \boldsymbol{b}, \boldsymbol{x}_n)$ and $w(\boldsymbol{b}_n, \boldsymbol{b}, \boldsymbol{x}_n, N_{n-1})$ are unimodal functions of $\boldsymbol{b}$ as explained in Section 5.3; hence, $g(\boldsymbol{b})$, the logarithmic function (which is strictly increasing) of w_n, is also unimodal.

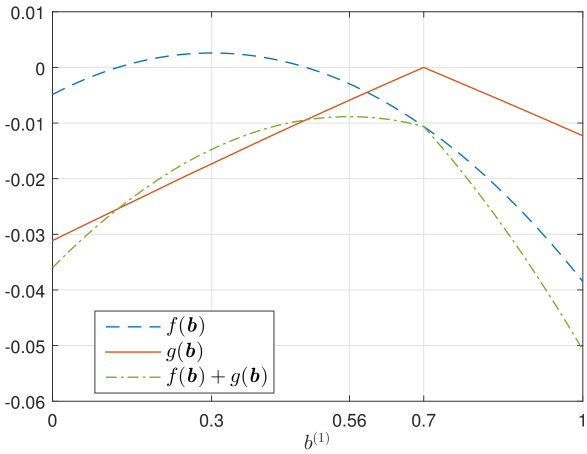
Even though it is true that the sum of two unimodal functions is not necessarily unimodal as shown in Figure 10, $f(\boldsymbol{b}) + g(\boldsymbol{b})$ is either unimodal or strictly increasing (decreasing) under a certain condition:

(i) If $f(\boldsymbol{b})$ is a constant, then $f(\boldsymbol{b}) + g(\boldsymbol{b})$ is unimodal without any conditions.

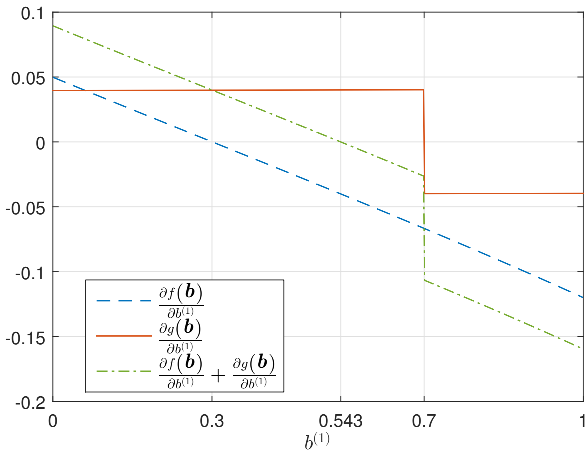
(ii) If $f(\boldsymbol{b})$ is unimodal, then the unimodality of the sum depends on the number of points that satisfy the equality: $\frac{\partial f(\boldsymbol{b})}{\partial \boldsymbol{b}} + \frac{\partial g(\boldsymbol{b})}{\partial \boldsymbol{b}} = \boldsymbol{0}$, which denotes the local maximum of the sum. Figure {11(a), 11(b)} shows an example of the sum of two unimodal functions by using {(31), (39)}, where $\arg\max_{\boldsymbol{b} \in \Delta^1} f(\boldsymbol{b}) = [0.3 \; 0.7]^{\mathrm{T}}$, and $\arg\max_{\boldsymbol{b} \in \Delta^1} g(\boldsymbol{b}) = [0.7 \; 0.3]^{\mathrm{T}}$. Hence, there exists a possibility that the sum has multiple local maxima between 0.3 and 0.7 (e.g. Figure 10(b) and 10(c)). However, the sum



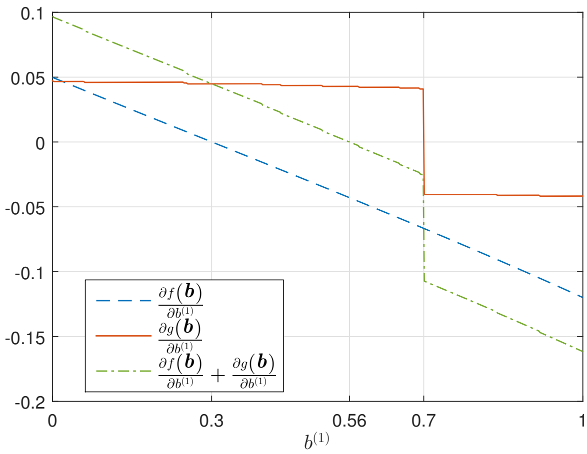
(a) $g(\boldsymbol{b}) = 2\ln w_n$, where $w_n = (\boldsymbol{b}_n, \boldsymbol{b}, \boldsymbol{x}_n)$ in (31).



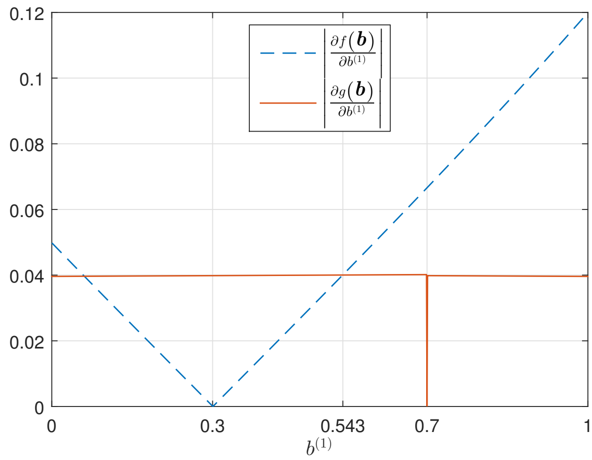
(b) $g(\boldsymbol{b}) = 2\ln w_n$, where $w_n = (\boldsymbol{b}_n, \boldsymbol{b}, \boldsymbol{x}_n, N_{n-1} = 10^7)$ in (39). Limit order book data of INTC ($b^{(1)}$) and MSFT ($b^{(2)}$) on 21 Jun 2012 at 16:00:00 was used.



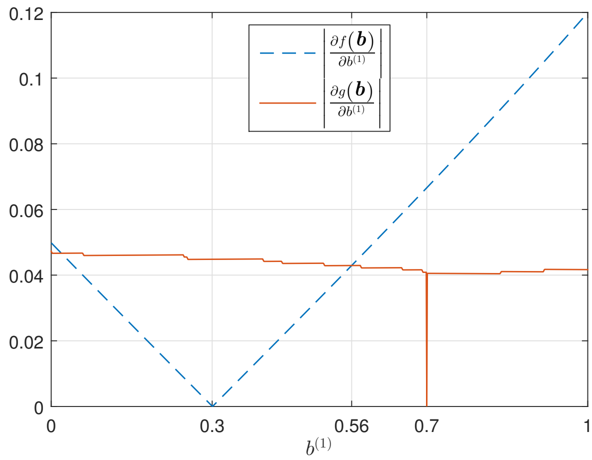
(c) First-order numerical differentiation of the upper figure.



(d) First-order numerical differentiation of the upper figure.



(e) Absolute value of the upper figure.



(f) Absolute value of the upper figure.

Figure 11. If $f(\boldsymbol{b})$ is unimodal, then $f(\boldsymbol{b}) + g(\boldsymbol{b})$ is also unimodal under low proportional transaction cost rates ($f(\boldsymbol{b}) = \ln\left\langle \boldsymbol{b}, [1.151\ 0.876]^{\mathrm{T}} \right\rangle + \ln\left\langle \boldsymbol{b}, [0.836\ 1.136]^{\mathrm{T}} \right\rangle$, $c_p = c_s = 1\%$, $\boldsymbol{b}_n = [0.7\ 0.3]^{\mathrm{T}}$, and $\boldsymbol{x}_n = [1.1\ 1.1]^{\mathrm{T}}$). More non-differentiable points are observed in Figure {11(d),11(f)} than Figure {11(c),11(e)} because the average market impact cost function is piecewise as shown in Figure 5.

has a unique local maximum at the single point: $\{0.543, 0.560\}$, since the first-order partial derivative of the sum is 0 at $\{0.543, 0.560\}$ as shown in Figure {11(c), 11(d)}. This is because both i) the opposite sign (i.e. $\frac{\partial f(\boldsymbol{b})}{\partial b^{(1)}} < 0, \frac{\partial g(\boldsymbol{b})}{\partial b^{(1)}} > 0, \forall b^{(1)} \in (0.3, 0.7)$) as shown in Figure {11(c), 11(d)}, and ii) the crossing between $\left|\frac{\partial f(\boldsymbol{b})}{\partial b^{(1)}}\right|$ and $\left|\frac{\partial g(\boldsymbol{b})}{\partial b^{(1)}}\right|$ only once in the interval $(0.3, 0.7)$ as shown in Figure {11(e), 11(f)}. Of course, it is possible that they cross more than once in the interval since both are increasing functions in the interval (i.e. $\frac{\partial\left|\frac{\partial f(\boldsymbol{b})}{\partial b^{(1)}}\right|}{\partial b^{(1)}} = -\frac{\partial^2 f(\boldsymbol{b})}{\left(\partial b^{(1)}\right)^2} = \sum_{i \in J_n} \frac{\left(x_{i+1}^{(1)} - x_{i+1}^{(2)}\right)^2}{\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle^2} > 0, \forall b^{(1)} \in (0.3, 0.7)$, and $\frac{\partial\left|\frac{\partial g(\boldsymbol{b})}{\partial b^{(1)}}\right|}{\partial b^{(1)}} = \frac{\partial^2 g(\boldsymbol{b})}{\left(\partial b^{(1)}\right)^2} > 0, \forall b^{(1)} \in (0.3, 0.7)$[1]). However, if the proportional TC rate is low (technically, low $\frac{c_p + c_s}{1 + c_p}$ from (C8)), the second partial derivative of $g(\boldsymbol{b})$ is approximated as 0 (i.e. $\frac{\partial^2 g(\boldsymbol{b})}{\left(\partial b^{(1)}\right)^2} \approx 0$), which in turn makes $\left|\frac{\partial g(\boldsymbol{b})}{\partial b^{(1)}}\right|$ {a constant, piecewise constant function} in the case of {(31), (39)} as shown in Figure {11(e), 11(f)}. As a result, $\left|\frac{\partial f(\boldsymbol{b})}{\partial b^{(1)}}\right|$ and $\left|\frac{\partial g(\boldsymbol{b})}{\partial b^{(1)}}\right|$ cross only once in the interval $(0.3, 0.7)$ when $\frac{c_p + c_s}{1 + c_p}$ is low, which in turn makes $f(\boldsymbol{b}) + g(\boldsymbol{b})$ unimodal.

(iii) If $f(\boldsymbol{b})$ is strictly increasing (decreasing), then $f(\boldsymbol{b}) + g(\boldsymbol{b})$ is either unimodal or strictly increasing (decreasing) under the same condition of low proportional TC rates as (ii). Even if there exists an interval of the opposite sign between $\frac{\partial f(\boldsymbol{b})}{\partial b^{(1)}}$ and $\frac{\partial g(\boldsymbol{b})}{\partial b^{(1)}}$, their absolute values cross at most once if the latter is almost constant.

Consequently, $f(\boldsymbol{b}) + g(\boldsymbol{b})$ is either unimodal or strictly increasing (decreasing) under the sufficient condition of low proportional TC rates (i.e. low $\frac{c_p + c_s}{1 + c_p}$), which implies that a local maximum is unique. Thus, the local optimum is guaranteed to be the global optimum when $\frac{c_p + c_s}{1 + c_p}$ is low.

7.4. *Initial value of optimisation*

$\boldsymbol{b}_{n+1}^{\star}$ in (40) is recommended as the initial value of local optimisation, which makes $w_n = 1$ (i.e. $\boldsymbol{b}_{n+1}^{\star}$ is the portfolio vector of zero TCs by not rebalancing a portfolio), at the end of the n-th period. This is in order to reduce the computation time of the optimisation: calculating $g(\boldsymbol{b}) = |J_n| \ln w_n$ in (50) with TCF $w_n = w(\boldsymbol{b}_n, \boldsymbol{b}, \boldsymbol{x}_n, N_{n-1})$ in (39) requires using the price and volume data of LOBs between the first absolute level of LOB and a higher absolute level of LOB (a positive level means buying stocks, whereas a negative level means selling stocks) as $\boldsymbol{b}$ is farther away from $\boldsymbol{b}_{n+1}^{\star}$, which in turn causes a heavier burden to calculate (24). If the initial value is $\boldsymbol{b}_{n+1}^{\star}$, the heavy computation can be avoided at least at the early stage of the optimisation.

8. Simulations (backtesting)

Monte Carlo (MC) simulations consisting of independent trials of random stock selection—each stock has an equal chance of being selected—were conducted to compare the performance among the existing methods and the proposed method. This is similar to (Kozat and Singer 2011, Section 4), while the differences are i) the number of trials, increased from 50 to 100 in order to generate more

[1] The proof of $\frac{\partial^2 g(\boldsymbol{b})}{\left(\partial b^{(1)}\right)^2} > 0$ is in Appendix C in the case of (31), but no proof of $\frac{\partial^2 g(\boldsymbol{b})}{\left(\partial b^{(1)}\right)^2} > 0$ is provided in this chapter in the case of (39). Meanwhile, $\frac{\partial^2 g(\boldsymbol{b})}{\left(\partial b^{(1)}\right)^2} > 0$ is observed at differentiable points in Figure 11(b) by using the numerical differentiation: $\frac{\partial^2 g(\boldsymbol{b})}{\left(\partial b^{(1)}\right)^2} \approx \frac{g\left(\boldsymbol{b} + [h\ -h]^{\mathrm{T}}\right) - 2g(\boldsymbol{b}) + g\left(\boldsymbol{b} - [h\ -h]^{\mathrm{T}}\right)}{h^2}$, where $h = 10^{-3}$.

reliable results, and ii) the number of selected stocks, increased from 3 to 30 in order to generate more practical results.

10-level (i.e. 10 levels of the ask side and 10 levels of the bid side) historical LOB data[1] of NASDAQ 100 Index Components[2] (30 components are randomly selected among the 100 components at each trial) between 1 Jan 2008 and 31 Dec 2015 (total 2015 trading days) was downloaded from Limit Order Book System: The Efficient Reconstructor (LOBSTER),[3] and the LOB data was sampled with the period of one day at the end of regular NASDAQ stock market trading (i.e. 4:00:00 p.m. Eastern Time).[4] Also, all OPS methods in this section rebalanced a portfolio at the closing time on every U.S. trading day (i.e. rebalancing once a trading day).

The 8-year period of the NASDAQ data set is much shorter than the 44-year period of an NYSE data set between 1962 and 2006,[5] used in (Györfi and Vajda 2008, Györfi and Walk 2012, Horváth and Urbán 2012) for their experiments. Hence, the pattern-matching-based OPS methods (e.g. Györfi and Vajda (2008)'s and the proposed method) with the 8-year data might suffer from low performance due to insufficient data to discover the tendencies of the market vectors, as similarly observed in (Györfi *et al.* 2006, Table 4.1). Consequently, the two-stage splitting scheme (Györfi *et al.* 2012, p. 102):[6]

(i) learning phase (that ranges between 1 Jan 1998 and 31 Dec 2007): only data collection is conducted without portfolio rebalancing;

(ii) concurrent phase (that ranges between 1 Jan 2008 and 31 Dec 2015): both data collection and portfolio rebalancing using the learning and concurrent phases are conducted concurrently;

was employed in the following experiments for Györfi and Vajda (2008)'s and the proposed method.

Structural breaks are not considered under the assumption that stock returns are stationary time series. However, the concurrent phase might be broken into multiple parts if structural breaks exist. To be specific, the following steps are required: detecting structural breaks (multiple breaks tests may be conducted), estimating the number of change points, and testing change-point of returns and volatility (Andreou and Ghysels 2009). For example, the concurrent phase can be broken into the two parts: before and after the 2008 financial crisis, if a structural break exists at the stock price collapse between September and November in 2008 as shown in Figure 19(b). This is because any financial crisis could well be thought of as a switch in regime that is often reflected in a structural break in the market volatility (Chakrabarti and Sen 2013, p. 8).

The actual number of stock candidates is $100 - 34 - 5 = 61$ by the splitting: 34 companies listed on NASDAQ after 1 Jan 1998 (their historical data is not enough for the learning phase), and 5 companies delisted (ALTR, AMLN, DELL, and TLAB were acquired by other companies, and FWLT was voluntarily delisted from NASDAQ) before 31 Dec 2015 (their historical data is not enough for the concurrent phase). Therefore, the number of possible portfolio combinations is $\binom{61}{30} = 2.3 \times 10^{17}$, and the portion of $\frac{100}{2.3\times10^{17}} = 4.3 \times 10^{-16}$ is covered by the MC simulations.

The range of proportional TC rate was set as $c_p = 0$ and $0.00184\% \le c_s \le 0.5\%$ (i.e. $0.00184\% \le \frac{c_p+c_s}{1+c_p} \le 0.5\%$). This is because the securities transaction tax rates in most of the G20 countries vary between 0.1% and 0.5% (Matheson 2011), and those in the United States in

---

1 If accessing LOB data at greater than level 10 is required, ask price and volume at level $i \in \mathbb{Z}_{>10}$ are estimated as $P_i = P_{10} + \frac{P_{10}-P_{-1}}{10}(i-10)$, $V_i = \frac{\sum_{k=1}^{10} V_k}{10}$, respectively. Similarly, if accessing LOB data at less than level -10 is required, bid price and volume at level $i \in \mathbb{Z}_{<-10}$ are estimated as $P_i = P_{-10} + \frac{P_{-10}-P_1}{10}(-i-10)$, $V_i = \frac{\sum_{k=1}^{10} V_{-k}}{10}$, respectively.

2 Historical, not current, NASDAQ 100 Index Components on 1 Jan 2008 was downloaded from `http://marketcapitalizations.com/historical-data/historical-components-nasdaq/`.

3 LOBSTER (`https://lobsterdata.com/`) has LOB data from 27 Jun 2007 to the present, and the LOB data of LOBSTER does not include hidden LOBs (Huang and Polak 2011, Table 1).

4 1:00:00 p.m. data was used for the following NASDAQ early closing dates: 3 Jul 2008, 28 Nov 2008, 24 Dec 2008, 27 Nov 2009, 24 Dec 2009, 26 Nov 2010, 25 Nov 2011, 3 Jul 2012, 23 Nov 2012, 24 Dec 2012, 3 Jul 2013, 29 Nov 2013, 24 Dec 2013, 3 Jul 2014, 28 Nov 2014, 24 Dec 2014, 27 Nov 2015, and 24 Dec 2015.

5 The NYSE data is downloadable at `http://www.cs.bme.hu/~oti/portfolio/data.html`.

6 The original intention to introduce the two-stage splitting scheme by Györfi *et al.* (2012) is to prove its uselessness by the assumption of the stationarity and long-term investment.

2015 are $c_p = 0$ and $c_s = 0.184$ bps.[1] However, stock brokerage commissions were ignored by an assumption that institutional investors, who pay tiny commissions, rather than individual investors, are the main users of OPS.

When calculating the price relative of asset j of the n-th day, cash dividends, stock dividends, and stock splits should be considered as

$$x_n^{(j)} = \frac{M_n^{(j)} \frac{A_n^{(j)}}{C_n^{(j)}}}{M_{n-1}^{(j)} \frac{A_{n-1}^{(j)}}{C_{n-1}^{(j)}}}, \tag{52}$$

where $\left\{M_n^{(j)}, C_n^{(j)}, A_n^{(j)}\right\}$ is the {closing mid, closing, adjusted closing} price from {LOBSTER, Yahoo Finance, Yahoo Finance} of asset j of the n-th day. However, the proportion of the data mismatching between LOBSTER and Yahoo Finance (i.e. $(P_1 \neq C) \vee (P_{-1} \neq C)$, where P_1 (P_{-1}) is the best ask (bid) price at the closing time from LOBSTER, and C is the closing price from Yahoo Finance) for the 55 stocks and 2015 trading days was as high as 69.6%. Therefore, the price relative of asset j of the n-th day was calculated as

$$x_n^{(j)} = \frac{A_n^{(j)}}{A_{n-1}^{(j)}}. \tag{53}$$

The following MC simulations are categorised into two parts according to whether the liquidity of assets is unlimited or limited. The first is to provide a benchmark for the comparison between the ideal assumption (i.e. unlimited liquidity) and the real stock market (i.e. limited liquidity), and the second is to demonstrate both i) the performance deterioration of OPS by MICs and ii) the superiority of the proposed method in the environment of the limited liquidity. Additionally, graphical comparisons and computation time analysis are provided.

The MATLAB codes of the following experiments have been uploaded on `http://www.mathworks.com/matlabcentral/fileexchange/56496` to avoid any potential ambiguity of the MC simulations.[2] You may leave comments on the web page; any feedback or bug report is welcome.

8.1. *In the case that the liquidity of assets is unlimited*

In the case that the liquidity of assets is unlimited, TCF w_n in (31) is employed to calculate both the portfolio vector of next day $\boldsymbol{b}_{n+1}$ and net wealth N_n in (28) at the end of every day, but the LOB data is disregarded.

8.1.1. **Comparison between (Cover 1991) and (Blum and Kalai 1999).**

Since performance comparison between (Cover 1991) and (Blum and Kalai 1999) had not been conducted by Blum and Kalai (1999), it was carried out in this chapter. Their difference (i.e. Blum and Kalai (1999) takes into account c_p and c_s when calculating $\boldsymbol{b}_{n+1}$, whereas Cover (1991) does not) is described in Algorithm 2 in Appendix A.

The numerical integral in both (2) and (14) is performed by using MC methods. Let $\boldsymbol{b}(k)$ be the k-th ($k \in \{1, 2, \ldots, K\}$) random numbers of the Dirichlet distribution with the d-dimensional

---

[1] Order making fiscal year 2015 annual adjustments to transaction fee rates, U.S. Securities and Exchange Commission [Release No. 34-74057/15 Jan 2015].

[2] MATLAB fmincon (a local, not global, optimisation solver), which finds minimum of constrained nonlinear multivariable function, was utilised to solve (16) for both the existing method with (31) and the proposed method with (39). MATLAB R2011b and R2014b resulted in slightly different solutions of $\boldsymbol{h}^{(l)}(\boldsymbol{x}_{1:n})$ in (16), and the older version was used to perform the simulations of this chapter.

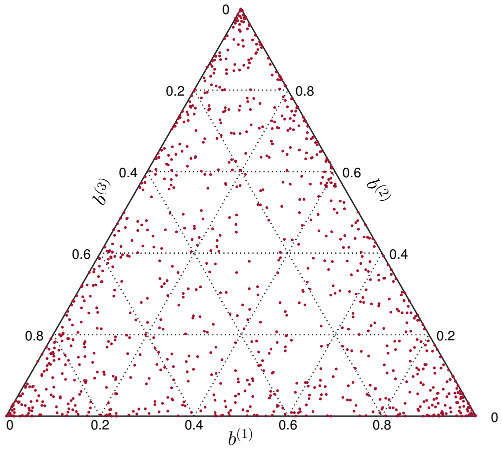


Figure 12. An example scatter plot of the Dirichlet distribution with $\boldsymbol{\alpha} = [1/2\ 1/2\ 1/2]^{\mathrm{T}}$ and $K = 10^3$.

Table 2. Statistics of annualised returns for comparison between Cover and Blum's method when market liquidity is unlimited ($c_p = 0$).

c_s (%)	0.1		0.2		0.3		0.4		0.5	
	Cover	Blum	Cover	Blum	Cover	Blum	Cover	Blum	Cover	Blum
P-value of JB test	0.374	0.374	0.376	0.376	0.378	0.378	0.380	0.380	0.381	0.382
Standard deviation (%)	1.16	1.16	1.16	1.16	1.15	1.15	1.15	1.15	1.15	1.15
Mean (%)	12.7	12.7	12.6	12.6	12.4	12.4	12.3	12.3	12.2	12.2
Difference of means[a] (bps)	-0.012		-0.022		-0.033		-0.042		-0.051	
P-value of t-test	0.999		0.999		0.998		0.998		0.997	

[a]Difference equals average annualised return of Blum minus that of Cover.

concentration parameter vector $\boldsymbol{\alpha} = [1/2\ 1/2\ \ldots\ 1/2]^{\mathrm{T}}$, where $K = 10^3$ is the number of samplings (Figure 12 shows an example). The portfolio vector in (2) is then approximated as (Ishijima 2001)

$$\boldsymbol{b}_{n+1} = \frac{\sum_{k=1}^{K} \boldsymbol{b}(k) S_n(\boldsymbol{b}(k), \boldsymbol{x}_{1:n})}{\sum_{k=1}^{K} S_n(\boldsymbol{b}(k), \boldsymbol{x}_{1:n})}. \tag{54}$$

Table 2 shows the performance comparison between the two methods by using the annualised return:

$$\left(\frac{S_T}{S_0}\right)^{\frac{252}{T}} - 1, \tag{55}$$

where T is the number of total trading days, and 252 is the number of trading days in a year. Even though the negative differences between the two means for all c_s imply that the Cover's method is better than the Blum's method, the statistical significance of the difference in means by unpaired two-sample t-tests with unequal variances, whose null hypothesis is that the data in two groups comes from independent random samples from normal distributions with equal means but different variances, is negligible for all c_s,[1] where the assumption of the normal distribution was tested by the Jarque–Bera (JB) tests.

The t-tests (the t-test denotes the unpaired two-sample t-test with unequal variances here and in the remainder of this chapter) answer whether the performance difference is fundamental or whether it is due to random fluctuations (Simon 2013, p. 631). In other words, if a p-value of

[1] Although the sample standard deviations at each c_s in Table 2 are similar, unpaired two-sample t-tests with equal variance, whose null hypothesis is that the data in two groups comes from independent random samples from normal distributions with equal means and equal but unknown variances, were not performed. This is because even when the variances are identical, the unequal variance t-test performs just as effectively as the equal variance t-test in terms of Type I error (Ruxton 2006).

the t-test is less than a significance level (e.g. 0.05), the performance difference is fundamental.[1] However, the p-values of the t-tests in Table 2 are greater than or equal to 0.05 for all c_s, which means that the performance difference between the Cover's method and the Blum's method is due to random fluctuations. As a result, the Blum's method is not compared to any other methods in the subsequent experiments.

8.1.2. Comparison among (Kozat and Singer 2011), (Cover 1991), and (Györfi and Vajda 2008).

[1] The p-value of the t-test is interpreted as the probability that a difference in the mean values would be obtained, given that the population means of two methods are equivalent. I.e. the p-value is not equal to the probability that the population means are equivalent (Simon 2013, p. 635).

Table 3. Statistics of annualised returns for comparison among Kozat, Cover, and Györfi's method when market liquidity is unlimited ($c_p = 0$).

c_s (%)	B&H	0.00184 Kozat	0.00184 Cover	0.00184 Györfi	0.1 Kozat	0.1 Cover	0.1 Györfi	0.2 Kozat	0.2 Cover	0.2 Györfi
P-value of JB test	0.419	0.603	0.768	**0.038**	0.602	0.76***	**0.011**	0.603	0.758	**0.009**
Standard deviation (%)	1.15	1.19	1.34	7.39	1.19	1.34	6.98	1.19	1.33	5.74
Mean (%)	11.3	12.1	12.8	21.4	12.1	12.7	17.6	12.1	12.6	16.1
Difference of means[a] (%)	-	0.80***	1.54***	10.10***	0.79***	1.40***	6.28***	0.79***	1.25***	4.82***
P-value of t-test	-	2.7×10^{-6}	9.7×10^{-16}	1.4×10^{-24}	3.3×10^{-6}	1.6×10^{-13}	2.1×10^{-14}	4.0×10^{-6}	2.3×10^{-11}	4.7×10^{-13}

c_s (%)	0.3 Kozat	0.3 Cover	0.3 Györfi	0.4 Kozat	0.4 Cover	0.4 Györfi	0.5 Kozat	0.5 Cover	0.5 Györfi
P-value of JB test	0.603	0.753	0.081	0.604	0.750	0.063	0.602	0.747	0.066
Standard deviation (%)	1.19	1.33	5.62	1.19	1.33	5.81	1.19	1.33	5.89
Mean (%)	12.1	12.4	15.0	12.1	12.3	14.5	12.1	12.1	14.0
Difference of means[a] (%)	0.78***	1.10***	3.69***	0.77***	0.95***	3.20***	0.76***	0.80***	2.69***
P-value of t-test	4.8×10^{-6}	2.5×10^{-9}	3.5×10^{-9}	5.9×10^{-6}	1.8×10^{-7}	4.1×10^{-7}	7.1×10^{-6}	8.7×10^{-6}	1.8×10^{-5}

[a]Difference equals average annualised return of the corresponding method at each c_s minus that of buy-and-hold (B&H). $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

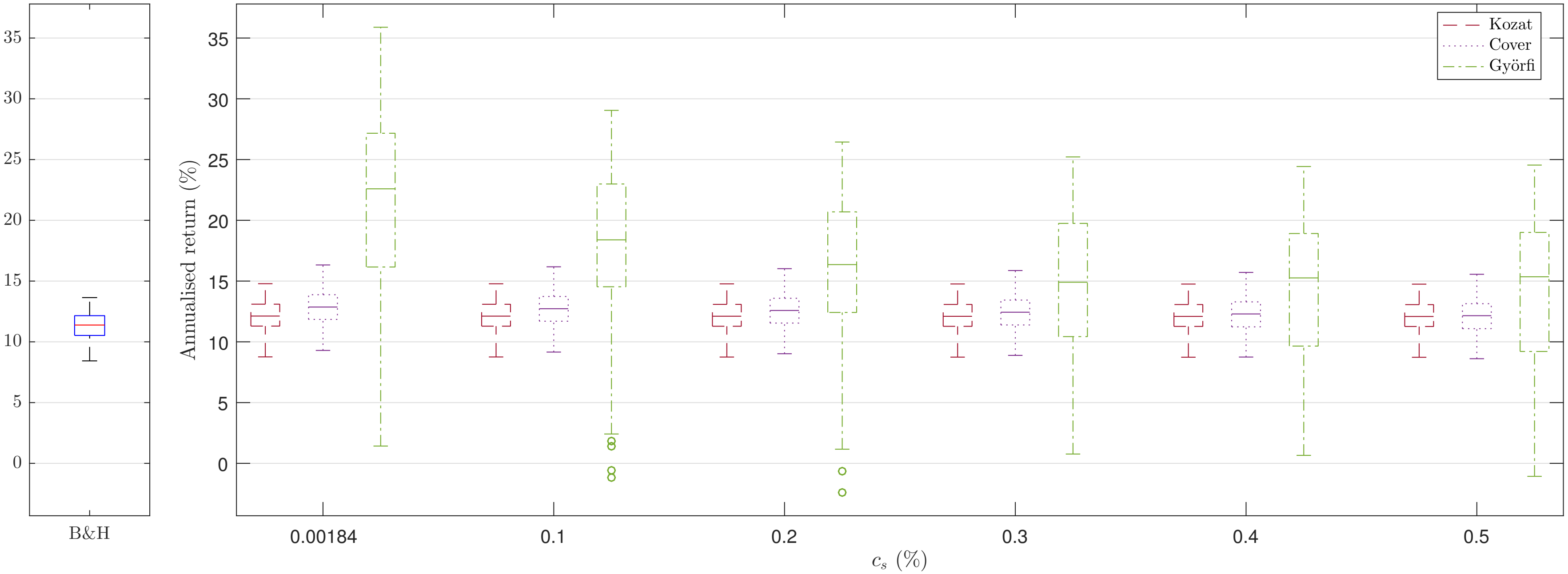


Figure 13. Box plots of annualised returns for comparison among Kozat, Cover, and Györfi's method when market liquidity is unlimited ($c_p = 0$).

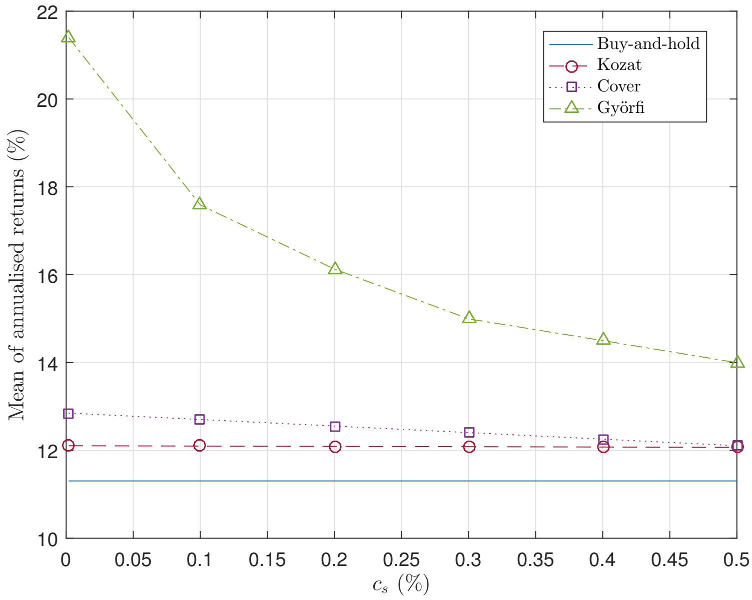


Figure 14. Mean of annualised returns for comparison among Kozat, Cover, and Györfi's method when market liquidity is unlimited ($c_p = 0$).

The t-tests were performed to compare the performance between a benchmark and OPS methods (Table 3 shows the results), where a strategy of buy-and-hold (B&H) with the initial portfolio $\boldsymbol{b}_1$ is the benchmark against all OPS methods, and its performance is independent of c_p or c_s as it does not incur any TCs. The Kozat, Cover, and Györfi's method show the positive differences with p-values less than 0.01 when $c_s \leq 0.5\%$, which means that these OPS methods are highly fundamentally superior to B&H when $c_s \leq 0.5\%$.

However, Wilcoxon rank sum tests (also called Wilcoxon–Mann–Whitney tests), whose null hypothesis is that data in two groups are samples from continuous distributions (even non-normal distributions) with equal medians, were not performed even though the violation of the normality assumption of the t-test was confirmed by the JB test with the significance level of 0.05 as shown in Table 3 (p-values of the JB test less than 0.05 are marked in bold, which implies non-normality). This is because the t-test is superior to the Wilcoxon rank sum test when variances (i.e. the variance of annualised returns of B&H and that of each OPS method) differ (Skovlund and Fenstad 2001).

Standard deviation in Table 3 can be interpreted as the sensitivity of the annualised return to the random stock selection: the Kozat's method and the Györfi's method are the least and most sensitive, respectively, for all c_s. In addition, Figure 14 (Figure 13) shows the decreasing trend of the performance as c_s increases by using the mean (box plots) of annualised returns of each method: the Kozat's method and the Györfi's method show the least and most decreasing trend, respectively.

Figure 13 shows that the higher expected return does not always guarantee the higher profits. In other words, there is no best OPS method, but investors may choose a preferable OPS method by considering both the expected return and the risk (i.e. standard deviation of annualised returns). Meanwhile, negative returns are observed in the low outliers of the Györfi's method when $c_s \geq 0.1\%$, resulting in the loss of money (i.e. $S_T < S_0$).

8.2. *In the case that the liquidity of assets is limited*

In the case that the liquidity of assets is limited, TCF w_n in (39) is employed to calculate the net wealth N_n in (28) at the end of every day. Therefore, the performance of all OPS methods depends on initial wealth S_0 because TCF w_n in (39) is a function of S_n: i.e. the greater S_0; the greater MICs, which in turn causes less performance of OPS compared to Section 8.1. However, only the

Table 4. Statistics of annualised returns for comparison among Kozat, Cover, Györfi's, and proposed method when market liquidity is limited and $S_0 = 10^4$ ($c_p = 0$).

c_s (%)	B&H	0.00184 Kozat	0.00184 Cover	0.00184 Györfi	0.00184 Proposed	0.1 Kozat	0.1 Cover	0.1 Györfi	0.1 Proposed	0.2 Kozat	0.2 Cover	0.2 Györfi	0.2 Proposed
P-value of JB test	0.419	0.601	0.735	**0.009**	**0.027**	0.602	0.728	**0.009**	**0.016**	0.604	0.723	**0.020**	0.149
Standard deviation (%)	1.15	1.19	1.33	8.50	7.00	1.19	1.33	7.81	5.89	1.19	1.32	5.92	5.38
Mean (%)	11.3	12.1	12.7	16.0	17.9	12.1	12.6	14.5	16.1	12.1	12.4	14.4	15.0
Difference of means[a] (%)	-	0.79***	1.39***	4.67***	6.61***	0.79***	1.25***	3.22***	4.84***	0.78***	1.10***	3.14***	3.74***
P-value of t-test	-	3.4×10^{-6}	1.6×10^{-13}	3.6×10^{-7}	2.3×10^{-15}	4.1×10^{-6}	2.2×10^{-11}	8.8×10^{-5}	1.1×10^{-12}	5.0×10^{-6}	2.3×10^{-9}	9.4×10^{-7}	6.0×10^{-10}

c_s (%)	0.3 Kozat	0.3 Cover	0.3 Györfi	0.3 Proposed	0.4 Kozat	0.4 Cover	0.4 Györfi	0.4 Proposed	0.5 Kozat	0.5 Cover	0.5 Györfi	0.5 Proposed
P-value of JB test	0.603	0.716	0.074	0.066	0.603	0.712	0.059	0.056	0.603	0.707	0.063	0.067
Standard deviation (%)	1.19	1.32	5.74	5.55	1.19	1.32	5.89	5.84	1.19	1.32	5.95	5.93
Mean (%)	12.1	12.3	13.9	14.3	12.1	12.1	13.7	13.8	12.1	12.0	13.3	13.2
Difference of means[a] (%)	0.77***	0.95***	2.64***	3.04***	0.76***	0.80***	2.39***	2.52***	0.76***	0.65***	2.02***	1.91***
P-value of t-test	6.0×10^{-6}	1.7×10^{-7}	1.7×10^{-5}	4.6×10^{-7}	7.3×10^{-6}	8.4×10^{-6}	1.2×10^{-4}	4.9×10^{-5}	8.8×10^{-6}	2.5×10^{-4}	1.2×10^{-3}	2.1×10^{-3}

[a]Difference equals average annualised return of the corresponding method at each c_s minus that of buy-and-hold (B&H). $^*p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

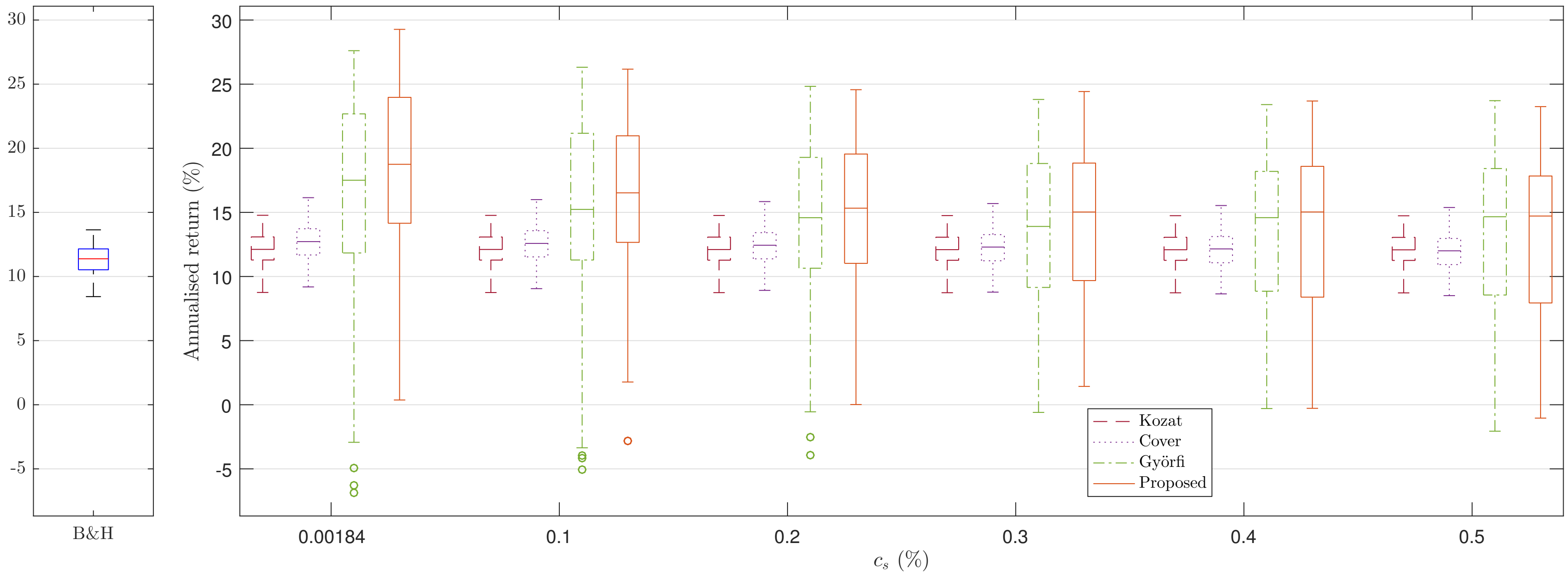


Figure 15. Box plot of annualised returns for comparison among Kozat, Cover, Györfi's, and proposed method when market liquidity is limited and $S_0 = 10^4$ ($c_p = 0$).

Table 5. Statistics of annualised returns for comparison among Kozat, Cover, Györfi's, and proposed method when market liquidity is limited and $S_0 = 10^5$ ($c_p = 0$).

c_s (%)	B&H	0.00184 Kozat	0.00184 Cover	0.00184 Györfi	0.00184 Proposed	0.1 Kozat	0.1 Cover	0.1 Györfi	0.1 Proposed	0.2 Kozat	0.2 Cover	0.2 Györfi	0.2 Proposed
P-value of JB test	0.419	0.603	0.731	**0.008**	**0.027**	0.601	0.726	**0.009**	**0.015**	0.602	0.722	**0.021**	0.165
Standard deviation (%)	1.15	1.19	1.33	8.51	6.79	1.19	1.33	7.83	5.71	1.19	1.32	5.92	5.36
Mean (%)	11.3	12.1	12.7	15.3	17.6	12.1	12.6	14.2	15.9	12.1	12.4	14.3	14.7
Difference of means[a] (%)	-	0.79***	1.39***	3.95***	6.28***	0.79***	1.25***	2.93***	4.61***	0.78***	1.10***	3.01***	3.40***
P-value of t-test	-	3.4×10^{-6}	1.7×10^{-13}	1.2×10^{-5}	6.2×10^{-15}	4.1×10^{-6}	2.3×10^{-11}	3.4×10^{-4}	2.5×10^{-12}	5.0×10^{-6}	2.5×10^{-9}	2.3×10^{-6}	1.0×10^{-8}

c_s (%)	0.3 Kozat	0.3 Cover	0.3 Györfi	0.3 Proposed	0.4 Kozat	0.4 Cover	0.4 Györfi	0.4 Proposed	0.5 Kozat	0.5 Cover	0.5 Györfi	0.5 Proposed
P-value of JB test	0.603	0.717	0.073	0.059	0.603	0.711	0.059	0.057	0.605	0.706	0.063	0.077
Standard deviation (%)	1.19	1.32	5.74	5.45	1.19	1.32	5.89	5.78	1.19	1.32	5.95	5.81
Mean (%)	12.1	12.3	13.9	13.9	12.1	12.1	13.7	13.4	12.1	12.0	13.3	12.9
Difference of means[a] (%)	0.77***	0.95***	2.57***	2.58***	0.76***	0.80***	2.35***	2.14***	0.76***	0.65***	1.99***	1.64***
P-value of t-test	6.0×10^{-6}	1.8×10^{-7}	2.6×10^{-5}	1.0×10^{-5}	7.3×10^{-6}	8.8×10^{-6}	1.6×10^{-4}	4.5×10^{-4}	8.8×10^{-6}	2.6×10^{-4}	1.4×10^{-3}	6.7×10^{-3}

[a]Difference equals average annualised return of the corresponding method at each c_s minus that of buy-and-hold (B&H). $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

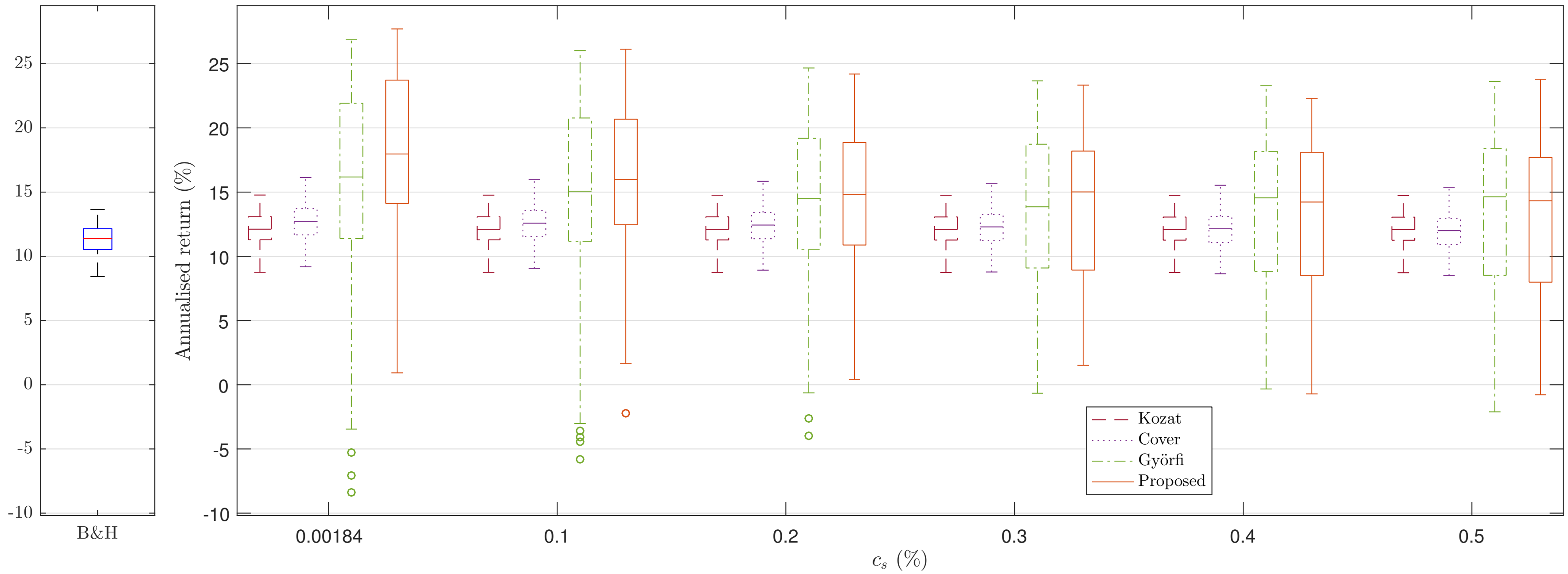


Figure 16. Box plot of annualised returns for comparison among Kozat, Cover, Györfi, and proposed method when market liquidity is limited and $S_0 = 10^5$ ($c_p = 0$).

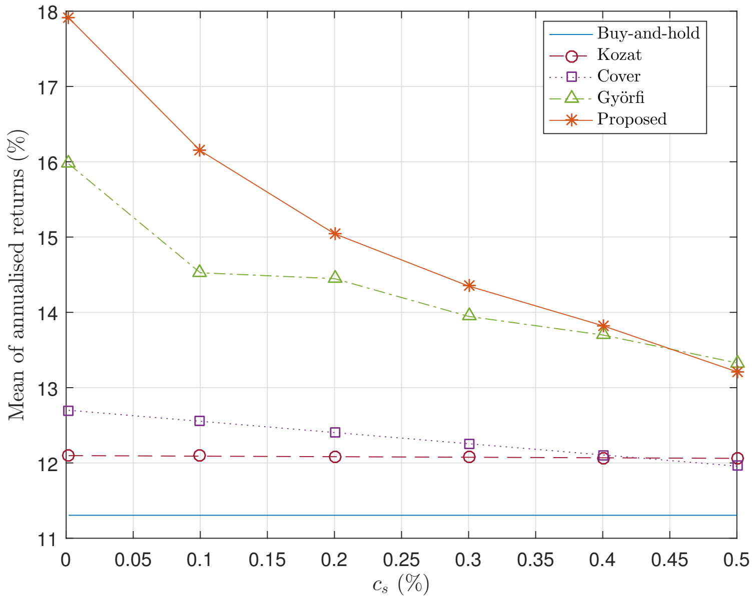


Figure 17. Mean of annualised returns for comparison among Kozat, Cover, Györfi's, and proposed method when market liquidity is limited and $S_0 = 10^4$ ($c_p = 0$).

Table 6. Mean difference of annualised returns between Györfi's and proposed method when $S_0 = 10^4$ ($c_p = 0$).

c_s (%)	0.00184	0.1	0.2	0.3	0.4	0.5
Difference of means[a] (%)	1.94*	1.62*	0.60	0.40	0.12	−0.12
P-value of t-test	0.080	0.099	0.457	0.614	0.882	0.889

[a]Difference equals average annualised return of the proposed method minus that of Györfi. $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

proposed method takes into account the LOB data as well as c_p, c_s when calculating $\boldsymbol{b}_{n+1}$.

8.2.1. **When initial wealth S_0 is small $\left(S_0 = 10^4\right)$.** When initial wealth S_0 is as small as USD 10k, the performance deterioration of the Györfi's method is severe when comparing between Figure 14 and 17 (or between Figure 13 and 15). In addition, the performance difference between the Györfi's and proposed method is marginally statistically significant when $c_s \leq 0.1\%$ as shown in Table 6, which means that the proposed method is not effective compared to the Györfi's method for small-sized funds with high proportional TCs. Moreover, the p-value tends to increase as c_s increases, which proves the low effectiveness of the proposed method with high proportional TCs. This was foreseen in Section 5.3: the proportion of MICs in TCs decreases as c_p or c_s increases, which in turn makes the proposed method less different from the Györfi's method.

8.2.2. **When initial wealth S_0 is large $\left(S_0 = 10^5\right)$.** When S_0 is as large as USD 100k (a hundred thousand US dollars is not a large fund size but relatively large for the Györfi's and proposed method in terms of MICs), the performance deterioration of the Györfi's method is higher than the proposed method when comparing between Figure 17 and 18 (or between

Table 7. Mean difference of annualised returns between Györfi's and proposed method when $S_0 = 10^5$ ($c_p = 0$).

c_s (%)	0.00184	0.1	0.2	0.3	0.4	0.5
Difference of means[a] (%)	2.33**	1.68*	0.39	0.01	−0.21	−0.35
P-value of t-test	0.034	0.085	0.626	0.991	0.797	0.673

[a]Difference equals average annualised return of the proposed method minus that of Györfi. $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

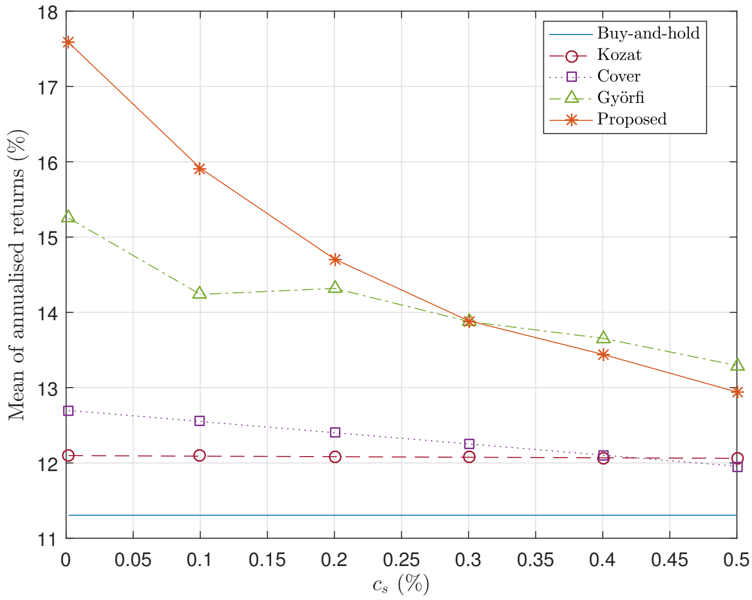


Figure 18. Mean of annualised returns for comparison among Kozat, Cover, Györfi, and proposed method when market liquidity is limited and $S_0 = 10^5$ ($c_p = 0$).

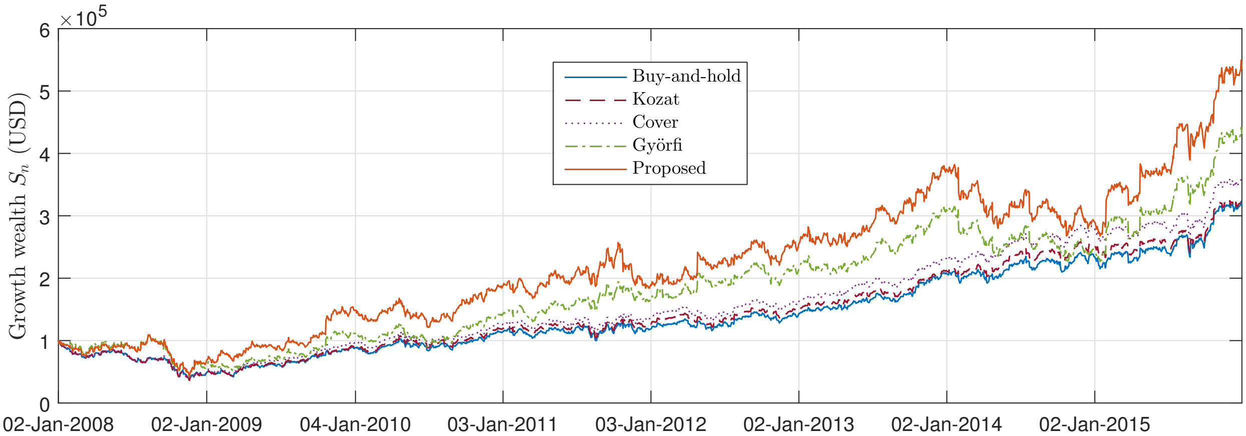
Figure 15 and 16). To be specific, the proposed method is fundamentally superior to the Györfi method when $c_s = 0.00184\%$ and $S_0 = 10^5$, while it is marginally fundamentally superior to the Györfi method when $c_s = 0.00184\%$ and $S_0 = 10^4$. This means that the proposed method is more effective than the Györfi method for large-sized funds and low TC rates (i.e. small value of $\frac{c_p+c_s}{1+c_p}$).

The performance of the Kozat's method and the Cover's method does not change even when S_0 increases from USD 10^4 to 10^5 as shown in Figure 17 and 18. This is for the following two reasons: i) $\boldsymbol{b}_{n+1}$ is independent of S_n in the case of the Kozat's method and the Cover's method. ii) The Kozat's method and the Cover's method trade much less than the Györfi's and the proposed method. I.e. the market order size $q_n^{(j)}$ in (39) of the Kozat's method and the Cover's method is much less than that of the Györfi's and the proposed method (this will be shown in Figure 21); hence, $\bar{p}\left(q_n^{(j)}\right)$ in (39) is replaced with either P_1 or P_{-1} (i.e. the best ask or bid price) for both $S_0 = 10^4$ and $S_0 = 10^5$, which results in the same TCF and same performance for each c_s between $S_0 = 10^4$ and $S_0 = 10^5$.

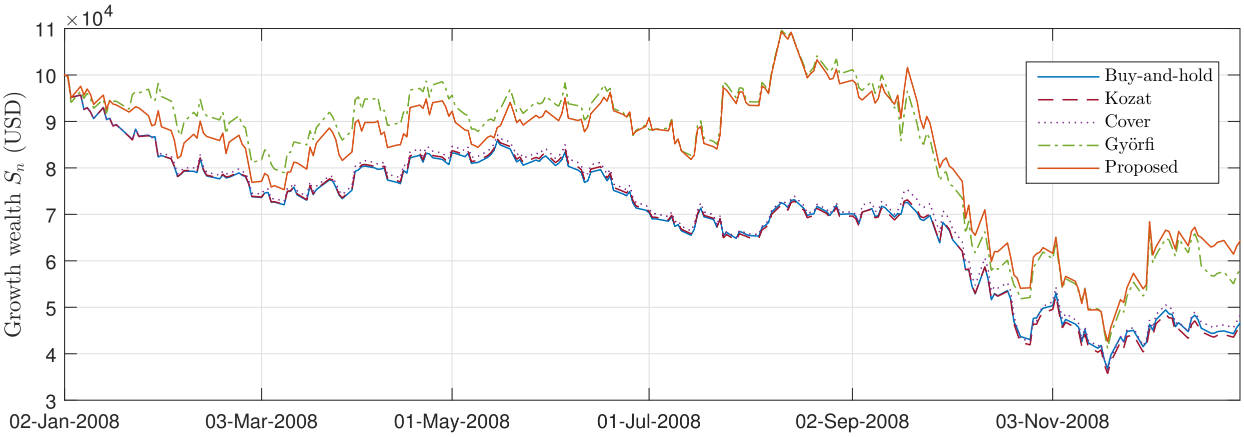
8.3. *Graphical comparisons*

Figure 19 shows the growth wealth S_n of a portfolio of five stocks when market liquidity is limited, and when initial wealth S_0 is USD 100k, where Figure 19(b) is the magnified plot of the beginning part of Figure 19(a). All the OPS methods made huge losses at the beginning part due to the financial crisis of 2008, but they have been converted to profits by the bull NASDAQ stock market for seven years since 2009.

Figure 20 shows the proportion of portfolio (i.e. the portfolio vector $\boldsymbol{b}_n$) that made the growth wealth in Figure 19(b), in the form of area plots. The portfolio vector of B&H changes over time, as shown in Figure 20(a), as the prices of assets change over time, and Figure 20(b) is similar to Figure 20(a) because the Kozat's method tries to minimize TCs. On the one hand, the Cover's method made an almost constant portfolio vector over time as shown in Figure 20(c), but on the other hand, the Györfi's and the proposed method generated abrupt changes of the portfolio vector over time, as shown in Figure 20(d) and 20(e). Meanwhile, Figure 20(e) shows fewer spikes than Figure 20(d) as the proposed method impedes the rapid portfolio changes: i.e. it considers MICs and makes a decision that a small change of the portfolio vector is more profitable than a large change.



(a) Between 2 Jan 2008 and 31 Dec 2015 (annualised return is buy-and-hold 15.5%; Kozat 15.7%; Cover 17.1%; Györfi 20.1%; proposed method 23.4%).



(b) Between 2 Jan 2008 and 31 Dec 2008.

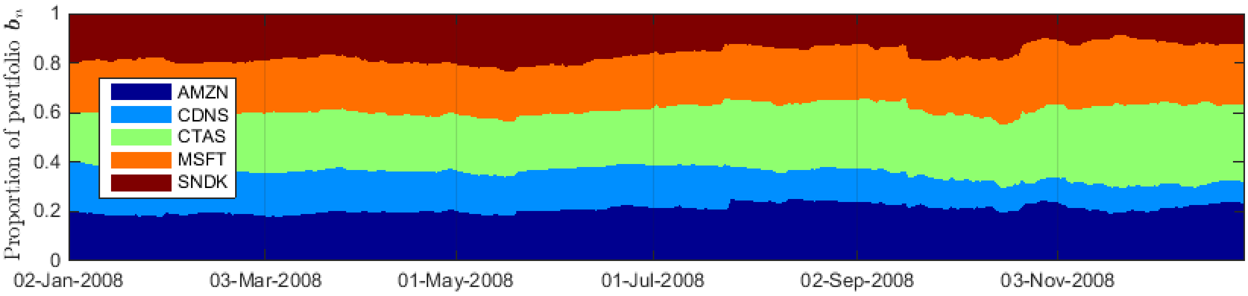
Figure 19. Growth wealth over time when the portfolio consists of AMZN, CDNS, CTAS, MSFT, and SNDK ($S_0 = 10^5$, $c_p = 0$, $c_s = 0.00184\%$).

Figure 21 shows market orders $q_n^{(j)}, \forall j = \{1, 2, \ldots, 5\}$ that made the growth wealth in Figure 19(b), calculated from (36). The amplitude of the market order varies among the OPS methods. In particular, Figure 21(d), by the proposed method, shows smaller amplitude than Figure 21(c), by the Györfi's method; the amplitude difference between Figure 21(d) and 21(c) corresponds to the difference between Figure 20(e) and 20(d).

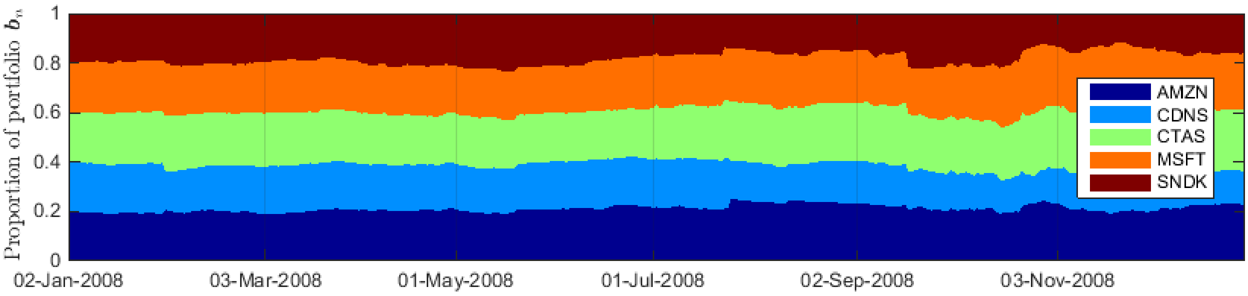
Figure 22 shows TCs (including MICs) by the marker orders in Figure 21 and confirms that the more TCs are charged as the greater magnitude of market order is generated, where TC including MIC at the end of the n-th day is calculated as

$$C_n = (1 - w_n) S_n, \tag{56}$$

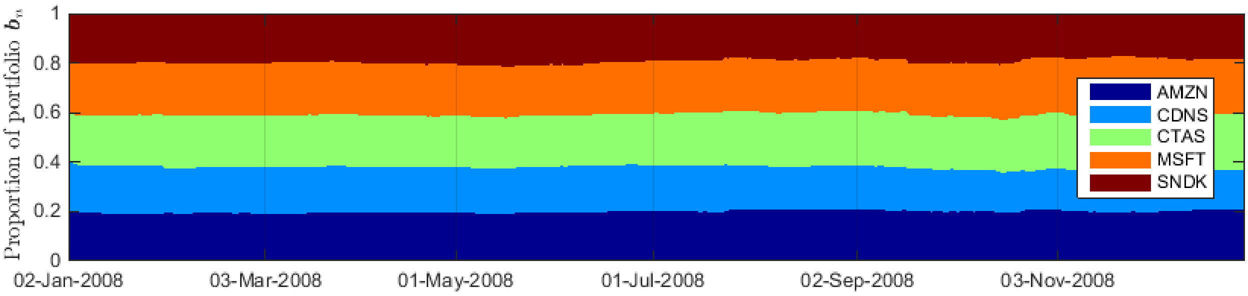
from (26) and (28). Both the Györfi's and the proposed method lead to zero TCs (i.e. $C_n = 0$, which means no trading at the end of the n-th day) on some days as shown in Figure 22(b), whereas the Kozat's method and the Cover's method do not, as shown in Figure 22(a). The zero TCs of the Györfi's and proposed method are caused by the two possibilities: no matching (i.e. $J_n = \varnothing$ in (17) for all l) and the dominance of TCs (i.e. $\boldsymbol{b}_{n+1}^{\star}$ in (40) is the solution of (16) for all l).



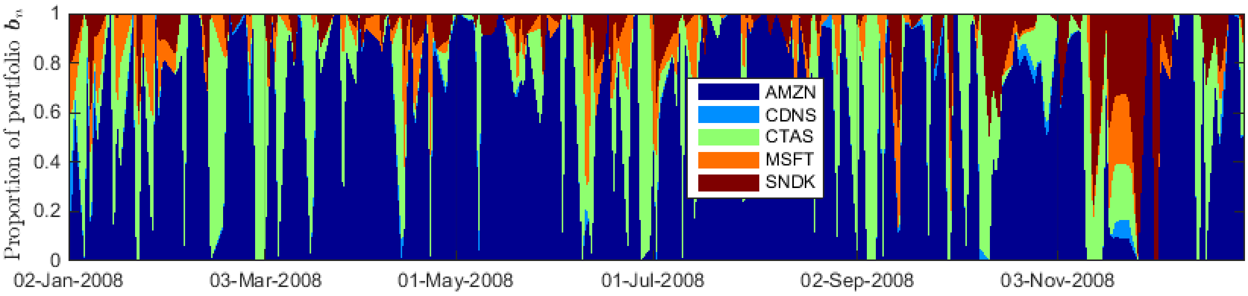
(a) Buy-and-hold.



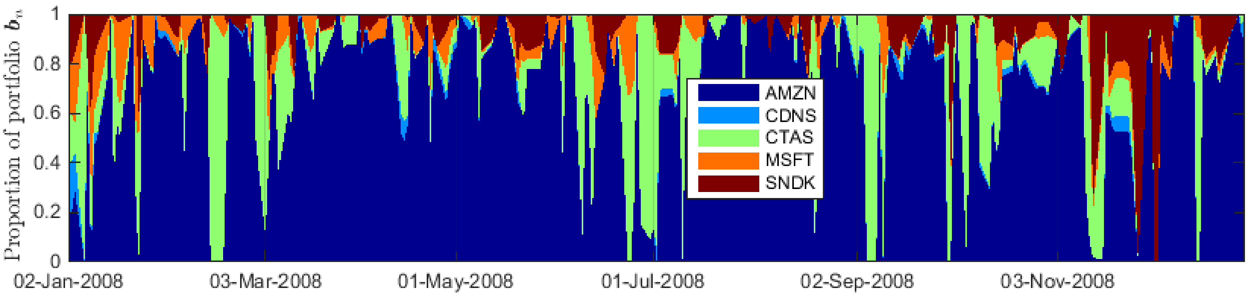
(b) Kozat.



(c) Cover.

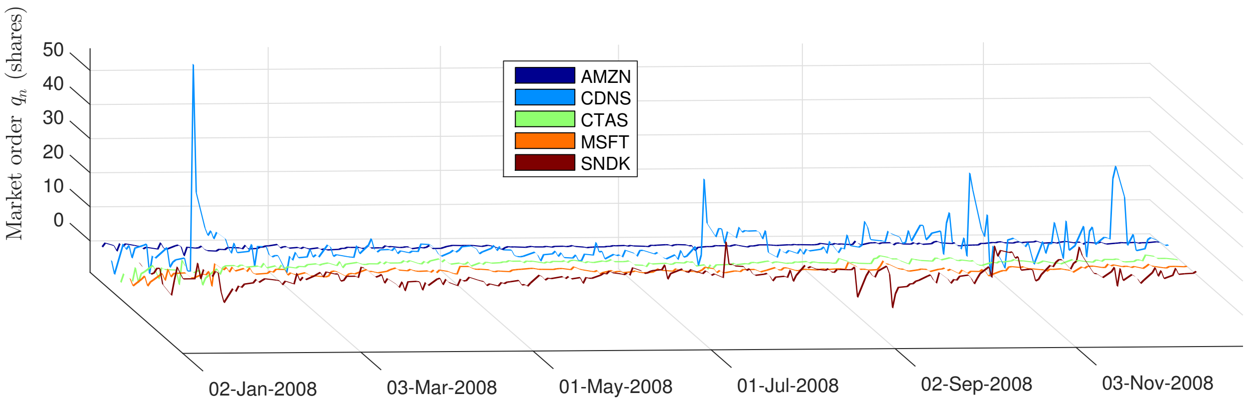


(d) Györfi.

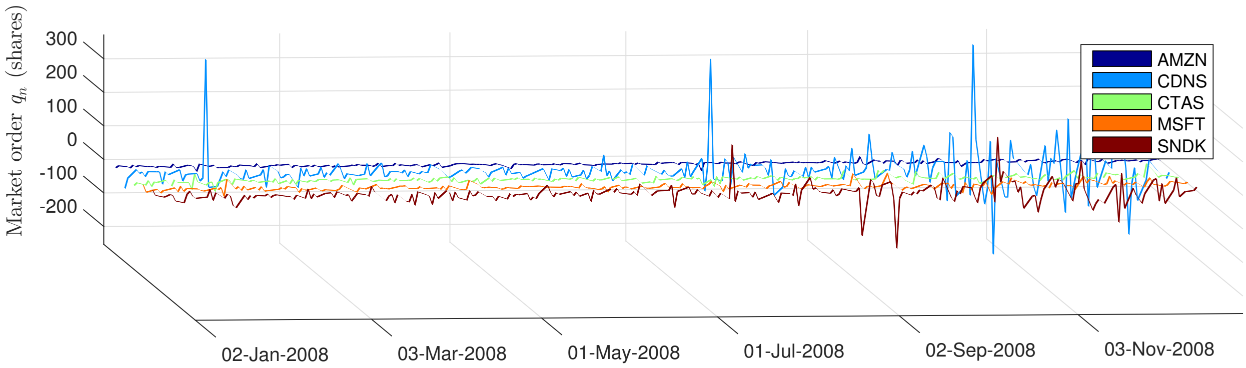


(e) Proposed method.

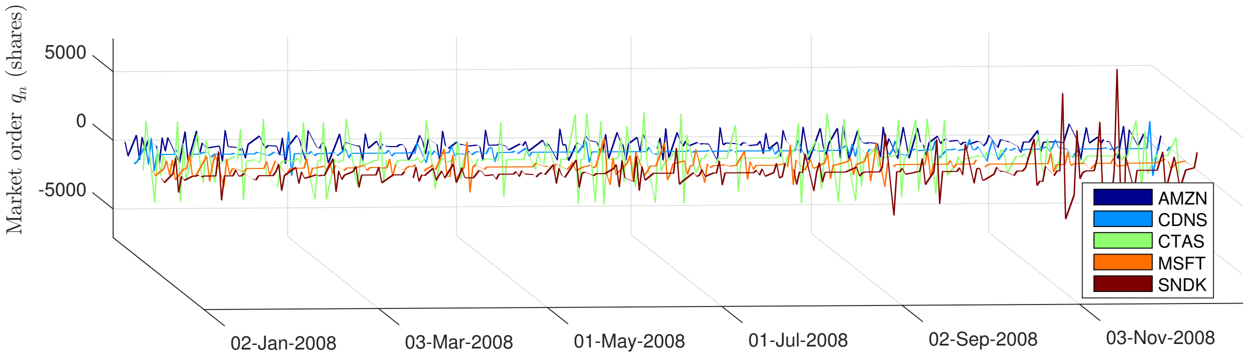
Figure 20. Proportion of portfolio over time between 2 Jan 2008 and 31 Dec 2008 ($S_0 = 10^5,\ c_p = 0,\ c_s = 0.00184\%$).



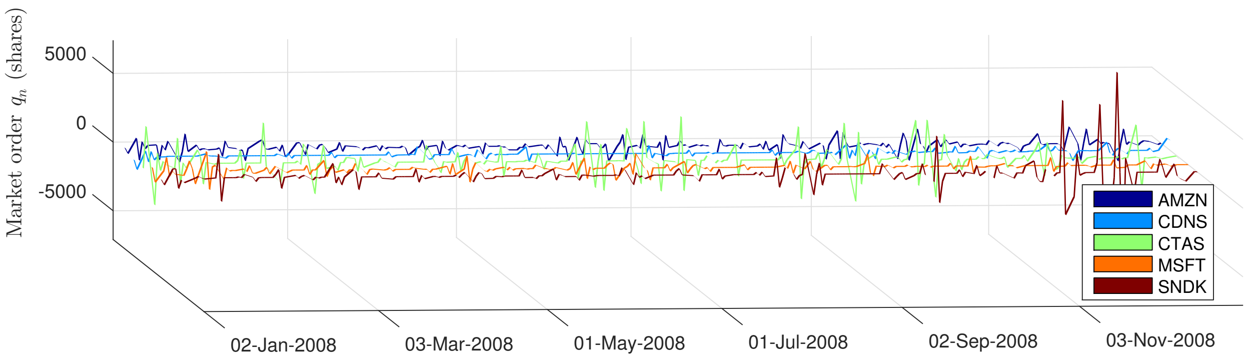
(a) Kozat.



(b) Cover.

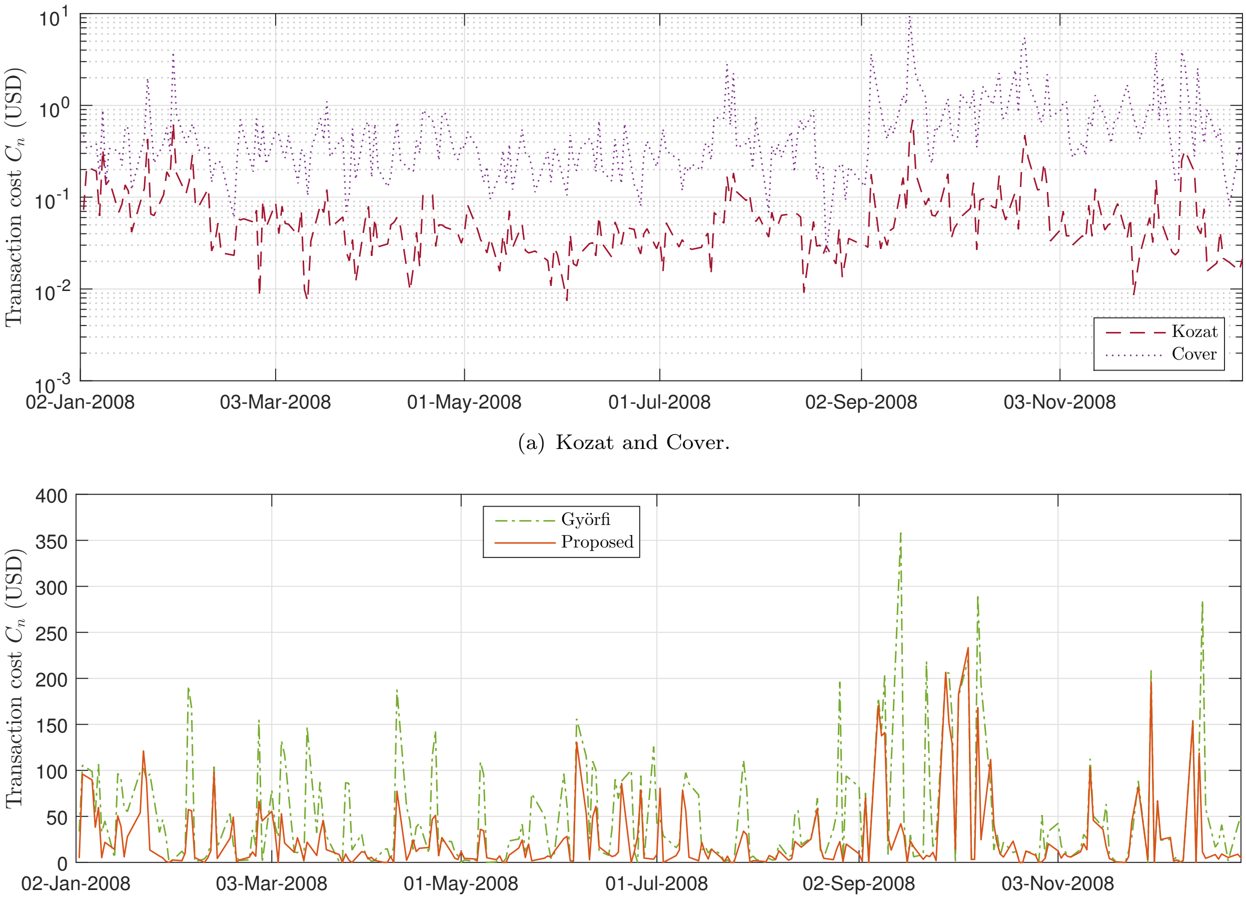


(c) Györfi.



(d) Proposed method.

Figure 21. Market order over time between 2 Jan 2008 and 31 Dec 2008 ($S_0 = 10^5$, $c_p = 0$, $c_s = 0.00184\%$). Positive (negative) values of market order indicate buying (selling) stocks.



(a) Kozat and Cover.

(b) Györfi's and proposed method.

Figure 22. Transaction costs including market impact costs over time between 2 Jan 2008 and 31 Dec 2008 ($S_0 = 10^5$, $c_p = 0$, $c_s = 0.00184\%$).

Table 8. Sample mean and standard deviation of computation time ($d = 30$).

	Kozat	Cover	Györfi	Proposed $(S_0 = 10^4)$	Proposed $(S_0 = 10^5)$
Time (seconds)	0.20 ± 0.02	0.03 ± 0.00	0.84 ± 1.31	0.97 ± 0.87	1.79 ± 1.22

8.4. *Computation time*

Table 8 shows the computation time to calculate $\boldsymbol{b}_{n+1}$ for the simulations in Section 8.2 by using AMD Opteron$^{\text{TM}}$ 6376 CPU and MATLAB R2011b. The computation time to calculate $\boldsymbol{b}_{n+1}$ depends on n (i.e. the greater n; the longer computation time) in the case of the Kozat's, Györfi's, and proposed method. Thus, only the worst case, $n = 2013$,[1] was measured for all the OPS methods. It takes longer for the proposed method to calculate $\boldsymbol{b}_{n+1}$ by the additional time to calculate MICs than the Györfi's method. In addition, the computation time of the proposed method increases as the initial wealth S_0 increases. This is because the larger wealth causes the heavier burden to calculate $\bar{p}(q)$ in (24) by additionally using LOB data in higher absolute levels.

[1] 30 Dec 2015 ($n = 2013$), not 31 Dec 2015 ($n = 2014$), was the last day to calculate $\boldsymbol{b}_{n+1}$, where 2 Jan 2008 corresponds to $n = 0$ (i.e. time 0; see Figure 1).

9. Conclusion

The foremost contribution of this chapter has been to develop the new TCF model in Section 5.2 and 5.3 by considering MICs, quantified by LOB data, as well as the proportional TCs, while the previous TCF model (Györfi and Vajda 2008) in Section 5.1 considered only the proportional TCs. Secondly, it has been applied to both i) measuring OPS performance in a more practical way by considering MICs (it was overestimated in all the previous OPS studies without MICs) and ii) developing the new OPS method, described in Algorithm 1 (Section 8.2.2 showed its superiority to the existing method (Györfi and Vajda 2008) for large-sized funds with low TC rates).

References

Alfonsi, A., Fruth, A. and Schied, A., Constrained portfolio liquidation in a limit order book model. In *Proceedings of the Advances in mathematics of finance*, Vol. 83 of *Banach Center publications*, pp. 9--25, 2008.

Andreou, E. and Ghysels, E., Structural breaks in financial time series. In *Handbook of financial time series*, pp. 839--870, 2009, Springer.

Bauwens, L., Pohlmeier, W. and Veredas, D., *High frequency financial econometrics: Recent developments*, 2007, Physica-Verlag.

Bean, A.J. and Singer, A.C., Universal switching and side information portfolios under transaction costs using factor graphs. *IEEE Journal of Selected Topics in Signal Processing*, 2012, **6**, 351--365.

Ben-Tal, A., Margalit, T. and Nemirovski, A., Robust modeling of multi-stage portfolio problems. In *High performance optimization*, pp. 303--328, 2000, Springer.

Blum, A. and Kalai, A., Universal portfolios with and without transaction costs. *Machine Learning*, 1999, **35**, 193--205.

Borodin, A. and El-Yaniv, R., *Online computation and competitive analysis*, 1998, Cambridge University Press.

Borodin, A., El-Yaniv, R. and Gogan, V., Can we learn to beat the best stock. *Journal of Artificial Intelligence Research*, 2004, pp. 579--594.

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

Brown, D.B. and Smith, J.E., Dynamic portfolio optimization with transaction costs: Heuristics and dual bounds. *Management Science*, 2011, **57**, 1752--1770.

Bussieck, M.R. and Vigerske, S., MINLP solver software. In *Wiley Encyclopedia of Operations Research and Management Science*, 2011, Wiley.

Chakrabarti, G. and Sen, C., *Momentum trading on the Indian stock market*, 2013, Springer.

Cover, T.M., Universal portfolios. *Mathematical finance*, 1991, **1**, 1--29.

Cover, T.M. and Ordentlich, E., Universal portfolios with side information. *IEEE Transactions on Information Theory*, 1996, **42**, 348--363.

Cover, T.M. and Thomas, J.A., *Elements of information theory* (2nd edn), 2006, Wiley.

Damodaran, A., *Investment philosophies: Successful strategies and the investors who made them work* (2nd edn), 2012, Wiley.

Dantzig, G.B. and Infanger, G., Multi-stage stochastic linear programs for portfolio optimization. *Annals of Operations Research*, 1993, **45**, 59--76.

Das, P., Online convex optimization and its application to online portfolio selection. PhD thesis, University of Minnesota, 2014.

Das, P., Johnson, N. and Banerjee, A., Online lazy updates for portfolio selection with transaction costs. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pp. 202--208, 2013.

Das, P., Johnson, N. and Banerjee, A., Online portfolio selection with group sparsity. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 1185--1191, 2014.

DeMiguel, V., Nogales, F.J. and Uppal, R., Stock return serial dependence and out-of-sample portfolio performance. *Review of Financial Studies*, 2014, **27**, 1031--1073.

Ferreira, M.A. and Santa-Clara, P., Forecasting stock market returns: The sum of the parts is more than the whole. *Journal of Financial Economics*, 2011, **100**, 514--537.

Freitas, F.D., De Souza, A.F. and de Almeida, A.R., Prediction-based portfolio optimization model using

neural networks. *Neurocomputing*, 2009, **72**, 2155--2170.
Gaivoronski, A.A. and Pflug, G., Value-at-risk in portfolio optimization: Properties and computational approach. *Journal of Risk*, 2005, **7**, 1--31.
Gârleanu, N. and Pedersen, L.H., Dynamic trading with predictable returns and transaction costs. *Journal of Finance*, 2013, **68**, 2309--2340.
Györfi, L., Lugosi, G. and Udina, F., Nonparametric kernel-based sequential investment strategies. *Mathematical Finance*, 2006, **16**, 337--357.
Györfi, L., Ottucsák, G. and Urbán, A., Empirical log-optimal portfolio selections: A survey. In *Machine learning for financial engineering*, pp. 81--118, 2012, Imperial College Press.
Györfi, L., Udina, F. and Walk, H., Experiments on universal portfolio selection using data from real markets. Technical report, 2008a.
Györfi, L., Udina, F. and Walk, H., Nonparametric nearest neighbor based empirical portfolio selection strategies. *Statistics & Decisions*, 2008b, **26**, 145--157.
Györfi, L., Urbán, A. and Vajda, I., Kernel-based semi-log-optimal empirical portfolio selection strategies. *International Journal of Theoretical and Applied Finance*, 2007, **10**, 505--516.
Györfi, L. and Vajda, I., Growth optimal investment with transaction costs. In *Algorithmic Learning Theory*, pp. 108--122, 2008, Springer.
Györfi, L. and Walk, H., Empirical portfolio selection strategies with proportional transaction costs. *IEEE Transactions on Information Theory*, 2012, **58**, 6320--6331.
Horváth, M. and Urbán, A., Growth optimal portfolio selection with short selling and leverage. In *Machine learning for financial engineering*, pp. 153--178, 2012, Imperial College Press.
Huang, R. and Polak, T., LOBSTER: Limit order book reconstruction system. Working paper, available at http://ssrn.com/abstract=1977207, 2011.
Ishijima, H., Numerical methods for universal portfolios. In *Proceedings of the Quantitative Methods in Finance Conference*, 2001.
Kozat, S.S. and Singer, A.C., Universal switching linear least squares prediction. *IEEE Transactions on Signal Processing*, 2008, **56**, 189--204.
Kozat, S.S. and Singer, A.C., Universal semiconstant rebalanced portfolios. *Mathematical Finance*, 2011, **21**, 293--311.
Levy, H. and Post, T., *Investments*, 2005, Pearson Education.
Li, B. and Hoi, S.C., Online portfolio selection: A survey. *ACM Computing Surveys*, 2014, **46**, 35.
Li, D. and Ng, W.L., Optimal dynamic portfolio selection: Multiperiod mean-variance formulation. *Mathematical Finance*, 2000, **10**, 387--406.
Luenberger, D.G., *Investment science*, 1998, Oxford University Press.
Markowitz, H.M., *Portfolio selection: efficient diversification of investments*, 1959, Yale University Press.
Matheson, T., Taxing financial transactions: Issues and evidence. Working paper 11/54, International Monetary Fund, 2011.
Olsson, R., Portfolio management under transaction costs: Model development and Swedish evidence. PhD thesis, Umeå University, 2005.
Ormos, M. and Urbán, A., Performance analysis of log-optimal portfolio strategies with transaction costs. *Quantitative Finance*, 2013, **13**, 1587--1597.
Otranto, E., Asset allocation using flexible dynamic correlation models with regime switching. *Quantitative Finance*, 2010, **10**, 325--338.
Ottucsák, G. and Vajda, I., An asymptotic analysis of the mean-variance portfolio selection. *Statistics & Decisions*, 2007, **25**, 63--86.
Palczewski, J., Poulsen, R., Schenk-Hoppé, K.R. and Wang, H., Dynamic portfolio optimization with transaction costs and state-dependent drift. *European Journal of Operational Research*, 2015, **243**, 921--931.
Pristas, G., Limit order book dynamics and asset liquidity. PhD thesis, University of Zurich, 2007.
Ruxton, G.D., The unequal variance t-test is an underused alternative to Student's t-test and the Mann--Whitney U test. *Behavioral Ecology*, 2006, **17**, 688--690.
Simchi-Levi, D., Chen, X. and Bramel, J., *The logic of logistics: Theory, algorithms, and applications for logistics management* (3rd edn), 2014, Springer.
Simon, D., *Evolutionary optimization algorithms*, 2013, Wiley.
Skovlund, E. and Fenstad, G.U., Should we always choose a nonparametric test when comparing two apparently nonnormal distributions?. *Journal of Clinical Epidemiology*, 2001, **54**, 86--92.

Steinbach, M.C., Markowitz revisited: Mean-variance models in financial portfolio analysis. *SIAM review*, 2001, **43**, 31--85.
Sydsæter, K., Strøm, A. and Berck, P., *Economists' mathematical manual* (4th edn), 2010, Springer.
Tunc, S., Donmez, M.A. and Kozat, S.S., Optimal investment under transaction costs: A threshold rebalanced portfolio approach. *IEEE Transactions on Signal Processing*, 2013, **61**, 3129--3142.

Appendix A: Algorithms of existing methods

Algorithm 2: Difference between (Cover 1991) and (Blum and Kalai 1999).

Input: c_p, c_s, S_0, $\boldsymbol{b}_1$, and $\boldsymbol{x}_{1:T}$, where T is the last period of trading.
Output: terminal wealth S_T.
`// Initialisation`
$K \leftarrow 10^3$
$N_0 \leftarrow S_0$
for $k \leftarrow 1$ *to* K **do**
  $\boldsymbol{b}(k) \leftarrow \mathrm{Dir}(\boldsymbol{\alpha})$ `// where Dir(`$\boldsymbol{\alpha}$`) is the random number generator of the Dirichlet distribution with the` d`-dimensional concentration parameter vector` $\boldsymbol{\alpha} = [1/2\ \ 1/2\ \ \ldots\ \ 1/2]^{\mathrm{T}}$.
  $N_{\mathrm{past}}(k) \leftarrow 1$
end
`// A loop to update` $\boldsymbol{b}_{n+1}$
for $n \leftarrow 1$ *to* T **do**
  for $k \leftarrow 1$ *to* K **do**
    switch method **do**
      case Cover **do**
        $N_{\mathrm{past}}(k) \leftarrow N_{\mathrm{past}}(k)\langle \boldsymbol{b}(k), \boldsymbol{x}_n \rangle$
      case Blum **do**
        $N_{\mathrm{past}}(k) \leftarrow N_{\mathrm{past}}(k)\langle \boldsymbol{b}(k), \boldsymbol{x}_n \rangle w(\boldsymbol{b}(k), \boldsymbol{b}(k), \boldsymbol{x}_n)$ `// by using (31).`
    end
  end
  $\boldsymbol{b}_{n+1} \leftarrow \dfrac{\sum_{k=1}^{K} \boldsymbol{b}(k) N_{\mathrm{past}}(k)}{\sum_{k=1}^{K} N_{\mathrm{past}}(k)}$
  $S_n \leftarrow N_{n-1}\langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle$
  if unlimited liquidity **then**
    $N_n \leftarrow S_n w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_n)$ `// by using (31).`
  else
    $N_n \leftarrow S_n w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_n, N_{n-1})$ `// by using (39).`
  end
end

This section is composed of Algorithm 2 and 3.

Appendix B: Concavity of Equation (4)

Let us check whether $\sum_{i \in J_n} \ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1} \rangle$ in (4) is a concave function of $\boldsymbol{b} \in \Delta^{d-1}$ or not. The first-order partial derivative of $\ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1} \rangle$ with respect to $b^{(j)}$ (Figure 9 shows the direction of the

Algorithm 3: Universal semi-constant rebalanced portfolio (Kozat and Singer 2011, p. 305).

Input: c_p, c_s, S_0, $\boldsymbol{b}_1$, and $\boldsymbol{x}_{1:T}$, where T is the last period of trading.
Output: terminal wealth S_T.

// Initialisation
$N_0 \leftarrow S_0$
for $j \leftarrow 1$ *to* d **do**
    $W_0(0,j) \leftarrow b_1^{(j)}$
end
// A loop to update $\boldsymbol{b}_{n+1}$
for $n \leftarrow 1$ *to* T **do**
    // Update $W_n(\boldsymbol{x}_n, n, j), \forall j$ for the upward paths (see Figure 2)
    for $j \leftarrow 1$ *to* d **do**
        $W_n(\boldsymbol{x}_n, n, j) \leftarrow 0$ // A bug ($W_n(\boldsymbol{x}_n, n, j)$ is set as 0 for several times in (Kozat and Singer 2011, Figure 3.2)) has been fixed.
    end
    for $i \leftarrow 0$ *to* $n-1$ **do**
        $\tau \leftarrow 0$ // where τ is the total wealth of $W_{n-1}(\boldsymbol{x}_{n-1}, i, \cdot)$.
        for $j \leftarrow 1$ *to* d **do**
            $\tau \leftarrow \tau + W_{n-1}(\boldsymbol{x}_{n-1}, i, j)$
        end
        for $j \leftarrow 1$ *to* d **do**
            $W_n(\boldsymbol{x}_n, n, j) \leftarrow W_n(\boldsymbol{x}_n, n, j) + \tau \dfrac{1/2}{n-i} b_1^{(j)} x_n^{(j)}$ // A bug ($b_n^{(j)}$ is used instead of $b_1^{(j)}$ in (Kozat and Singer 2011, Figure 3.2)) has been fixed.
        end
    end
    // Update $W_n(\boldsymbol{x}_n, n, j), \forall j$ for the horizontal paths (see Figure 2)
    for $i \leftarrow 0$ *to* $n-1$ **do**
        for $j \leftarrow 1$ *to* d **do**
            $W_n(\boldsymbol{x}_n, i, j) \leftarrow W_{n-1}(\boldsymbol{x}_{n-1}, i, j) \dfrac{n-i-1/2}{n-i} x_n^{(j)}$
        end
    end
    $\boldsymbol{b}_{n+1} \leftarrow \sum_{i=0}^{n} \sum_{j=1}^{d} \dfrac{W_n(\boldsymbol{x}_n, i, j)}{\sum_{k=0}^{n} \sum_{l=1}^{d} W_n(\boldsymbol{x}_n, k, l)} \left(\dfrac{n-i+1/2}{n-i+1} \boldsymbol{e}_j + \dfrac{1/2}{n-i+1} \boldsymbol{b}_1 \right)$ // where $\boldsymbol{e}_j \stackrel{\text{def}}{=} [0 \ \ldots \ 0 \ 1 \ 0 \ \ldots \ 0]^{\mathrm{T}}$ is a vector of all zeros except a single one at location j.
    $S_n \leftarrow N_{n-1} \langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle$
    if unlimited liquidity **then**
        $N_n \leftarrow S_n w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_n)$ // by using (31).
    else
        $N_n \leftarrow S_n w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_n, N_{n-1})$ // by using (39).
    end
end

partial derivative when $d = 3$) is

$$\frac{\partial \ln \langle \boldsymbol{b}, \boldsymbol{x}_{i+1} \rangle}{\partial b^{(j)}} = \frac{x_{i+1}^{(j)} - x_{i+1}^{(j')}}{\langle \boldsymbol{b}, \boldsymbol{x}_{i+1} \rangle}, \tag{B1}$$

where $j' \overset{\text{def}}{=} \begin{cases} d, & \text{if } j = 1 \\ j-1, & \text{otherwise} \end{cases}$. The second-order partial derivative is

$$\frac{\partial^2 \ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle}{\left(\partial b^{(j)}\right)^2} = -\frac{\left(x_{i+1}^{(j)} - x_{i+1}^{(j')}\right)^2}{\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle^2}, \tag{B2}$$

and the second-order mixed partial derivative is

$$\frac{\partial^2 \ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle}{\partial b^{(j)} \partial b^{(k)}} = -\frac{\left(x_{i+1}^{(j)} - x_{i+1}^{(j')}\right)\left(x_{i+1}^{(k)} - x_{i+1}^{(k')}\right)}{\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle^2}, \tag{B3}$$

where $k' \overset{\text{def}}{=} \begin{cases} d, & \text{if } k = 1 \\ k-1, & \text{otherwise} \end{cases}$, and $j \neq k$.

Theorem B.1 *A (twice differentiable) function $\mathbb{R}^d \to \mathbb{R}$ is concave (convex) if and only if its Hessian matrix is negative (positive) semidefinite.*

The Hessian matrix of $\ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle$ when $d = 2$ is

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 \ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle}{\left(\partial b^{(1)}\right)^2} & \frac{\partial^2 \ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle}{\partial b^{(1)} \partial b^{(2)}} \\ \frac{\partial^2 \ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle}{\partial b^{(1)} \partial b^{(2)}} & \frac{\partial^2 \ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle}{\left(\partial b^{(2)}\right)^2} \end{bmatrix} = -\begin{bmatrix} a_1^2 & a_1 a_2 \\ a_1 a_2 & a_2^2 \end{bmatrix}, \tag{B4}$$

where $a_j \overset{\text{def}}{=} \frac{x_{i+1}^{(j)} - x_{i+1}^{(j')}}{\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle}$.

Theorem B.2 *A symmetric matrix $\mathbf{A}$ is negative (positive) semidefinite if and only if all eigenvalues of $\mathbf{A}$ are nonpositive (nonnegative).*

To find the eigenvalues of $\mathbf{H}$, we need to set $\det(\mathbf{H} - \lambda \mathbf{I})$ equal to 0:

$$\det(\mathbf{H} - \lambda \mathbf{I}) = \left(-a_1^2 - \lambda\right)\left(-a_2^2 - \lambda\right) - (a_1 a_2)^2 = 0, \tag{B5}$$

where $\mathbf{I}$ is the identity matrix, and solve (B5) for λ. This results in the eigenvalues of $\mathbf{H}$:

$$\lambda_1 = 0,\ \lambda_2 = -\left(a_1^2 + a_2^2\right) = -\left(\frac{\left(x_{i+1}^{(1)} - x_{i+1}^{(2)}\right)^2}{\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle^2} + \frac{\left(x_{i+1}^{(2)} - x_{i+1}^{(1)}\right)^2}{\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle^2}\right) \leq 0, \tag{B6}$$

which means that $\ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle$ is concave when $d = 2$ by theorem B.1 and B.2, and the eigenvalues of $\mathbf{H}$ for the general case of $d \geq 3$ are

$$\lambda_1 = 0,\ \lambda_2 = 0,\ \ldots,\ \lambda_{d-1} = 0,\ \lambda_d = -\sum_{j=1}^{d} a_j^2 \leq 0, \tag{B7}$$

which means that $\ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle$ is concave for all $d \in \mathbb{Z}_{\geq 2}$ by theorem B.1 and B.2. Consequently, $\sum_{i \in J_n} \ln\langle \boldsymbol{b}, \boldsymbol{x}_{i+1}\rangle$ in (4) is a concave function of $\boldsymbol{b} \in \Delta^{d-1}$ by the fact that the sum of concave (convex) functions is concave (convex).

Appendix C: Concavity of transaction cost factor

Let us check whether TCF w_n in (31) is a concave function of $\boldsymbol{b}_{n+1} \in \Delta^{d-1}$ or not by rewriting (31) as

$$\frac{c_p + c_s}{1 + c_p} \sum_{l \in G_n} \left(\frac{b_n^{(l)} x_n^{(l)}}{\langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle} - b_{n+1}^{(l)} w_n \right) = 1 - w_n, \tag{C1}$$

where the set G_n is defined as $G_n \stackrel{\text{def}}{=} \left\{ j \in \{1, 2, \ldots, d\} \,\middle|\, \frac{b_n^{(j)} x_n^{(j)}}{\langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle} \geq b_{n+1}^{(j)} w_n \right\}$, and G_n has, if there is no trading at the end of the n-th period, the property of $G_n = \{1, 2, \ldots, d\} \Leftrightarrow w_n = 1 \Leftrightarrow b_{n+1}^{(j)} = \frac{b_n^{(j)} x_n^{(j)}}{\langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle}, \forall j$. Equation (C1) can be rewritten again as

$$w_n = \frac{1 - \frac{c_p + c_s}{1 + c_p} \sum_{l \in G_n} \frac{b_n^{(l)} x_n^{(l)}}{\langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle}}{1 - \frac{c_p + c_s}{1 + c_p} \sum_{l \in G_n} b_{n+1}^{(l)}}. \tag{C2}$$

The first-order partial derivative of w_n with respect to $b_{n+1}^{(j)}$ (Figure 9 shows the direction of the partial derivative when $d = 3$) is

$$\frac{\partial w_n}{\partial b_{n+1}^{(j)}} = \begin{cases} \dot{w}_n, & \text{if } j \in G_n \ \wedge \ j' \notin G_n \\ -\dot{w}_n, & \text{if } j \notin G_n \ \wedge \ j' \in G_n \\ 0, & \text{otherwise} \end{cases}, \tag{C3}$$

where

$$\dot{w}_n \stackrel{\text{def}}{=} \frac{\frac{c_p + c_s}{1 + c_p} \left(1 - \frac{c_p + c_s}{1 + c_p} \sum_{l \in G_n} \frac{b_n^{(l)} x_n^{(l)}}{\langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle} \right)}{\left(1 - \frac{c_p + c_s}{1 + c_p} \sum_{l \in G_n} b_{n+1}^{(l)} \right)^2}, \tag{C4}$$

and $j' \stackrel{\text{def}}{=} \begin{cases} d, & \text{if } j = 1 \\ j - 1, & \text{otherwise} \end{cases}$. Hence, w_n is not differentiable with respect to $b_{n+1}^{(j)}$ when a true statement among i) $j \in G_n \ \wedge \ j' \notin G_n$, ii) $j \notin G_n \ \wedge \ j' \in G_n$, and iii) $(j \in G_n \ \wedge \ j' \in G_n) \vee (j \notin G_n \ \wedge \ j' \notin G_n)$ in (C3) changes as $\boldsymbol{b}_{n+1}$ changes. For example, w_n is not differentiable at the maximum point, $\boldsymbol{b}_{n+1}^\star$ in (40), since the left partial derivative:

$$\frac{\partial_- w_n}{\partial b_{n+1}^{(j)}} = \dot{w}_n = \frac{\frac{c_p + c_s}{1 + c_p}}{1 - \frac{c_p + c_s}{1 + c_p}}, \tag{C5}$$

and the right partial derivative:

$$\frac{\partial_+ w_n}{\partial b_{n+1}^{(j)}} = -\dot{w}_n = -\frac{\dfrac{c_p + c_s}{1 + c_p}}{1 - \dfrac{c_p + c_s}{1 + c_p}}, \tag{C6}$$

are not equal. The second-order partial derivative is

$$\frac{\partial^2 w_n}{\left(\partial b_{n+1}^{(j)}\right)^2} = \begin{cases} \ddot{w}_n, & \text{if } (j \in G_n \ \wedge \ j' \notin G_n) \vee (j \notin G_n \ \wedge \ j' \in G_n) \\ 0, & \text{otherwise} \end{cases}, \tag{C7}$$

where

$$\ddot{w}_n \stackrel{\text{def}}{=} \frac{2\left(\dfrac{c_p + c_s}{1 + c_p}\right)^2 \left(1 - \dfrac{c_p + c_s}{1 + c_p} \displaystyle\sum_{l \in G_n} \frac{b_n^{(l)} x_n^{(l)}}{\langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle}\right)}{\left(1 - \dfrac{c_p + c_s}{1 + c_p} \displaystyle\sum_{l \in G_n} b_{n+1}^{(l)}\right)^3}. \tag{C8}$$

The second-order mixed partial derivative is

$$\frac{\partial^2 w_n}{\partial b_{n+1}^{(j)} \partial b_{n+1}^{(k)}} = \begin{cases} \ddot{w}_n, & \text{if } (j \in G_n \wedge j' \notin G_n \wedge k \in G_n \wedge k' \notin G_n) \\ & \vee (j \notin G_n \wedge j' \in G_n \wedge k \notin G_n \wedge k' \in G_n) \\ -\ddot{w}_n, & \text{if } (j \in G_n \wedge j' \notin G_n \wedge k \notin G_n \wedge k' \in G_n) \\ & \vee (j \notin G_n \wedge j' \in G_n \wedge k \in G_n \wedge k' \notin G_n) \\ 0, & \text{otherwise} \end{cases}, \tag{C9}$$

where $k' \stackrel{\text{def}}{=} \begin{cases} d, & \text{if } k = 1 \\ k - 1, & \text{otherwise} \end{cases}$, and $j \neq k$. The concavity of w_n in (31) at differentiable points is determined by theorem B.1 and B.2; the Hessian matrix of w_n at differentiable points when $d = 2$ is

$$\mathbf{H} = \begin{bmatrix} \dfrac{\partial^2 w_n}{\left(\partial b^{(1)}\right)^2} & \dfrac{\partial^2 w_n}{\partial b^{(1)} \partial b^{(2)}} \\ \dfrac{\partial^2 w_n}{\partial b^{(1)} \partial b^{(2)}} & \dfrac{\partial^2 w_n}{\left(\partial b^{(2)}\right)^2} \end{bmatrix} = \begin{bmatrix} \ddot{w}_n & -\ddot{w}_n \\ -\ddot{w}_n & \ddot{w}_n \end{bmatrix}.[1] \tag{C10}$$

To find the eigenvalues of $\mathbf{H}$, we need to set $\det(\mathbf{H} - \lambda \mathbf{I})$ equal to 0:

$$\det(\mathbf{H} - \lambda \mathbf{I}) = (\ddot{w}_n - \lambda)^2 - \ddot{w}_n^2 = 0. \tag{C11}$$

This results in the eigenvalues of $\mathbf{H}$:

$$\lambda_1 = 0, \ \lambda_2 = 2\ddot{w}_n, \tag{C12}$$

[1] Only the two cases are considered: $G_n = \{1\}$ and $G_n = \{2\}$, even though there are four cases: $G_n = \varnothing$, $G_n = \{1\}$, $G_n = \{2\}$, and $G_n = \{1, 2\}$. Firstly, G_n cannot be the empty set because of $G_n = \varnothing \Leftrightarrow \frac{b_n^{(j)} x_n^{(j)}}{\langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle} < b_{n+1}^{(j)}, \forall j$ (i.e. $\frac{b_n^{(j)} x_n^{(j)}}{\langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle} < b_{n+1}^{(j)}, \forall j$ is false by $\boldsymbol{b}_n, \boldsymbol{b}_{n+1} \in \Delta^1$). Secondly, $\boldsymbol{b}_{n+1}^{\star}$ in (40), which satisfies $\boldsymbol{b}_{n+1} = \boldsymbol{b}_{n+1}^{\star} \Leftrightarrow G_n = \{1, 2\}$, is the non-differentiable point as mentioned in (C5) and (C6).

where $\ddot{w}_n > 0$ by the inequalities: $0 < \dfrac{c_p + c_s}{1 + c_p} < 1$, $0 \leq \displaystyle\sum_{l \in G_n} \dfrac{b_n^{(l)} x_n^{(l)}}{\langle \boldsymbol{b}_n, \boldsymbol{x}_n \rangle} \leq 1$, and $0 \leq \displaystyle\sum_{l \in G_n} b_{n+1}^{(l)} \leq 1$. The eigenvalues of $\mathbf{H}$ for the general case of $d \geq 3$ are

$$\lambda_1 = 0, \ \lambda_2 = 0, \ \ldots, \ \lambda_{d-1} = 0, \ \lambda_d = 2\ddot{w}_n. \tag{C13}$$

Consequently, w_n in (31) at differentiable points is convex because all the eigenvalues of the Hessian matrix are nonnegative.

However, w_n is quasi-concave by theorem 5.4: TCF w_n in (31) is a unimodal function of $\boldsymbol{b}_{n+1}$ because w_n strictly decreases as $\boldsymbol{b}_{n+1}$ goes away from the maximum point $\boldsymbol{b}^{\star}_{n+1}$ in (40), as shown in Figure 6(a), 6(c), and 6(e).[1]

In summary, TCF w_n in (31) is a continuous (as shown in Figure 6(b), 6(d), and 6(f)) but not differentiable and quasi-concave but not concave function of $\boldsymbol{b}_{n+1} \in \Delta^{d-1}$.

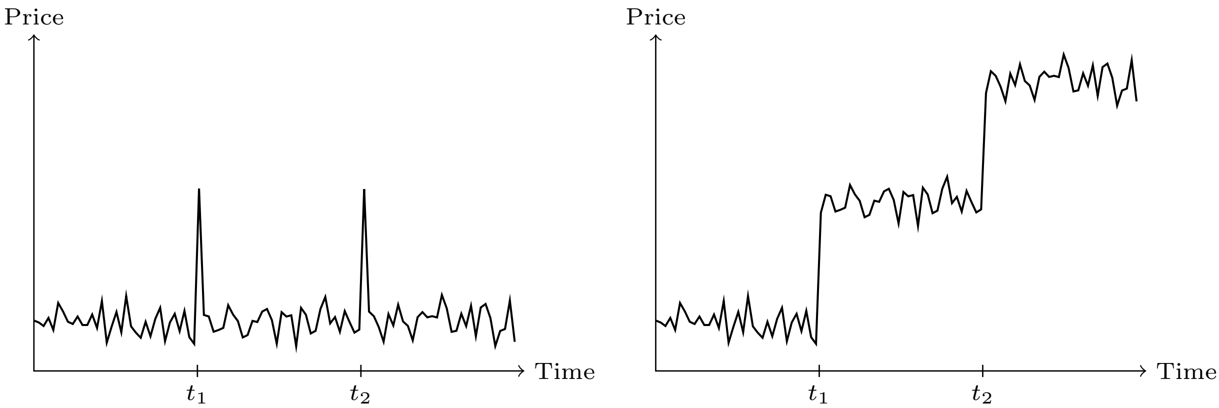
[1] The mathematical proof of the unimodality is not provided in this chapter.

Chapter 3. Algorithmic trading in limit order books for online portfolio selection

1. Introduction

Online portfolio selection (OPS) rebalances a portfolio in every period (e.g. daily or weekly) in order to maximise the portfolio's expected terminal wealth in the long run.[1] It is for a multi-period investment and different from Markowitz mean-variance portfolio selection (MVPS), which is for a single-period investment. OPS aims for higher expected terminal wealth without considering variance (risk), but MVPS considers trade-off between the mean (expected wealth) and variance.

OPS directly optimises a portfolio in terms of the long-term investment without forecasting stock returns (Li and Hoi 2014); hence, OPS does not use in-sample and out-of-sample tests. In contrast, Valle *et al.* (2014a,b, 2015)'s portfolios achieve certain objectives in an in-sample period. Valle *et al.* (2014a) made an absolute return portfolio by conducting the three steps: i) minimising the regression slope of a portfolio, ii) maximising the regression intercept, and iii) minimising transaction costs. Valle *et al.* (2014b) constructed a market-neutral portfolio by not using the pairs trading but minimising the absolute value of the correlation between portfolio return and index return (a constraint that the average return of the optimised portfolio is greater than or equal to that of the benchmark index was set, in order to pursue high returns as well as the zero correlation). Valle *et al.* (2015) constructed a factor-neutral portfolio that minimises the three factors' effect of the Fama–French model.



(a) Temporary impact.

(b) Permanent impact.

Figure 1. Two types of stock price trajectory affected by two subsequent purchases at time t_1 and t_2 (similar graphs are in (Agliardi and Gençay 2014, p. 36)).

A problem of OPS is that it needs to send large market orders whenever it rebalances a large-sized portfolio in every period. However, a very large market order cannot be executed if asset liquidity is limited at the rebalancing time. Even if it is executed, it may cause a permanent, rather than

[1] Literature surveys of OPS were carried out by Das (2014, pp. 22–29), Li and Hoi (2014), and Ha (2017, Section 3).

Table 1. A 5-level limit order book of Microsoft Corporation, traded on NASDAQ, on 21 Jun 2012 at 16:00:00 (downloaded from https://lobsterdata.com/info/DataSamples.php). Bid-ask spread is USD 0.01, and midpoint price is USD 30.135.

	Level	Price (USD)	Volume (shares)
Asks	5	30.18	110,006
	4	30.17	86,886
	3	30.16	65,399
	2	30.15	80,663
	1	30.14	16,600
Bids	-1	30.13	-50,426
	-2	30.12	-83,306
	-3	30.11	-8,506
	-4	30.10	-43,838
	-5	30.09	-167,371

temporary, impact on asset prices (Figure 1 shows an example)[1] and may make OPS strategies unprofitable. This is because OPS is based on an assumption that portfolio rebalancing by OPS in the current period does not affect the stock prices of the portfolio in the next period; the assumption is no longer valid if the permanent impact occurs.

Existing trading algorithms (Almgren and Chriss 2001, Kissell *et al.* 2004, Alfonsi *et al.* 2008, Lorenz 2008, Alfonsi *et al.* 2010, Guéant *et al.* 2012) may be considered for OPS to rebalance a large-sized portfolio. However, they are impractical as they do not use limit order book (LOB; Table 1 is an example) data. Almgren and Chriss (2001), Kissell *et al.* (2004) mathematically models market impact but not considers LOB. Alfonsi *et al.* (2008, 2010) models the shape of LOB as a block or a continuous function rather than employs LOB data. Guéant *et al.* (2012) uses LOB data to calibrate the intensity parameters of trading execution, but they do not directly use LOB data for optimal trading.

Therefore, the aim of this chapter is to develop an algorithmic trading strategy that splits a very large market order into a number of consecutive market orders to minimise overall market impact costs (MICs, also called price impact costs, can be generated by an investor who trades on an asset, pushing the price up when buying the asset and pushing it down while selling (Damodaran 2012, Chapter 5)) by taking into account LOB data. This is because the market can usually absorb these smaller slices, resulting in reduced MICs (Kissell *et al.* 2003, p. 196).

The contributions of this chapter are:

- in Section 5, to propose an intraday trading algorithm, compatible with any OPS method, available for any fund size, by considering real-time LOB data;
- in Section 6, to present backtesting results of the proposed intraday trading strategy for the real-world data (historical NASDAQ LOB data).

Furthermore, it should be noted that this is the first attempt to combine OPS with algorithmic trading.

The rest of this chapter is organised as follows. Section 2 reviews algorithmic trading strategies. Section 3 lists mathematical notations. Section 4 reviews a transaction cost factor (TCF) model which reflects MICs. Section 5 explains the proposed method of optimal intraday trading for multi-asset portfolios. Section 6 presents the backtesting results. Lastly, Section 7 gives a conclusion.

2. Literature review of algorithmic trading

Algorithmic trading is the computerised execution of financial instruments following pre-specified rules and guidelines (Kissell 2013, p. 269). It is classified by Kissell (2013, pp. 17–20) as follows:

[1] The two kinds of market impact, temporary and permanent impact, have been distinguished by Almgren and Chriss (2001).

(i) **arrival price algorithm** that optimises a trading path in order to balance the trade-off between cost and risk at a user-specified level of risk aversion;
(ii) **implementation shortfall algorithm**, similar to the arrival price algorithm, but which incorporates real-time adaptation, while the arrival price algorithm does not (the trading path of implementation shortfall algorithm is updated by real-time data on every intraday trading, while that of arrival price algorithm is determined before trading and does not change during intraday trading);
(iii) **black box algorithm** that searches for profiting opportunities and makes investment decisions based on market signals (e.g. asset prices and trading volume).

2.1. *Arrival price algorithm*

Almgren and Chriss (2001) and Kissell *et al.* (2004) commonly proposed an efficient frontier (akin to the Markowitz efficient frontier in the portfolio theory) in a two-dimensional plane whose axes are i) the expected value of MIC arising from the temporary and permanent market impact (see Figure 1) and ii) its variance, which comes from price volatility. Hence, it allows an investor to choose his or her trading strategy for portfolio management with a user-specified parameter of risk aversion. The difference between (Almgren and Chriss 2001) and (Kissell *et al.* 2004) is how to derive the equation of MIC. The former was derived from consecutive trades, whereas the latter was derived from an aggregate trade.

Alfonsi *et al.* (2008) suggested a trading strategy that splits a very large market order for a single-asset, not a multi-asset portfolio, into a number of consecutive market orders to reduce the expected overall market impact. It determines the size of the individual orders with a parameter of the resilience rate of a block-shaped LOB (it is assumed that an LOB consists of a continuous price distribution of orders with a constant height). However, it does not consider the risk of the price volatility as they assumed traders are risk-neutral unlike (Almgren and Chriss 2001, Kissell *et al.* 2004). Therefore, it minimises the expected value of MIC regardless of the risk.

Alfonsi *et al.* (2010) extended the trading strategy in LOBs of the constant function (Alfonsi *et al.* 2008) to that in LOBs of a general shape function. Alfonsi *et al.* (2010) modelled discrete data of LOB (volume and price in Table 1) as a continuous function of LOB density (volume density at a given price). Both the strategies have the same optimal solution of intermediate orders: $\xi_1 = \xi_2 = \cdots = \xi_{N-1}$, where ξ_n is the size of the market order placed at time t_n, and t_N is the ending time of trading. However, the optimal initial market order ξ_0 for the generally shaped LOBs is expressed as an implicit equation, while that for the block-shaped LOBs is expressed as a closed-form equation.

2.2. *Implementation shortfall algorithm*

A path-dependent (dynamic) trading strategy, whose trading path is updated by real-time data, by Lorenz (2008, Chapter 2–3) is superior in terms of generating a more efficient frontier to the path-independent (static) trading strategy, whose trading path is determined before trading starts, by Almgren and Chriss (2001). The superiority of the dynamic strategy comes from trading faster and reducing the risk of the price volatility for the remaining time in the future if there was a windfall trading gain (i.e. lower trading cost) in the past.

Guéant *et al.* (2012)'s trading strategy is similar to Almgren and Chriss (2001) in terms of liquidating a certain quantity of a single-asset, not a multi-asset portfolio, within a given time horizon. However, they are different in terms of order types and optimisation methods. Guéant *et al.* (2012) sends limit orders by considering both price risk (stock price follows a Brownian motion) and non-execution risk (limit orders are sometimes not executed), whereas Almgren and Chriss (2001) sends market orders by considering the trade-off between price risk and MICs. Guéant *et al.* (2012) uses the Hamilton–Jacobi–Bellman equation to solve the stochastic control problem of optimal liquidation, whereas Almgren and Chriss (2001) uses quadratic programming to construct efficient frontier where the trade-off between price risk and MICs is binding.

2.3. *Black box algorithm*

Avellaneda and Lee (2010) constructed a statistical arbitrage strategy of a market-neutral long–short portfolio, i.e. pairs trading: long 1 dollar in a stock and short β_j dollars in the j-th factor, where a multi-factor regression model decomposes a stock return R into a systematic component $\sum_{j=1}^{m} \beta_j F_j$ (m is the number of factors) and an idiosyncratic component $\tilde{R}$:

$$R = \sum_{j=1}^{m} \beta_j F_j + \tilde{R}. \tag{1}$$

This method generates trading signals of buy, sell, or close of long (short) position by using the mean-reverting property of the long–short portfolio's return.

Cont *et al.* (2010) proposed a Markov model of the short-term dynamics of an LOB. The volume of limit orders (see Table 1) is modeled as a Markov state, where a state transition occurs by a limit order, a market order, or a stop order (cancellation of a limit order). Furthermore, Cont *et al.* (2010) showed an application of this model to high-frequency trading by making a short-term prediction of the mid price. This statistical arbitrage makes the round-trip transaction. It enters a long position when the probability of the mid price increasing is high, and it exits the position either when a profit is secured or when a loss of one tick is made.

Tan *et al.* (2011) explained how to detect stock cycles from historical stock prices. They used mean reverting property of stock prices and provided a learning framework to trade on the cycles. To be specific, long positions are held after detecting troughs of stock cycles, and short positions are held after detecting peaks of stock cycles. In addition, a dynamic asset switching strategy was proposed as the detection of stock cycles is for individual stocks.

The investment strategy by Caldeira and Moura (2013) is also pairs trading like the strategy by Avellaneda and Lee (2010). Their difference is the long–short portfolio. Caldeira and Moura (2013) made a pair of two stocks (their stock selection algorithm is based on the cointegration tests: if two stocks are cointegrated, then it is possible to form a mean-reverting stationary process from a linear combination of stock A and B), while Avellaneda and Lee (2010) made a pair of one stock and multiple exchange-traded funds, or a pair of one stock and factors calculated from the principal component analysis.

Mousavi *et al.* (2014) proposed a multi-tree genetic programming model that i) extracts profitable trading rule bases for a multi-asset portfolio from historical data (daily closing price and transaction volume) and ii) updates the portfolio weights over time. Even though it generates a distinct decision rule for each stock, the rules for multiple stocks are evolved simultaneously, and the correlations among multiple stocks are considered. Besides, the proposed model considers the risks and transaction costs from active trading.

Bendtsen and Peña (2016) developed a single-asset trading algorithm with either long or closed position (i.e. no short selling) by using technical indicators (e.g. the moving average, and the relative strength index). Its goal is to generate buy or sell signals for trading a stock by learning and predicting the stock movement. To be specific, they made gated Bayesian network learn a lower risk investment strategy than the buy-and-hold strategy. The network goes back and forth between the buy and sell phases, and it looks for an opportunity to buy or sell shares.

Xu *et al.* (2017) found an arbitrage opportunity through an empirical analysis of the LOB resiliency of stocks traded on Shenzhen Stock Exchange. Their analysis shows that buy (sell) market orders attract more buy (sell) limit orders especially i) when the bid-ask spread is one tick and ii) when the buy (sell) market order size is less than the best ask (bid) volume (the best ask (bid) volume is the volume of LOB at level 1 (-1); see Table 1).

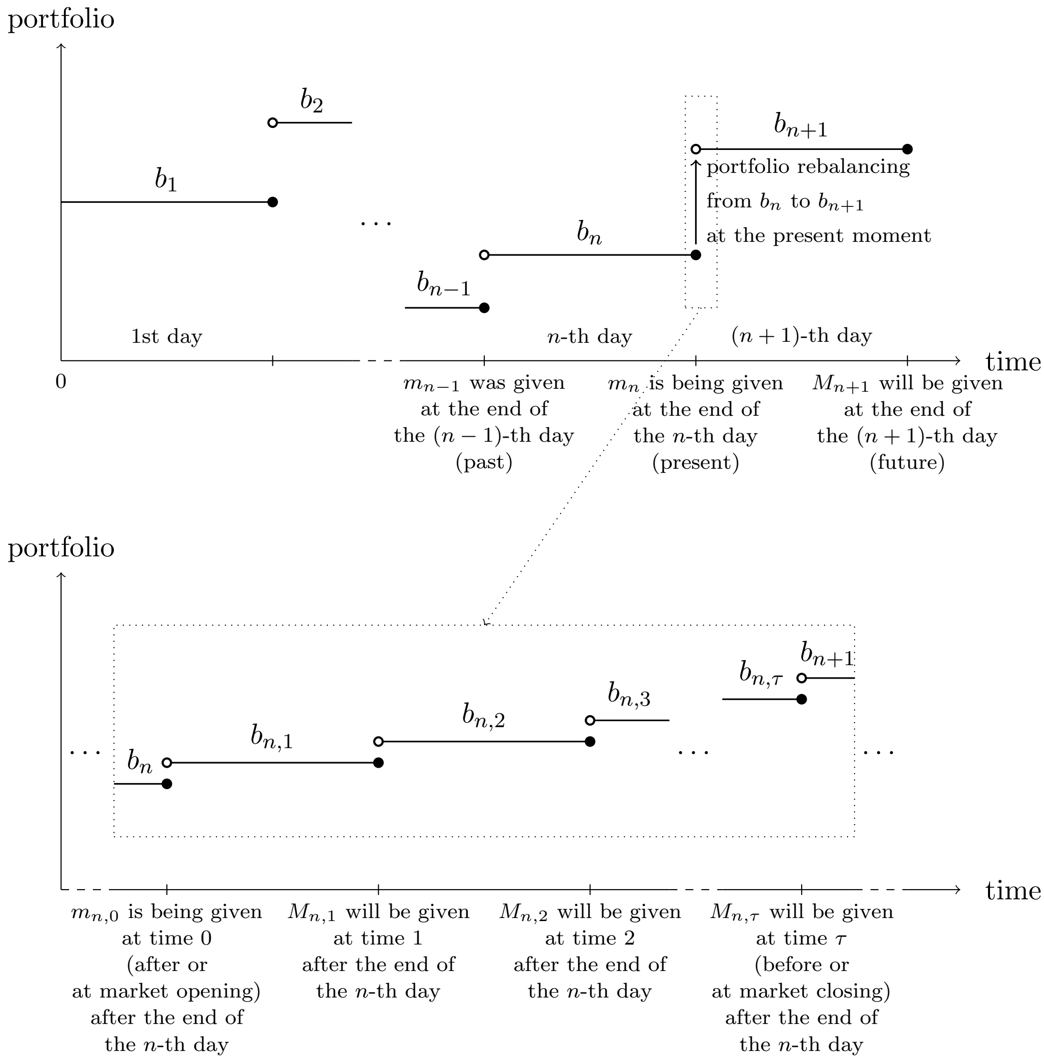


Figure 2. The timeline of intraday trading when the present moment is the end of n-th day. OPS rebalances a portfolio at the end of every trading day (a day ends at time 0, e.g. 9:30 a.m. or 10:00 a.m., not midnight, in this chapter), and an algorithmic trading strategy cushions the shock of the portfolio rebalancing from b_n to b_{n+1}.

Krauss *et al.* (2017) generated daily trading signals from lagged returns of stocks. They conducted nonparametric nonlinear regression between the lagged returns and one-day-ahead return. In particular, the following nonlinear regression methods were employed: deep neural networks, gradient-boosted trees, and random forests. At the last step, a daily portfolio (that is going long for the stocks of higher expected returns and going short for those of lower expected returns) is constructed from the combined signals of the three methods.

3. Notations

The following notations are used in this chapter.

- A lowercase italic letter x indicates a deterministic scalar value, while a capital italic letter X indicates a random variable. A lowercase italic bold letter $\boldsymbol{x}$ indicates a deterministic vector, while a capital italic bold letter $\boldsymbol{X}$ indicates a multivariate random variable (i.e. a random vector). A capital upright bold letter $\mathbf{X}$ denotes a deterministic matrix or a random matrix.
- $\boldsymbol{b}_n = \left[b_n^{(1)}\ b_n^{(2)}\ \ldots\ b_n^{(d)}\right]^{\mathrm{T}}$ is a portfolio vector of d risky assets (there is no risk-free asset in

the portfolio) on the n-th day (see Figure 2), where $n \in \mathbb{Z}_{\geq 1}$, $b_n^{(j)} \in \mathbb{R}_{\geq 0}$ (i.e. neither short selling nor buying stocks on margin is permitted), and $\sum_{j=1}^{d} b_n^{(j)} = 1$ (i.e. $b_n^{(j)}$ is the proportion of a portfolio invested in asset $j \in \{1, 2, \ldots, d\}$ at the n-th day). Hence, $\boldsymbol{b}_n \in \Delta^{d-1}$, where $\Delta^{d-1} = \left\{ \left[b^{(1)}\ b^{(2)}\ \ldots\ b^{(d)} \right]^{\mathrm{T}} \in \mathbb{R}^d_{\geq 0} \,\middle|\, \sum_{j=1}^{d} b^{(j)} = 1 \right\}$ is the standard $(d-1)$-simplex.

- $\boldsymbol{b}_{n,t} \in \Delta^{d-1}$ is an intraday portfolio vector at time t after the end of the n-th day, where $t \in \{0, 1, \ldots, \tau\}$, and τ is the number of intraday tradings (see Figure 2; the portfolio rebalancing from $\boldsymbol{b}_{n,\tau}$ to $\boldsymbol{b}_{n+1}$ is not counted as an intraday trading).
- $\boldsymbol{b}_1 = [1/d\ 1/d\ \ldots\ 1/d]^{\mathrm{T}}$ is an initial portfolio vector.
- A deterministic value $m_n^{(j)}$ (if n is a past or present day), or random variable $M_n^{(j)}$ (if n is a future day) is the mid price of asset j at the end of the n-th day.
- A deterministic value $m_{n,t}^{(j)}$ (if n, t is in the past or at the present), or random variable $M_{n,t}^{(j)}$ (if n, t is in the future) is the mid price of asset j at time t after the end of the n-th day. Technically, $m_{n,0}^{(j)} = m_n^{(j)}$ and $M_{n,0}^{(j)} = M_n^{(j)}$.
- A deterministic value $x_n^{(j)} = \dfrac{m_n^{(j)}}{m_{n-1}^{(j)}}$, or random variable $X_n^{(j)} = \dfrac{M_n^{(j)}}{M_{n-1}^{(j)}}$ $\left(\text{or } X_n^{(j)} = \dfrac{M_n^{(j)}}{m_{n-1}^{(j)}} \right)$ is the relative price of asset j for one day at the end of the n-th day.
- A deterministic vector $\boldsymbol{x}_n = \left[x_n^{(1)}\ x_n^{(2)}\ \ldots\ x_n^{(d)} \right]^{\mathrm{T}} \in \mathbb{R}^d_{>0}$, or multivariate random variable $\boldsymbol{X}_n = \left[X_n^{(1)}\ X_n^{(2)}\ \ldots\ X_n^{(d)} \right]^{\mathrm{T}} \in \mathbb{R}^d_{>0}$ is a market vector, an array of the relative prices of all assets, at the end of the n-th day.
- A deterministic value $x_{n,t}^{(j)} = \dfrac{m_{n,t}^{(j)}}{m_{n,t-1}^{(j)}}$, or random variable $X_{n,t}^{(j)} = \dfrac{M_{n,t}^{(j)}}{M_{n,t-1}^{(j)}}$ $\left(\text{or } X_{n,t}^{(j)} = \dfrac{M_{n,t}^{(j)}}{m_{n,t-1}^{(j)}} \right)$ is the intraday relative price of asset j between time $t-1$ and t after the end of the n-th day, where $m_{n,-1} \overset{\text{def}}{=} m_{n-1,\tau}$, and $M_{n,-1} \overset{\text{def}}{=} M_{n-1,\tau}$.
- A deterministic vector $\boldsymbol{x}_{n,t} = \left[x_{n,t}^{(1)}\ x_{n,t}^{(2)}\ \ldots\ x_{n,t}^{(d)} \right]^{\mathrm{T}} \in \mathbb{R}^d_{>0}$, or multivariate random variable $\boldsymbol{X}_{n,t} = \left[X_{n,t}^{(1)}\ X_{n,t}^{(2)}\ \ldots\ X_{n,t}^{(d)} \right]^{\mathrm{T}} \in \mathbb{R}^d_{>0}$ is an intraday market vector, an array of the intraday relative prices of all assets, at time t after the end of the n-th day.

4. Review of market impact costs and transaction cost factor

4.1. *Market impact costs as a function of order size in a limit order book*

MIC occurs when rebalancing a portfolio, and it can be written as a function of order volumes and prices in LOBs. The average MIC as a function of order size q is defined as (Olsson 2005, Chapter 2.3)

$$\begin{aligned} &\pi(q, m, p_1, p_2, \ldots, p_{-1}, p_{-2}, \ldots, v_1, v_2, \ldots, v_{-1}, v_{-2}, \ldots) \\ &\overset{\text{def}}{=} \frac{|\bar{p}(q, m, p_1, p_2, \ldots, p_{-1}, p_{-2}, \ldots, v_1, v_2, \ldots, v_{-1}, v_{-2}, \ldots) - m|}{m}, \end{aligned} \tag{2}$$

where $m = \frac{p_{-1}+p_1}{2}$ is the midpoint between the best bid and ask price, called mid price. The average price per share for the order size q is defined as

$$\bar{p}(q, m, p_1, p_2, \ldots, p_{-1}, p_{-2}, \ldots, v_1, v_2, \ldots, v_{-1}, v_{-2}, \ldots)$$
$$\stackrel{\text{def}}{=} \begin{cases} \dfrac{\sum\limits_{i=k+1}^{-1} p_i v_i + p_k\left(q - \sum\limits_{i=k+1}^{-1} v_i\right)}{q}, & \text{if } q < v_{-1} \\ p_{-1}, & \text{if } v_{-1} \leq q < 0 \\ m, & \text{if } q = 0 \\ p_1, & \text{if } 0 < q \leq v_1 \\ \dfrac{\sum\limits_{i=1}^{k-1} p_i v_i + p_k\left(q - \sum\limits_{i=1}^{k-1} v_i\right)}{q}, & \text{if } v_1 < q \end{cases}, \tag{3}$$

where positive (negative) q means buying (selling) stocks, p_i and v_i with positive (negative) i are the quoted ask (bid) price and volume at level i, respectively (p_i and v_i correspond to the second and third column of Table 1, respectively, where $v_i \geq 0, v_{-i} \leq 0, \forall i \in \mathbb{Z}_{\geq 1}$), and the highest (lowest) trading level k when $q > v_1$ ($q < v_{-1}$) is

$$k = \left\{ x \in \mathbb{Z}_{\geq 2} \;\middle|\; \sum_{i=1}^{x-1} v_i < q \leq \sum_{i=1}^{x} v_i \right\}, \tag{4a}$$

$$\left(k = \left\{ x \in \mathbb{Z}_{\leq -2} \;\middle|\; \sum_{i=x}^{-1} v_i \leq q < \sum_{i=x+1}^{-1} v_i \right\} \right). \tag{4b}$$

I.e. k represents the level in the order book where the q-th share would be executed.

4.2. *Transaction cost factor with both proportional transaction costs and market impact costs*

The net wealth at time t after the end of the n-th day is a deterministic value $\nu_{n,t}$ if n, t is in the past or at the present, while it is a random variable $N_{n,t}$ if n, t is in the future (the notations of random variables are omitted in the subsequent expressions), and it is defined as

$$\nu_{n,t} \stackrel{\text{def}}{=} s_{n,t} - \gamma_{n,t}, \tag{5}$$

where $s_{n,t}$ is the growth wealth at time t after the end of the n-th day, and $\gamma_{n,t}$ is transaction cost (TC) at time t after the end of the n-th day. The growth wealth $s_{n,t}$ can be calculated from the previous net wealth $\nu_{n,t-1}$:

$$s_{n,t} = \nu_{n,t-1} \sum_{j=1}^{d} b_{n,t}^{(j)} x_{n,t}^{(j)} = \nu_{n,t-1} \langle \boldsymbol{b}_{n,t}, \boldsymbol{x}_{n,t} \rangle, \tag{6}$$

where $\nu_{n,-1} \overset{\text{def}}{=} \nu_{n-1,\tau}$, and $\langle \cdot, \cdot \rangle$ denotes the inner product. Transaction cost factor (TCF) at time t after the end of the n-th day is defined as (Györfi and Vajda 2008)

$$w_{n,t} \overset{\text{def}}{=} \frac{\nu_{n,t}}{s_{n,t}}, \tag{7}$$

where $w_{n,t} \in (0,1] \Leftrightarrow \gamma_{n,t} \in [0, s_{n,t})$.

Let us calculate $w_{n,t}$ when rebalancing from $\boldsymbol{b}_{n,t}$ to $\boldsymbol{b}_{n,t+1}$. If both proportional TCs and MICs are considered, the growth wealth $s_{n,t}$ consists of the sum of the net wealth $\nu_{n,t}$, the MICs, the purchase TCs, and the sale TCs (Ha 2017):

$$s_{n,t} = \nu_{n,t} + \sum_{j=1}^{d} \left(\bar{p}\left(q_{n,t}^{(j)}\right) - m_{n,t}^{(j)}\right) q_{n,t}^{(j)} + c_p \sum_{j=1}^{d} \left(\bar{p}\left(q_{n,t}^{(j)}\right) q_{n,t}^{(j)}\right)^{+} + c_s \sum_{j=1}^{d} \left(-\bar{p}\left(q_{n,t}^{(j)}\right) q_{n,t}^{(j)}\right)^{+}, \tag{8}$$

where $c_p, c_s \in [0,1) : c_p + c_s > 0$ denotes the rate of proportional TCs when purchasing and selling stocks, respectively (some arguments of (3), $m, p_1, p_2, \ldots, p_{-1}, p_{-2}, \ldots, v_1, v_2, \ldots, v_{-1}, v_{-2}, \ldots$ of asset j, are omitted in (8) and the subsequent expressions for notational simplicity), $a^{+} \overset{\text{def}}{=} \max(0, a)$, $q_{n,t}^{(j)}$, an unknown order size of asset j, is

$$q_{n,t}^{(j)} = \frac{b_{n,t+1}^{(j)} s_{n,t} w_{n,t} - b_{n,t}^{(j)} x_{n,t}^{(j)} \nu_{n,t-1}}{m_{n,t}^{(j)}}, \tag{9}$$

and $b_{n,\tau+1}^{(j)} \overset{\text{def}}{=} b_{n+1}^{(j)}$. Equation (8) can be simplified, by the property of $a^{+} = a + (-a)^{+}$, as

$$s_{n,t} = \nu_{n,t} + \sum_{j=1}^{d} \left((1-c_s)\bar{p}\left(q_{n,t}^{(j)}\right) - m_{n,t}^{(j)}\right) q_{n,t}^{(j)} + (c_p + c_s) \sum_{j=1}^{d} \left(\bar{p}\left(q_{n,t}^{(j)}\right) q_{n,t}^{(j)}\right)^{+}, \tag{10}$$

and this can be rewritten, by (7), as

$$w_{n,t} = 1 - \frac{\sum_{j=1}^{d} \left((1-c_s)\bar{p}\left(q_{n,t}^{(j)}\right) - m_{n,t}^{(j)}\right) q_{n,t}^{(j)} + (c_p + c_s) \sum_{j=1}^{d} \left(\bar{p}\left(q_{n,t}^{(j)}\right) q_{n,t}^{(j)}\right)^{+}}{s_{n,t}}, \tag{11}$$

where $w_{0,t} \overset{\text{def}}{=} 1, \forall t \in \{0, 1, \ldots, \tau\}$ (i.e. there are no TCs between time 0 and τ on the 0-th day). Equation (9) and (11) are solvable by using a root-finding algorithm, where $w_{n,t} = w\left(\boldsymbol{b}_{n,t}, \boldsymbol{b}_{n,t+1}, \boldsymbol{x}_{n,t}, \nu_{n,t-1}\right)$ is an unknown variable (c_p and c_s are omitted for notational simplicity).

5. Proposed method of optimal intraday trading

This section explains

(i) how to calculate $\mathbb{E}[S_{n+1}|\mathcal{F}_n]$ in the case of no intraday trading (i.e. $\tau = 0$);
(ii) how to obtain an optimal trading path when rebalancing a portfolio from $\boldsymbol{b}_n$ to $\boldsymbol{b}_{n+1}$, given the number of intraday tradings $\tau \geq 1$ (see Figure 2);
(iii) how to calculate the optimal number of intraday tradings τ^*;
(iv) and how to consider real-time LOB data for optimal intraday trading.

The following assumptions are made for the simplicity of the proposed method:

- asset prices follow the multi-dimensional Brownian motion;
  - hence, the increments of asset prices are jointly normally distributed;
  - and the increments of asset prices are mutually independent for different assets $j \neq j'$ or different trading times $t \neq t'$;
- LOB at time $t' \in \{t+1, t+2, \ldots, \tau\}$ is the same as LOB at time t on the same day.

5.1. *No intraday trading* ($\tau = 0$)

Suppose that a probability space $(\Omega, \mathcal{F}, P)$ equipped with a filtration $\mathcal{F}_n$ is given. If we do not perform intraday trading, then the conditional expected value of the growth wealth at the end of the $(n+1)$-th day, given the past observations, is written, by (6) and (7), as

$$\begin{aligned} \mathbb{E}[S_{n+1}|\mathcal{F}_n] =& \mathbb{E}[N_n\langle \boldsymbol{b}_{n+1}, \boldsymbol{X}_{n+1}\rangle|\mathcal{F}_n] \\ =& \mathbb{E}[S_n W_n\langle \boldsymbol{b}_{n+1}, \boldsymbol{X}_{n+1}\rangle|\mathcal{F}_n] \\ =& \mathbb{E}[S_n W_n|\mathcal{F}_n]\mathbb{E}[\langle \boldsymbol{b}_{n+1}, \boldsymbol{X}_{n+1}\rangle|\mathcal{F}_n] \\ =& s_n w_n \mathbb{E}[\langle \boldsymbol{b}_{n+1}, \boldsymbol{X}_{n+1}\rangle], \end{aligned} \tag{12}$$

where

- $\mathcal{F}_n$ denotes the filtration of the process $\{\boldsymbol{X}_n\}_{n\in\mathbb{Z}_{\geq 1}}$ up to day n,
- N_n is stochastic net wealth at the end of the n-th day,
- $S_n = s_0 \prod_{i=1}^{n} \langle \boldsymbol{b}_i, \boldsymbol{X}_i\rangle$ is stochastic growth wealth at the end of the n-th day with an initial wealth s_0,
- $W_n = w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{X}_n, N_{n-1})$ is stochastic TCF at the end of the n-th day ($S_n W_n$ and $\langle \boldsymbol{b}_{n+1}, \boldsymbol{X}_{n+1}\rangle$ are mutually independent by the assumption that $\boldsymbol{X}_n$ and $\boldsymbol{X}_{n+1}$ are mutually independent),
- $s_n = s_0 \prod_{i=1}^{n} \langle \boldsymbol{b}_i, \boldsymbol{x}_i\rangle$ is deterministic growth wealth at the end of the n-th day,
- $w_n = w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_n, \nu_{n-1})$ is deterministic TCF at the end of the n-th day ($S_n W_n$ is converted to $s_n w_n$ by the conditional expectation given $\mathcal{F}_n$),
- and the market vector $\boldsymbol{X}_{n+1}$ is jointly normally distributed with the mean vector of all ones $[1\ 1\ \ldots\ 1]^{\mathrm{T}}$.

In other words, multi-asset price $\boldsymbol{M}_{n+1}$ follows the multi-dimensional Brownian motion with the zero drift:

$$(\boldsymbol{M}_{n+1} - \boldsymbol{m}_n) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_n), \tag{13}$$

where $\boldsymbol{M}_{n+1} = \left[M_{n+1}^{(1)}\ M_{n+1}^{(2)}\ \ldots\ M_{n+1}^{(d)}\right]^{\mathrm{T}}$ is the random mid-price vector at the end of the $(n+1)$-th day, $\boldsymbol{m}_n = \left[m_n^{(1)}\ m_n^{(2)}\ \ldots\ m_n^{(d)}\right]^{\mathrm{T}}$ is the deterministic mid-price vector at the end of the n-th day, $\boldsymbol{0}$ denotes the all-zero vector (forecasting the expected return of the next trading day is not performed in this chapter), and $\boldsymbol{\Sigma}_n$ is the covariance matrix of price changes between the end of the n-th day and the end of the $(n+1)$-th day. Hence, $\boldsymbol{M}_{n+1}$ is also jointly normally distributed as

$$\boldsymbol{M}_{n+1} \sim \mathcal{N}(\boldsymbol{m}_n, \boldsymbol{\Sigma}_n). \tag{14}$$

As $\boldsymbol{X}_{n+1}$ can be calculated as

$$\boldsymbol{X}_{n+1} = \mathbf{D}^{-1}\boldsymbol{M}_{n+1}, \tag{15}$$

where $\mathbf{D} = \mathrm{diag}(\boldsymbol{m}_n)$, it is jointly normally distributed as

$$\boldsymbol{X}_{n+1} \sim \mathcal{N}(\boldsymbol{1}, \mathbf{D}^{-1}\boldsymbol{\Sigma}_n\mathbf{D}^{-1}). \tag{16}$$

Consequently, $\mathbb{E}[S_{n+1}|\mathcal{F}_n]$ in (12) can be simplified as

$$\begin{aligned}
\mathbb{E}[S_{n+1}|\mathcal{F}_n] =& s_n w_n \mathbb{E}[\langle \boldsymbol{b}_{n+1}, \boldsymbol{X}_{n+1}\rangle] \\
=& s_n w_n \mathbb{E}\left[b_{n+1}^{(1)} X_{n+1}^{(1)} + b_{n+1}^{(2)} X_{n+1}^{(2)} + \cdots + b_{n+1}^{(d)} X_{n+1}^{(d)}\right] \\
=& s_n w_n \left(b_{n+1}^{(1)} \mathbb{E}\left[X_{n+1}^{(1)}\right] + b_{n+1}^{(2)} \mathbb{E}\left[X_{n+1}^{(2)}\right] + \cdots + b_{n+1}^{(d)} \mathbb{E}\left[X_{n+1}^{(d)}\right]\right) \\
=& s_n w_n \sum_{j=1}^{d} b_{n+1}^{(j)} \\
=& s_n w_n.
\end{aligned} \tag{17}$$

5.2. *Single intraday trading ($\tau = 1$)*

If we perform intraday trading only once per trading day, then the conditional expected value of the growth wealth at the end of the $(n+1)$-th day, given the past observations, is written, by (6), (7), and (12), as

$$\mathbb{E}[S_{n+1}|\mathcal{F}_n] = s_n w_n \mathbb{E}[\langle \boldsymbol{b}_{n,1}, \boldsymbol{X}_{n,1}\rangle W_{n,1} \langle \boldsymbol{b}_{n+1}, \boldsymbol{X}_{n+1,0}\rangle], \tag{18}$$

where

- $w_n = w(\boldsymbol{b}_n, \boldsymbol{b}_{n,1}, \boldsymbol{x}_{n,0}, \nu_{n-1,1})$ (w_n is TCF at the end of the n-th day when rebalancing a portfolio from $\boldsymbol{b}_n$ to $\boldsymbol{b}_{n,1}$),
    - $\boldsymbol{x}_{n,0} = \begin{bmatrix} \frac{m_n^{(1)}}{m_{n-1,1}^{(1)}} & \frac{m_n^{(2)}}{m_{n-1,1}^{(2)}} & \cdots & \frac{m_n^{(d)}}{m_{n-1,1}^{(d)}} \end{bmatrix}^{\mathrm{T}}$ ($\boldsymbol{x}_{n,0}$ is not equivalent to $\boldsymbol{x}_n = \begin{bmatrix} \frac{m_n^{(1)}}{m_{n-1}^{(1)}} & \frac{m_n^{(2)}}{m_{n-1}^{(2)}} & \cdots & \frac{m_n^{(d)}}{m_{n-1}^{(d)}} \end{bmatrix}^{\mathrm{T}}$),
- $\boldsymbol{X}_{n,1} = \begin{bmatrix} \frac{M_{n,1}^{(1)}}{m_n^{(1)}} & \frac{M_{n,1}^{(2)}}{m_n^{(2)}} & \cdots & \frac{M_{n,1}^{(d)}}{m_n^{(d)}} \end{bmatrix}^{\mathrm{T}}$,
- $W_{n,1} = w(\boldsymbol{b}_{n,1}, \boldsymbol{b}_{n+1}, \boldsymbol{X}_{n,1}, s_n w_n)$,
- and $\boldsymbol{X}_{n+1,0} = \begin{bmatrix} \frac{M_{n+1}^{(1)}}{M_{n,1}^{(1)}} & \frac{M_{n+1}^{(2)}}{M_{n,1}^{(2)}} & \cdots & \frac{M_{n+1}^{(d)}}{M_{n,1}^{(d)}} \end{bmatrix}^{\mathrm{T}}$ ($\boldsymbol{X}_{n+1,0}$ is not equivalent to $\boldsymbol{X}_{n+1} = \begin{bmatrix} \frac{M_{n+1}^{(1)}}{m_n^{(1)}} & \frac{M_{n+1}^{(2)}}{m_n^{(2)}} & \cdots & \frac{M_{n+1}^{(d)}}{m_n^{(d)}} \end{bmatrix}^{\mathrm{T}}$).

Of course, the MIC function at time 1 after the end of the n-th day to calculate $W_{n,1}$ is stochastic as LOBs are continuously updated by other investors between time 0 and 1. However, the random variables $P_i^{(j)}, V_i^{(j)}, \forall i, j$ in LOBs at time 1 are omitted in (18) for the simple expression, where $P_i^{(j)}$ ($V_i^{(j)}$) is a random variable of the quoted price (volume) of asset j at level i in LOBs.

Under the assumption that the two random vectors, $\boldsymbol{X}_{n,1}$ and $\boldsymbol{X}_{n+1,0}$, are mutually independent, Equation (18) can be rewritten again as

$$\mathbb{E}[S_{n+1}|\mathcal{F}_n] = s_n w_n \mathbb{E}[\langle \boldsymbol{b}_{n,1}, \boldsymbol{X}_{n,1}\rangle W_{n,1}] \mathbb{E}[\langle \boldsymbol{b}_{n+1}, \boldsymbol{X}_{n+1,0}\rangle], \tag{19}$$

but $\langle \boldsymbol{b}_{n,1}, \boldsymbol{X}_{n,1}\rangle$ and $W_{n,1}$ are mutually dependent as $W_{n,1}$ is a function of $\boldsymbol{X}_{n,1}$. Finally, by the property of $\mathbb{E}[\langle \boldsymbol{b}_{n+1}, \boldsymbol{X}_{n+1,0}\rangle] = 1$, proved in (17), $\mathbb{E}[S_{n+1}|\mathcal{F}_n]$ in (18) can be simplified as

$$\mathbb{E}[S_{n+1}|\mathcal{F}_n] = s_n w_n \mathbb{E}[\langle \boldsymbol{b}_{n,1}, \boldsymbol{X}_{n,1}\rangle W_{n,1}]. \tag{20}$$

Our goal is to find the optimal portfolio vector $\boldsymbol{b}^*_{n,1}$ that maximises $\mathbb{E}[S_{n+1}|\mathcal{F}_n]$, and this is a stochastic programming problem as follows:

$$\begin{aligned}
\boldsymbol{b}^*_{n,1} &= \operatorname*{arg\,max}_{\boldsymbol{b}_{n,1}\in\Delta^{d-1}} \mathbb{E}[S_{n+1}|\mathcal{F}_n] \\
&= \operatorname*{arg\,max}_{\boldsymbol{b}_{n,1}\in\Delta^{d-1}} w_n \mathbb{E}[\langle \boldsymbol{b}_{n,1}, \boldsymbol{X}_{n,1}\rangle W_{n,1}] \\
&= \operatorname*{arg\,max}_{\boldsymbol{b}_{n,1}\in\Delta^{d-1}} w_n \int_{\boldsymbol{x}_{n,1}\in\mathbb{R}^d_{>0}} \langle \boldsymbol{b}_{n,1}, \boldsymbol{x}_{n,1}\rangle w_{n,1} f(\boldsymbol{x}_{n,1}) d\boldsymbol{x}_{n,1},
\end{aligned} \tag{21}$$

where $w_{n,1} = w(\boldsymbol{b}_{n,1}, \boldsymbol{b}_{n+1}, \boldsymbol{x}_{n,1}, s_n w_n)$, $f(\boldsymbol{x}_{n,1})$ is the probability density function (PDF) of the multivariate normal distribution $\boldsymbol{X}_{n,1} \sim \mathcal{N}(\boldsymbol{1}, \mathbf{D}^{-1}\boldsymbol{\Sigma}_{n,1}\mathbf{D}^{-1})$, and $\mathbf{D} = \operatorname{diag}(\boldsymbol{m}_n)$ ($\boldsymbol{b}^*_{n,1}$ is not a function of s_n as the growth wealth at the end of the n-th day $s_n = s_0 \prod_{i=1}^n \langle \boldsymbol{b}_i, \boldsymbol{x}_i\rangle$ is independent of $\boldsymbol{b}_{n,1}$). As a result, not only forecasting $\boldsymbol{\Sigma}_{n,1}$, the covariance matrix of price changes between time 0 and 1 after the end of the n-th day (i.e. multivariate intraday volatility),[1] but also calculating the Monte Carlo numerical integration (a closed-form solution of $\mathbb{E}[S_{n+1}|\mathcal{F}_n]$ does not exist as that of $w_{n,1}$ does not) are required to obtain $\boldsymbol{b}^*_{n,1}$.

Therefore, to avoid both the intraday forecasting and the heavy computation (ultimately, to make the stochastic programming problem in (21) simpler), the *expected value solution* of the stochastic programming, a suboptimal solution of $\boldsymbol{b}^*_{n,1}$, is calculated by replacing the random variable $\boldsymbol{X}_{n,1}$ in (21) with its expected value $\mathbb{E}[\boldsymbol{X}_{n,1}] = \boldsymbol{1}$ (Birge and Louveaux 2011, p. 165):

$$\begin{aligned}
\overline{\boldsymbol{b}}^*_{n,1} &= \operatorname*{arg\,max}_{\boldsymbol{b}_{n,1}\in\Delta^{d-1}} w_n \langle \boldsymbol{b}_{n,1}, \boldsymbol{1}\rangle \overline{w}_{n,1} \\
&= \operatorname*{arg\,max}_{\boldsymbol{b}_{n,1}\in\Delta^{d-1}} w_n \overline{w}_{n,1},
\end{aligned} \tag{22}$$

where $\overline{w}_{n,1} = w(\boldsymbol{b}_{n,1}, \boldsymbol{b}_{n+1}, \boldsymbol{1}, s_n w_n)$ (the unimodality of $w_n \overline{w}_{n,1}$ with respect to $\boldsymbol{b}_{n,1} \in \Delta^{d-1}$ is not proved in this chapter although an example is provided in Appendix A; therefore, a local optimum is not guaranteed to be a global optimum). $\overline{\boldsymbol{b}}^*_{n,1}$ is always worse than or equal to $\boldsymbol{b}^*_{n,1}$ in terms of the *value of the stochastic solution* (VSS), defined as the loss by not considering the random variations (Birge and Louveaux 2011, p. 9):

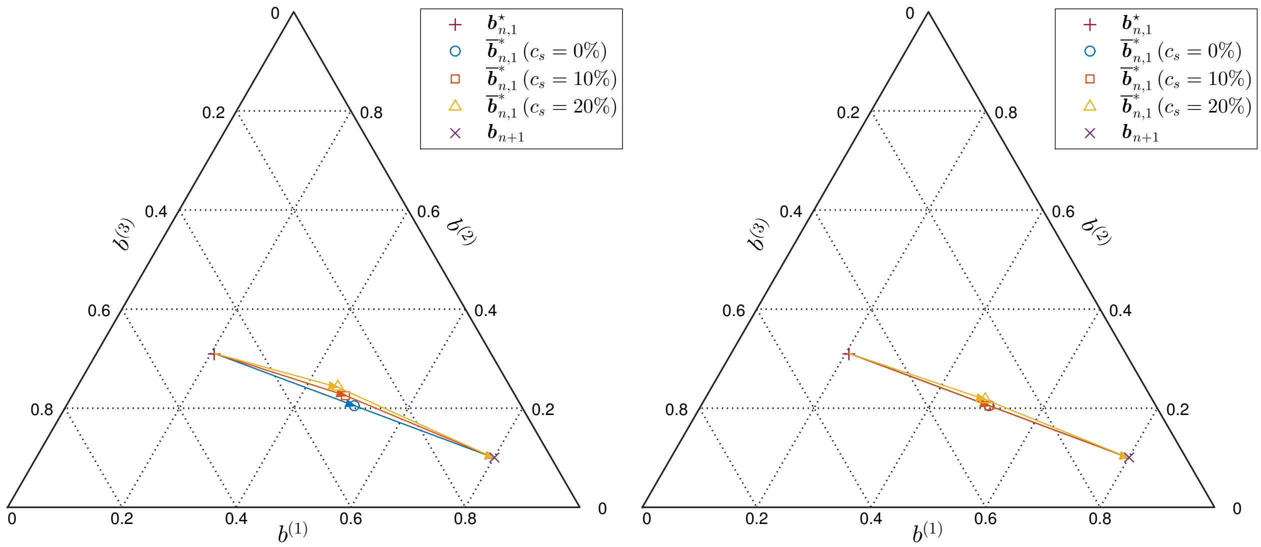
$$w^*_n \mathbb{E}[\langle \boldsymbol{b}^*_{n,1}, \boldsymbol{X}_{n,1}\rangle W^*_{n,1}] - \overline{w}^*_n \mathbb{E}\left[\left\langle \overline{\boldsymbol{b}}^*_{n,1}, \boldsymbol{X}_{n,1}\right\rangle \overline{W}^*_{n,1}\right] \geq 0, \tag{23}$$

where

- $w^*_n = w(\boldsymbol{b}_n, \boldsymbol{b}^*_{n,1}, \boldsymbol{x}_{n,0}, \nu_{n-1,1})$,
- $\overline{w}^*_n = w\left(\boldsymbol{b}_n, \overline{\boldsymbol{b}}^*_{n,1}, \boldsymbol{x}_{n,0}, \nu_{n-1,1}\right)$,
- $W^*_{n,1} = w(\boldsymbol{b}^*_{n,1}, \boldsymbol{b}_{n+1}, \boldsymbol{X}_{n,1}, s_n w^*_n)$,
- and $\overline{W}^*_{n,1} = w\left(\overline{\boldsymbol{b}}^*_{n,1}, \boldsymbol{b}_{n+1}, \boldsymbol{X}_{n,1}, s_n \overline{w}^*_n\right)$

(VSS is always nonnegative for any stochastic program because $\boldsymbol{b}^*_{n,1}$ is an optimal solution, while $\overline{\boldsymbol{b}}^*_{n,1}$ is just one solution of $\operatorname{arg\,max}_{\boldsymbol{b}\in\Delta^{d-1}} \mathbb{E}[S_{n+1}|\mathcal{F}_n]$ (Birge and Louveaux 2011, p. 166)). However,

[1] Intraday (each day is divided into 10-minute intervals) volatility forecasting by using the univariate GARCH model ($\boldsymbol{\Sigma}_{n,1}$ is assumed as a diagonal matrix) was proposed by Engle and Sokalska (2012).



(a) $\nu_{n-1,1}$ is USD 9×10^5.

(b) $\nu_{n-1,1}$ is USD 2×10^6.

Figure 3. The suboptimal path of single intraday trading is dependent on c_p, c_s, and $\nu_{n-1,1}$ ($c_p = 0$, $\boldsymbol{b}_n = [1/3\ 1/3\ 1/3]^{\mathrm{T}}$, $\boldsymbol{b}_{n+1} = [0.8\ 0.1\ 0.1]^{\mathrm{T}}$, and $\boldsymbol{x}_{n,0} = [0.6\ 0.9\ 1.4]^{\mathrm{T}}$). 10-level limit order book data of AAPL ($b^{(1)}$), AMZN ($b^{(2)}$), and GOOG ($b^{(3)}$) on 21 Jun 2012 at 16:00:00 was used.

the suboptimal solution $\overline{\boldsymbol{b}}^{*}_{n,1}$, obtainable by nonlinear programming, has the merits of computational simplicity by neither forecasting the intraday volatility nor conducting the Monte Carlo numerical integration.

$\overline{\boldsymbol{b}}^{*}_{n,1}$ is a function of c_p, c_s, and $\nu_{n-1,1}$ as shown in Figure 3 because w_n and $\overline{w}_{n,1}$ are a function of c_p, c_s, and $\nu_{n-1,1}$. To be specific, each arrow in Figure 3 indicates a suboptimal trading path from the starting point:

$$\boldsymbol{b}^{\star}_{n,1} \overset{\text{def}}{=} \frac{\boldsymbol{b}_n \odot \boldsymbol{x}_{n,0}}{\langle \boldsymbol{b}_n, \boldsymbol{x}_{n,0} \rangle}, \tag{24}$$

where $\odot$ denotes element-wise multiplication of vectors, and $w(\boldsymbol{b}_n, \boldsymbol{b}^{\star}_{n,1}, \boldsymbol{x}_{n,0}, \nu_{n-1,1}) = 1$, to the end point $\boldsymbol{b}_{n+1}$. The starting point is $\boldsymbol{b}^{\star}_{n,1}$, not $\boldsymbol{b}_n$ ($\boldsymbol{b}^{\star}_{n,1}$ is equivalent to $\boldsymbol{b}_n$ if there are no price changes). This is because $\boldsymbol{b}_n$ changes over time as price changes even if we do not rebalance a portfolio, as shown in Figure 11(a). Equation (24) indicates that $b^{\star(j)}_{n,1}$, the weight of asset j, increases (decreases) as $m^{(j)}_{n,0}$, the current price of asset j, has increased (decreased) compared to $m^{(j)}_{n-1,1}$, the previous price of asset j. I.e. if we do not trade at time 1 after the end of the n-th day (this is equivalent to $q^{(j)}_{n,1} = 0, \forall j \Leftrightarrow w(\boldsymbol{b}_n, \boldsymbol{b}_{n,1}, \boldsymbol{x}_{n,0}, \nu_{n-1,1}) = 1$), then $\boldsymbol{b}_{n,1} = \boldsymbol{b}^{\star}_{n,1}$ is satisfied.

5.3. *Multiple intraday tradings ($\tau \geq 2$)*

If we perform intraday trading more than once per day, then the conditional expected value of the growth wealth at the end of the $(n+1)$-th day, given the past observations, is written, by (6), (7), and (12), as

$$\mathbb{E}[S_{n+1}|\mathcal{F}_n] = s_n w_n \mathbb{E}\left[\left(\prod_{t=1}^{\tau} \langle \boldsymbol{b}_{n,t}, \boldsymbol{X}_{n,t} \rangle W_{n,t}\right) \langle \boldsymbol{b}_{n+1}, \boldsymbol{X}_{n+1,0} \rangle\right], \tag{25}$$

where

- $w_n = w(\boldsymbol{b}_n, \boldsymbol{b}_{n,1}, \boldsymbol{x}_{n,0}, \nu_{n-1,\tau})$,

- $\boldsymbol{x}_{n,0} = \left[\frac{m_n^{(1)}}{m_{n-1,\tau}^{(1)}} \ \frac{m_n^{(2)}}{m_{n-1,\tau}^{(2)}} \ \cdots \ \frac{m_n^{(d)}}{m_{n-1,\tau}^{(d)}} \right]^{\mathrm{T}}$,

- $\boldsymbol{X}_{n,t} = \begin{cases} \left[\frac{M_{n,1}^{(1)}}{m_n^{(1)}} \ \frac{M_{n,1}^{(2)}}{m_n^{(2)}} \ \cdots \ \frac{M_{n,1}^{(d)}}{m_n^{(d)}} \right]^{\mathrm{T}}, & \text{if } t = 1 \\ \left[\frac{M_{n,t}^{(1)}}{M_{n,t-1}^{(1)}} \ \frac{M_{n,t}^{(2)}}{M_{n,t-1}^{(2)}} \ \cdots \ \frac{M_{n,t}^{(d)}}{M_{n,t-1}^{(d)}} \right]^{\mathrm{T}}, & \text{if } 2 \leq t \leq \tau \end{cases}$,
- $W_{n,t} = w(\boldsymbol{b}_{n,t}, \boldsymbol{b}_{n,t+1}, \boldsymbol{X}_{n,t}, s_n w_n \prod_{t'=1}^{t-1} \langle \boldsymbol{b}_{n,t'}, \boldsymbol{X}_{n,t'} \rangle W_{n,t'})$,
  - $\prod_{t'=1}^{0}(\cdot) \stackrel{\text{def}}{=} 1$,
  - $\boldsymbol{b}_{n,\tau+1} \stackrel{\text{def}}{=} \boldsymbol{b}_{n+1}$,
- and $\boldsymbol{X}_{n+1,0} = \left[\frac{M_{n+1}^{(1)}}{M_{n,\tau}^{(1)}} \ \frac{M_{n+1}^{(2)}}{M_{n,\tau}^{(2)}} \ \cdots \ \frac{M_{n+1}^{(d)}}{M_{n,\tau}^{(d)}} \right]^{\mathrm{T}}$.

By the assumption that the random vectors, $\boldsymbol{X}_{n,1}, \boldsymbol{X}_{n,2}, \ldots, \boldsymbol{X}_{n,\tau}, \boldsymbol{X}_{n+1,0}$, are mutually independent, Equation (25) can be rewritten as

$$\mathbb{E}[S_{n+1}|\mathcal{F}_n] = s_n w_n \mathbb{E}\left[\prod_{t=1}^{\tau} \langle \boldsymbol{b}_{n,t}, \boldsymbol{X}_{n,t} \rangle W_{n,t} \right] \mathbb{E}[\langle \boldsymbol{b}_{n+1}, \boldsymbol{X}_{n+1,0} \rangle], \tag{26}$$

but $\langle \boldsymbol{b}_{n,t}, \boldsymbol{X}_{n,t} \rangle$ and $W_{n,t}$ are mutually dependent because $W_{n,t}$ is a function of $\boldsymbol{X}_{n,t}$. Also, $\langle \boldsymbol{b}_{n,t}, \boldsymbol{X}_{n,t} \rangle W_{n,t}$ and $\langle \boldsymbol{b}_{n,t'}, \boldsymbol{X}_{n,t'} \rangle W_{n,t'}$ are mutually dependent, where $t \neq t'$, because $W_{n,t}$ is a function of $\boldsymbol{X}_{n,1}, \boldsymbol{X}_{n,2}, \ldots, \boldsymbol{X}_{n,t-1}$ as well as $\boldsymbol{X}_{n,t}$. Finally, by using the property of $\mathbb{E}[\langle \boldsymbol{b}_{n+1}, \boldsymbol{X}_{n+1,0} \rangle] = 1$, proved in (17), $\mathbb{E}[S_{n+1}|\mathcal{F}_n]$ in (25) can be simplified as

$$\mathbb{E}[S_{n+1}|\mathcal{F}_n] = s_n w_n \mathbb{E}\left[\prod_{t=1}^{\tau} \langle \boldsymbol{b}_{n,t}, \boldsymbol{X}_{n,t} \rangle W_{n,t} \right]. \tag{27}$$

Our goal is to find the optimal portfolio vectors $\boldsymbol{b}_{n,1}^*, \boldsymbol{b}_{n,2}^*, \ldots, \boldsymbol{b}_{n,\tau}^*$ that maximises $\mathbb{E}[S_{n+1}|\mathcal{F}_n]$, and this is a stochastic programming problem as follows:

$$\begin{aligned} &\boldsymbol{b}_{n,1}^*, \boldsymbol{b}_{n,2}^*, \ldots, \boldsymbol{b}_{n,\tau}^* \\ &= \operatorname*{arg\,max}_{\boldsymbol{b}_{n,1}, \boldsymbol{b}_{n,2}, \ldots, \boldsymbol{b}_{n,\tau} \in \Delta^{d-1}} \mathbb{E}[S_{n+1}|\mathcal{F}_n] \\ &= \operatorname*{arg\,max}_{\boldsymbol{b}_{n,1}, \boldsymbol{b}_{n,2}, \ldots, \boldsymbol{b}_{n,\tau} \in \Delta^{d-1}} w_n \mathbb{E}\left[\prod_{t=1}^{\tau} \langle \boldsymbol{b}_{n,t}, \boldsymbol{X}_{n,t} \rangle W_{n,t} \right] \\ &= \operatorname*{arg\,max}_{\boldsymbol{b}_{n,1}, \boldsymbol{b}_{n,2}, \ldots, \boldsymbol{b}_{n,\tau} \in \Delta^{d-1}} w_n \int_{\boldsymbol{x}_{n,1} \in \mathbb{R}_{>0}^d} \cdots \int_{\boldsymbol{x}_{n,\tau} \in \mathbb{R}_{>0}^d} \prod_{t=1}^{\tau} \langle \boldsymbol{b}_{n,t}, \boldsymbol{x}_{n,t} \rangle w_{n,t} f(\boldsymbol{x}_{n,t}) d\boldsymbol{x}_{n,1} \cdots d\boldsymbol{x}_{n,\tau}, \end{aligned} \tag{28}$$

where

- $w_{n,t} = w(\boldsymbol{b}_{n,t}, \boldsymbol{b}_{n,t+1}, \boldsymbol{x}_{n,t}, s_n w_n \prod_{t'=1}^{t-1} \langle \boldsymbol{b}_{n,t'}, \boldsymbol{x}_{n,t'} \rangle w_{n,t'})$,
  - $\boldsymbol{b}_{n,\tau+1} \stackrel{\text{def}}{=} \boldsymbol{b}_{n+1}$,
- $f(\boldsymbol{x}_{n,t})$ is the PDF of the multivariate normal distribution $\mathcal{N}\left(\boldsymbol{1}, \mathbf{D}_{n,t-1}^{-1} \boldsymbol{\Sigma}_{n,t} \mathbf{D}_{n,t-1}^{-1}\right)$,
  - and $\mathbf{D}_{n,t-1} = \begin{cases} \operatorname{diag}(\boldsymbol{m}_n), & \text{if } t = 1 \\ \operatorname{diag}(\boldsymbol{M}_{n,t-1}), & \text{if } 2 \leq t \leq \tau \end{cases}$.

($\boldsymbol{b}_{n,1}^*, \boldsymbol{b}_{n,2}^*, \ldots, \boldsymbol{b}_{n,\tau}^*$ is not a function of s_n as the growth wealth at the end of the n-th day $s_n = s_0 \prod_{i=1}^{n} \langle \boldsymbol{b}_i, \boldsymbol{x}_i \rangle$ is independent of $\boldsymbol{b}_{n,1}, \boldsymbol{b}_{n,2}, \ldots, \boldsymbol{b}_{n,\tau}$). As a result, not only forecasting $\boldsymbol{\Sigma}_{n,t}$, the covariance matrix of price changes between time $t-1$ and t after the end of the n-th day (i.e. multivariate intraday volatility), but also calculating the Monte Carlo numerical integration (a closed-form solution of $\mathbb{E}[S_{n+1}|\mathcal{F}_n]$ does not exist as that of $w_{n,t}$ does not) are required to obtain $\boldsymbol{b}^*_{n,1}, \boldsymbol{b}^*_{n,2}, \ldots, \boldsymbol{b}^*_{n,\tau}$.

Therefore, to avoid both the intraday forecasting and the heavy computation (ultimately, to make the stochastic programming problem in (28) simpler), the *expected value solution* of the stochastic programming, a suboptimal solution of $\boldsymbol{b}^*_{n,1}, \boldsymbol{b}^*_{n,2}, \ldots, \boldsymbol{b}^*_{n,\tau}$, is calculated by replacing all the random variables $\boldsymbol{X}_{n,1}, \boldsymbol{X}_{n,2}, \ldots, \boldsymbol{X}_{n,\tau}$ in (28) with their expected values $\mathbb{E}[\boldsymbol{X}_{n,1}] = \mathbb{E}[\boldsymbol{X}_{n,2}] = \ldots = \mathbb{E}[\boldsymbol{X}_{n,\tau}] = \boldsymbol{1}$ (Birge and Louveaux 2011, p. 165):

$$\begin{aligned}\overline{\boldsymbol{b}}^*_{n,1}, \overline{\boldsymbol{b}}^*_{n,2}, \cdots, \overline{\boldsymbol{b}}^*_{n,\tau} &= \underset{\boldsymbol{b}_{n,1}, \boldsymbol{b}_{n,2}, \ldots, \boldsymbol{b}_{n,\tau} \in \Delta^{d-1}}{\arg\max} \quad w_n \prod_{t=1}^{\tau} \langle \boldsymbol{b}_{n,t}, \boldsymbol{1} \rangle \overline{w}_{n,t} \\ &= \underset{\boldsymbol{b}_{n,1}, \boldsymbol{b}_{n,2}, \ldots, \boldsymbol{b}_{n,\tau} \in \Delta^{d-1}}{\arg\max} \quad w_n \prod_{t=1}^{\tau} \overline{w}_{n,t},\end{aligned} \tag{29}$$

where $\overline{w}_{n,t} = w\left(\boldsymbol{b}_{n,t}, \boldsymbol{b}_{n,t+1}, \boldsymbol{1}, s_n w_n \prod_{t'=1}^{t-1} \overline{w}_{n,t'}\right)$, and $\boldsymbol{b}_{n,\tau+1} \stackrel{\text{def}}{=} \boldsymbol{b}_{n+1}$ (the unimodality of $w_n \prod_{t=1}^{\tau} \overline{w}_{n,t}$ with respect to $\boldsymbol{b}_{n,1}, \boldsymbol{b}_{n,2}, \ldots, \boldsymbol{b}_{n,\tau} \in \Delta^{d-1}$ is not proved in this chapter; therefore, a local optimum is not guaranteed to be a global optimum).[1] $\overline{\boldsymbol{b}}^*_{n,1}, \overline{\boldsymbol{b}}^*_{n,2}, \ldots, \overline{\boldsymbol{b}}^*_{n,\tau}$ is always worse than or equal to $\boldsymbol{b}^*_{n,1}, \boldsymbol{b}^*_{n,2}, \ldots, \boldsymbol{b}^*_{n,\tau}$ in terms of the VSS, defined as the loss by not considering the random variations (Birge and Louveaux 2011, p. 9):

$$w^*_n \mathbb{E}\left[\prod_{t=1}^{\tau} \left\langle \boldsymbol{b}^*_{n,t}, \boldsymbol{X}_{n,t} \right\rangle W^*_{n,t}\right] - \overline{w}^*_n \mathbb{E}\left[\prod_{t=1}^{\tau} \left\langle \overline{\boldsymbol{b}}^*_{n,t}, \boldsymbol{X}_{n,t} \right\rangle \overline{W}^*_{n,t}\right] \geq 0, \tag{31}$$

where

- $w^*_n = w(\boldsymbol{b}_n, \boldsymbol{b}^*_{n,1}, \boldsymbol{x}_{n,0}, \nu_{n-1,\tau})$,
- $\overline{w}^*_n = w\left(\boldsymbol{b}_n, \overline{\boldsymbol{b}}^*_{n,1}, \boldsymbol{x}_{n,0}, \nu_{n-1,\tau}\right)$,
- $W^*_{n,t} = w\left(\boldsymbol{b}^*_{n,t}, \boldsymbol{b}^*_{n,t+1}, \boldsymbol{X}_{n,t}, s_n w^*_n \prod_{t'=1}^{t-1} \left\langle \boldsymbol{b}^*_{n,t'}, \boldsymbol{X}_{n,t'} \right\rangle W^*_{n,t'}\right)$,
  - $\boldsymbol{b}^*_{n,\tau+1} \stackrel{\text{def}}{=} \boldsymbol{b}_{n+1}$,
- $\overline{W}^*_{n,t} = w\left(\overline{\boldsymbol{b}}^*_{n,t}, \overline{\boldsymbol{b}}^*_{n,t+1}, \boldsymbol{X}_{n,t}, s_n \overline{w}^*_n \prod_{t'=1}^{t-1} \left\langle \overline{\boldsymbol{b}}^*_{n,t'}, \boldsymbol{X}_{n,t'} \right\rangle \overline{W}^*_{n,t'}\right)$,
  - and $\overline{\boldsymbol{b}}^*_{n,\tau+1} \stackrel{\text{def}}{=} \boldsymbol{b}_{n+1}$.

However, the suboptimal solution $\overline{\boldsymbol{b}}^*_{n,1}, \overline{\boldsymbol{b}}^*_{n,2}, \ldots, \overline{\boldsymbol{b}}^*_{n,\tau}$, obtainable by nonlinear programming, has the merits of computational simplicity by neither forecasting the intraday volatility nor conducting the Monte Carlo numerical integration.

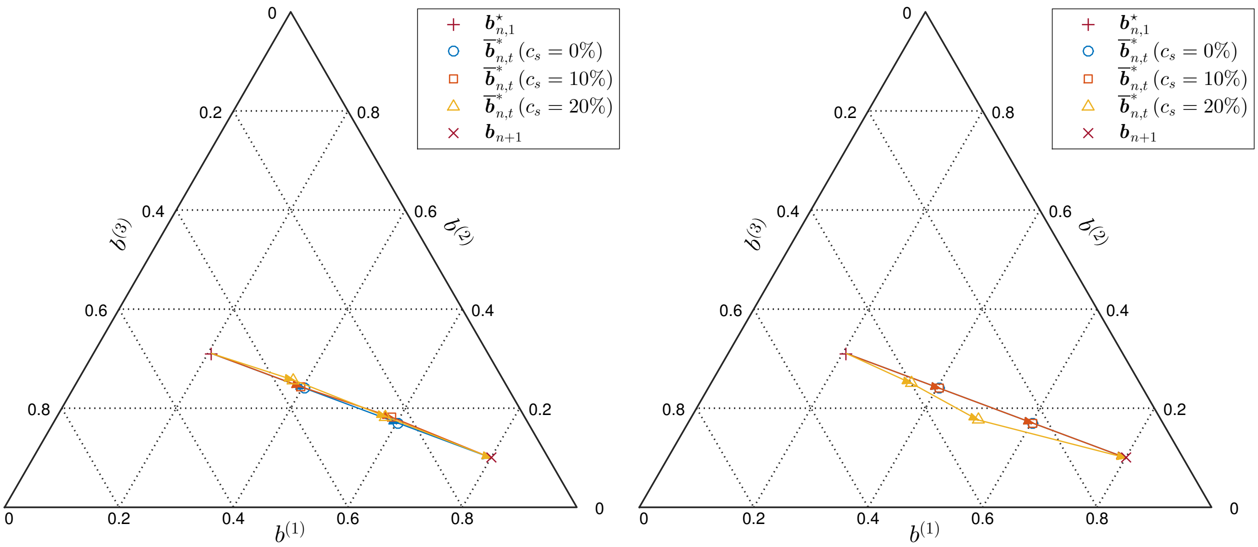
$\overline{\boldsymbol{b}}^*_{n,1}, \overline{\boldsymbol{b}}^*_{n,2}, \ldots, \overline{\boldsymbol{b}}^*_{n,\tau}$ is a function of c_p, c_s, and $\nu_{n-1,\tau}$ as shown in Figure 4 because w_n and $\overline{w}_{n,t}, \forall t \in \{1, 2, \ldots, \tau\}$ are a function of c_p, c_s, and $\nu_{n-1,\tau}$. To be specific, each arrow indicates a suboptimal trading path from the starting point $\boldsymbol{b}^\star_{n,1} = \frac{\boldsymbol{b}_n \odot \boldsymbol{x}_{n,0}}{\langle \boldsymbol{b}_n, \boldsymbol{x}_{n,0} \rangle}$ in (24) to the end point $\boldsymbol{b}_{n+1}$.

---

[1] If forecasting intraday expected return is possible, Equation (29) can be rewritten as the following equation by replacing $\boldsymbol{1}$ to $\mathbb{E}[\boldsymbol{X}_{n,t}]$

$$\overline{\boldsymbol{b}}^*_{n,1}, \overline{\boldsymbol{b}}^*_{n,2}, \cdots, \overline{\boldsymbol{b}}^*_{n,\tau} = \underset{\boldsymbol{b}_{n,1}, \boldsymbol{b}_{n,2}, \ldots, \boldsymbol{b}_{n,\tau} \in \Delta^{d-1}}{\arg\max} \quad w_n \prod_{t=1}^{\tau} \langle \boldsymbol{b}_{n,t}, \mathbb{E}[\boldsymbol{X}_{n,t}] \rangle \overline{w}_{n,t}, \tag{30}$$

where $\overline{w}_{n,t} = w\left(\boldsymbol{b}_{n,t}, \boldsymbol{b}_{n,t+1}, \mathbb{E}[\boldsymbol{X}_{n,t}], s_n w_n \prod_{t'=1}^{t-1} \langle \boldsymbol{b}_{n,t'}, \mathbb{E}[\boldsymbol{X}_{n,t'}] \rangle \overline{w}_{n,t'}\right)$, and $\boldsymbol{b}_{n,\tau+1} \stackrel{\text{def}}{=} \boldsymbol{b}_{n+1}$.



(a) $\nu_{n-1,\tau}$ is USD 9×10^5.

(b) $\nu_{n-1,\tau}$ is USD 3×10^6.

Figure 4. The suboptimal path of two intraday tradings is dependent on c_p, c_s, and $\nu_{n-1,\tau}$ ($c_p = 0$, $\boldsymbol{b}_n = [1/3\ 1/3\ 1/3]^{\mathrm{T}}$, $\boldsymbol{b}_{n+1} = [0.8\ 0.1\ 0.1]^{\mathrm{T}}$, and $\boldsymbol{x}_{n,0} = [0.6\ 0.9\ 1.4]^{\mathrm{T}}$). 10-level limit order book data of AAPL ($b^{(1)}$), AMZN ($b^{(2)}$), and GOOG ($b^{(3)}$) on 21 Jun 2012 at 16:00:00 was used.

A recommended initial value of $\boldsymbol{b}_{n,t}$ for the optimisation of (29) is

$$\boldsymbol{b}^{\dagger}_{n,t} = \boldsymbol{b}^{\star}_{n,1} + \frac{t}{\tau+1}(\boldsymbol{b}_{n+1} - \boldsymbol{b}^{\star}_{n,1}), \tag{32}$$

where $t \in \{1, 2, \ldots, \tau\}$ (i.e. $\boldsymbol{b}^{\dagger}_{n,t}, \forall t$ is linearly located between $\boldsymbol{b}^{\star}_{n,1}$ and $\boldsymbol{b}_{n+1}$ with the same distance). This is because the suboptimal portfolio vector $\overline{\boldsymbol{b}}^{*}_{n,t}, \forall t$ is not far from $\boldsymbol{b}^{\dagger}_{n,t}, \forall t$ as shown in Figure 3 and 4. As a result, the initial value $\boldsymbol{b}^{\dagger}_{n,t}$ will reduce the computation time for searching the solution.

5.4. *Optimal number of intraday tradings*

The optimal number of intraday tradings τ^* can be written as

$$\tau^* = \operatorname*{arg\,max}_{\tau \in \mathbb{Z}_{\geq 0}} \overline{s}_{n+1}(\tau), \tag{33}$$

where $\overline{s}_{n+1}(\tau)$ is the growth wealth at the end of the $(n+1)$-th day i) when a portfolio is rebalanced through the suboptimal trading path $\overline{\boldsymbol{b}}^{*}_{n,1}, \overline{\boldsymbol{b}}^{*}_{n,2}, \ldots, \overline{\boldsymbol{b}}^{*}_{n,\tau}$, given the number of intraday tradings τ, and ii) when neither prices nor LOBs change between the end of the n-th day and the end of the $(n+1)$-th day (see Figure 2). $\overline{s}_{n+1}(\tau)$ can be written as

$$\overline{s}_{n+1}(\tau) = \begin{cases} s_n w(\boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{x}_{n,0}, \nu_{n-1,\tau}), & \text{if } \tau = 0 \\ s_n \overline{w}^{*}_{n} \prod_{t=1}^{\tau} \overline{w}^{*}_{n,t}, & \text{if } \tau \geq 1 \end{cases}, \tag{34}$$

where

- $\overline{w}^{*}_{n} = w\left(\boldsymbol{b}_n, \overline{\boldsymbol{b}}^{*}_{n,1}, \boldsymbol{x}_{n,0}, \nu_{n-1,\tau}\right)$,
- $\overline{w}^{*}_{n,t} = w\left(\overline{\boldsymbol{b}}^{*}_{n,t}, \overline{\boldsymbol{b}}^{*}_{n,t+1}, \boldsymbol{1}, s_n \overline{w}^{*}_{n} \prod_{t'=1}^{t-1} \overline{w}^{*}_{n,t'}\right)$,
  - and $\overline{\boldsymbol{b}}^{*}_{n,\tau+1} \overset{\text{def}}{=} \boldsymbol{b}_{n+1}$.

Also, $\overline{s}_{n+1}(\tau)$ can be rewritten, by (5) and (6), as

$$\overline{s}_{n+1}(\tau) = s_n - \sum_{t=0}^{\tau} \gamma_{n,t}(\tau), \text{[1]} \tag{35}$$

where $\gamma_{n,t}(\tau)$ is TC at time t after the end of the n-th day, given the number of intraday tradings τ. This results, by (10) and the zero price change (i.e. $m_{n,t}^{(j)} = m_n^{(j)}, \forall t \in \{1, 2, \dots, \tau\}$), in

$$\begin{aligned}
&\tau^* \\
&= \underset{\tau \in \mathbb{Z}_{\geq 0}}{\arg\min} \sum_{t=0}^{\tau} \gamma_{n,t}(\tau) \\
&= \underset{\tau \in \mathbb{Z}_{\geq 0}}{\arg\min} \sum_{t=0}^{\tau} \left[\sum_{j=1}^{d} \left((1 - c_s)\bar{p}\left(\overline{q}_{n,t}^{(j)^*}(\tau)\right) - m_n^{(j)} \right) \overline{q}_{n,t}^{(j)^*}(\tau) + (c_p + c_s) \sum_{j=1}^{d} \left(\bar{p}\left(\overline{q}_{n,t}^{(j)^*}(\tau)\right) \overline{q}_{n,t}^{(j)^*}(\tau) \right)^+ \right],
\end{aligned} \tag{36}$$

where $\overline{q}_{n,t}^{(j)^*}(\tau)$ is the order size of asset j when rebalancing a portfolio by following the suboptimal trading path from $\overline{\boldsymbol{b}}_{n,t}^*$ to $\overline{\boldsymbol{b}}_{n,t+1}^*$, given the number of intraday tradings τ.

Equation (36) implies that overall TCs $\sum_{t=0}^{\tau} \gamma_{n,t}(\tau)$, consisting of proportional TCs and MICs, can be minimised when τ is large enough to make $\left|\overline{q}_{n,t}^{(j)^*}(\tau)\right|$ small for all j and all $t \in \{0, 1, \dots, \tau\}$. Also, Equation (36) indicates that the sufficient and necessary condition which minimises $\sum_{t=0}^{\tau} \gamma_{n,t}(\tau)$ is that all assets are traded at the best ask price $p_1^{(j)}$ or the best bid price $p_{-1}^{(j)}$ for all $t \in \{0, 1, \dots, \tau\}$: $\bar{p}\left(\overline{q}_{n,t}^{(j)^*}(\tau^*)\right) = \begin{cases} p_1^{(j)}, & \text{if } \overline{q}_{n,t}^{(j)^*}(\tau^*) > 0 \\ p_{-1}^{(j)}, & \text{if } \overline{q}_{n,t}^{(j)^*}(\tau^*) < 0 \\ m_n^{(j)}, & \text{otherwise} \end{cases}$, as $v_{-1}^{(j)} \leq \overline{q}_{n,t}^{(j)^*}(\tau^*) \leq v_1^{(j)}, \forall t$. Consequently, the overall

---

[1]

$$\begin{aligned}
\overline{s}_{n+1}(\tau) =& \nu_{n,\tau} \langle \boldsymbol{b}_{n+1}, \boldsymbol{1} \rangle \\
=& \nu_{n,\tau} \\
=& s_{n,\tau} - \gamma_{n,\tau} \\
=& \nu_{n,\tau-1} \langle \boldsymbol{b}_{n,\tau}, \boldsymbol{1} \rangle - \gamma_{n,\tau} \\
=& \nu_{n,\tau-1} - \gamma_{n,\tau} \\
=& s_{n,\tau-1} - \gamma_{n,\tau-1} - \gamma_{n,\tau} \\
=& s_n - \sum_{t=0}^{\tau} \gamma_{n,t}.
\end{aligned}$$

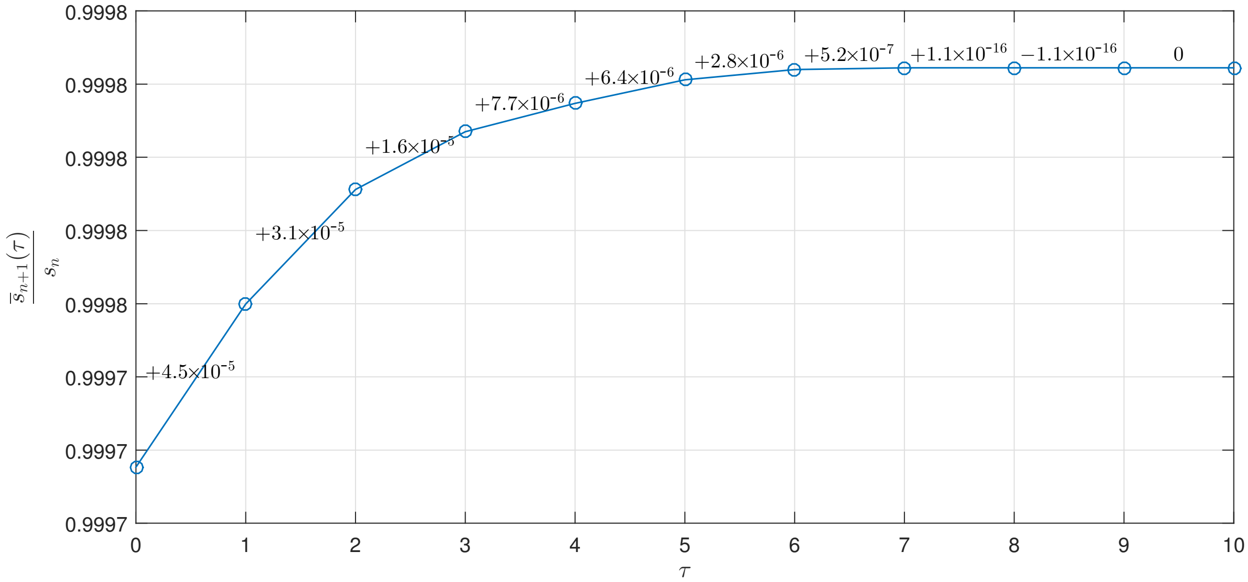


Figure 5. The optimal number of intraday tradings τ^* is not unique ($c_p = 0$, $c_s = 0.00218\%$, $\boldsymbol{b}_n = [1/3\ 1/3\ 1/3]^{\mathrm{T}}$, $\boldsymbol{b}_{n+1} = [0.8\ 0.1\ 0.1]^{\mathrm{T}}$, $\boldsymbol{x}_{n,0} = [0.6\ 0.9\ 1.4]^{\mathrm{T}}$, and $\nu_{n-1,\tau} = 2\times10^6$ USD). 10-level limit order book data of AAPL ($b^{(1)}$), AMZN ($b^{(2)}$), and GOOG ($b^{(3)}$) on 21 Jun 2012 at 16:00:00 was used.

TCs can be minimised when $\tau = \tau^*$ as

$$
\begin{aligned}
&\sum_{t=0}^{\tau^*} \gamma_{n,t}(\tau^*) \\
&= \sum_{t=0}^{\tau^*} \left[\sum_{j=1}^{d} \left((1-c_s)\bar{p}\left(\overline{q}_{n,t}^{(j)*}(\tau^*)\right) - m_n^{(j)} \right) \overline{q}_{n,t}^{(j)*}(\tau^*) + (c_p + c_s) \sum_{j=1}^{d} \left(\bar{p}\left(\overline{q}_{n,t}^{(j)*}(\tau^*)\right) \overline{q}_{n,t}^{(j)*}(\tau^*) \right)^+ \right] \\
&= \sum_{t=0}^{\tau^*} \sum_{j=1}^{d} \left[\left((1-c_s)p_1^{(j)} - m_n^{(j)} \right) \left(\overline{q}_{n,t}^{(j)*}(\tau^*) \right)^+ + \left(m_n^{(j)} - (1-c_s)p_{-1}^{(j)} \right) \left(-\overline{q}_{n,t}^{(j)*}(\tau^*) \right)^+ \right] \\
&\quad + (c_p + c_s) \sum_{t=0}^{\tau^*} \sum_{j=1}^{d} p_1^{(j)} \left(\overline{q}_{n,t}^{(j)*}(\tau^*) \right)^+ \\
&= \sum_{t=0}^{\tau^*} \sum_{j=1}^{d} \left[\left((1+c_p)p_1^{(j)} - m_n^{(j)} \right) \left(\overline{q}_{n,t}^{(j)*}(\tau^*) \right)^+ + \left(m_n^{(j)} - (1-c_s)p_{-1}^{(j)} \right) \left(-\overline{q}_{n,t}^{(j)*}(\tau^*) \right)^+ \right] \\
&= \sum_{j=1}^{d} \left[\left((1+c_p)p_1^{(j)} - m_n^{(j)} \right) \sum_{t=0}^{\tau^*} \left(\overline{q}_{n,t}^{(j)*}(\tau^*) \right)^+ + \left(m_n^{(j)} - (1-c_s)p_{-1}^{(j)} \right) \sum_{t=0}^{\tau^*} \left(-\overline{q}_{n,t}^{(j)*}(\tau^*) \right)^+ \right]
\end{aligned}
\tag{37}
$$

The optimal number of intraday tradings τ^*, a solution of (36), is not unique but $\tau^* \in \{\tau \in \mathbb{Z}_{\geq 0} | \tau \geq \tau^*_{\min}\}$, where $\tau^*_{\min}$ is the minimum optimal number of intraday tradings. This is because both $\sum_{t=0}^{\tau} \left(\overline{q}_{n,t}^{(j)*}(\tau)\right)^+$ and $\sum_{t=0}^{\tau} \left(-\overline{q}_{n,t}^{(j)*}(\tau)\right)^+$ in (37) are constants if $\tau \geq \tau^*_{\min}$. As a result, $\frac{\bar{s}_{n+1}(\tau)}{s_n}$ is a monotonically increasing function of τ as shown in Figure 5. Furthermore, $\frac{\bar{s}_{n+1}(\tau)}{s_n}$ does not change after $\tau^*_{\min} = 7$. Even though $\frac{\bar{s}_{n+1}(\tau)}{s_n}$ increases or decreases in the interval $7 \leq \tau \leq 9$ (each value above the line in Figure 5 is the change amount of $\frac{\bar{s}_{n+1}(\tau)}{s_n}$), the change is considered as numerical error. This is because the tolerance of the optimisation algorithm for solving $\max_{\boldsymbol{b}_{n,1},\boldsymbol{b}_{n,2},\ldots,\boldsymbol{b}_{n,\tau}\in\Delta^{d-1}} w_n \prod_{t=1}^{\tau} \overline{w}_{n,t}$ in (29) was set as $\epsilon = 10^{-6}$. To be specific, if $\left| f\left(\boldsymbol{b}_{n,1}^{k}, \boldsymbol{b}_{n,2}^{k}, \ldots, \boldsymbol{b}_{n,\tau}^{k}\right) - f\left(\boldsymbol{b}_{n,1}^{k+1}, \boldsymbol{b}_{n,2}^{k+1}, \ldots, \boldsymbol{b}_{n,\tau}^{k+1}\right)\right| < \epsilon$, then the iteration of the optimisation stops, where $f\left(\boldsymbol{b}_{n,1}^{k}, \boldsymbol{b}_{n,2}^{k}, \ldots, \boldsymbol{b}_{n,\tau}^{k}\right)$ is the value of the objective function $w_n \prod_{t=1}^{\tau} \overline{w}_{n,t}$ at $\boldsymbol{b}_{n,1}^{k}, \boldsymbol{b}_{n,2}^{k}, \ldots, \boldsymbol{b}_{n,\tau}^{k}$ (the superscript k indicates the iteration number).

Algorithm 1: How to obtain the minimum optimal number of intraday tradings.

Input: $\tau_{\max}$, $\nu_{n-1,\tau}$, $\boldsymbol{x}_{n,0}$, and limit order book data at the end of the n-th day.
Output: the minimum optimal number of intraday tradings $\tau^*_{\min}$.

1 calculate order size $q_{n,0}^{(j)}, \forall j$ when there is no intraday trading by using (9) and (11);
2 **if** $v_{-1}^{(j)} \le q_{n,0}^{(j)} \le v_{1}^{(j)}, \forall j$ **then**
3     $\tau^*_{\min} \leftarrow 0$;
4 **else**
5   **for** $\tau \leftarrow 1$ **to** $\tau_{\max}$ **do**
        `// τ is the number of intraday tradings`
6         $\overline{\boldsymbol{b}}_{n,1}^{*}, \overline{\boldsymbol{b}}_{n,2}^{*}, \ldots, \overline{\boldsymbol{b}}_{n,\tau}^{*} \leftarrow \arg\max_{\boldsymbol{b}_{n,1},\boldsymbol{b}_{n,2},\ldots,\boldsymbol{b}_{n,\tau}\in\Delta^{d-1}} w_n \prod_{t=1}^{\tau} \overline{w}_{n,t}$,
            where $w_n = w(\boldsymbol{b}_n, \boldsymbol{b}_{n,1}, \boldsymbol{x}_{n,0}, \nu_{n-1,\tau})$, $\overline{w}_{n,t} = w\left(\boldsymbol{b}_{n,t}, \boldsymbol{b}_{n,t+1}, \boldsymbol{1}, s_n w_n \prod_{t'=1}^{t-1} \overline{w}_{n,t'}\right)$,
            and $\boldsymbol{b}_{n,\tau+1} \stackrel{\text{def}}{=} \boldsymbol{b}_{n+1}$; `// from (29)`
7   **if** $v_{-1}^{(j)} \le \overline{q}_{n,t}^{(j)*}(\tau) \le v_{1}^{(j)}, \forall j \in \{1, 2, \ldots, d\}, \forall t \in \{0, 1, \ldots, \tau\}$ `// where` $\overline{q}_{n,t}^{(j)*}(\tau)$`, calculable from (9) and (11), is the order size of asset` j `when rebalancing a portfolio from` $\overline{\boldsymbol{b}}_{n,t}^{*}$ `to` $\overline{\boldsymbol{b}}_{n,t+1}^{*}$ `(`$\overline{\boldsymbol{b}}_{n,0}^{*} \stackrel{\text{def}}{=} \boldsymbol{b}_n$ `and` $\overline{\boldsymbol{b}}_{n,\tau+1}^{*} \stackrel{\text{def}}{=} \boldsymbol{b}_{n+1}$`), given the number of intraday tradings` τ
8   **then**
9   break;
10   **end**
11   **end**
12    $\tau^*_{\min} \leftarrow \tau$;
13 **end**

Algorithm 1 describes how to obtain the minimum optimal number of intraday tradings $\tau^*_{\min}$ from the property of $v_{-1}^{(j)} \le \overline{q}_{n,t}^{(j)*}(\tau^*) \le v_{1}^{(j)}, \forall j \in \{1, 2, \ldots, d\}, \forall t \in \{0, 1, \ldots, \tau\}$. This algorithm increases the number of intraday tradings τ from 0 until either when τ equals the upper limit $\tau_{\max}$, a user parameter decided by trading hours and an intraday trading interval (see the 5th line of Algorithm 1), or when trading all assets at the best ask or best bid price is possible (see the 7th line of Algorithm 1).

5.5. *Considering real-time limit order book data*

The proposed method described in Algorithm 2, an implementation shortfall algorithm (an intraday trading strategy is determined by real-time LOB data at every intraday trading time t) for a multi-asset portfolio, performs intraday trading by sending market orders, not limit orders, in the following order.

(i) The portfolio vector of next day $\boldsymbol{b}_{n+1}$ is obtained from an OPS algorithm (see the 2nd line of Algorithm 2) at the end of every trading day (the end of trading day is the market opening, not the midnight; see Figure 2).

Algorithm 2: Proposed method of optimal intraday trading.

Input: s_0, μ, $\tau_{\max}$, where s_0 is an initial wealth, and $(\mu - 1)$ is the number of rebalancing days by online portfolio selection.

1 **for** $n \leftarrow 1$ **to** $\mu - 1$ **do**
2   obtain $\boldsymbol{b}_{n+1}$ from an online portfolio selection algorithm;
3   **for** $t \leftarrow 0$ **to** $\tau_{\max}$ **do**
4   **if** $t = \tau_{\max}$ **then**
5   rebalance a portfolio from $\boldsymbol{b}_{n,t}$ to $\boldsymbol{b}_{n+1}$;
6   break; `// the for-loop of` t
7   **end**
8   receive real-time limit order book data at time t after the end of the n-th day;
9   calculate order size $q_{n,t}^{(j)}, \forall j$ for rebalancing a portfolio from $\boldsymbol{b}_{n,t}$ to $\boldsymbol{b}_{n+1}$ by using (9) and (11);
10   **if** $\left(v_{-1}^{(j)}\right)_{n,t} \leq q_{n,t}^{(j)} \leq \left(v_{1}^{(j)}\right)_{n,t}, \forall j$ `// where` $\left(v_{i}^{(j)}\right)_{n,t}$ `is the quoted volume of limit order book of asset` j `at level` i `at time` t `after the end of the` n`-th day`
11   **then**
12   rebalance a portfolio from $\boldsymbol{b}_{n,t}$ to $\boldsymbol{b}_{n+1}$;
13   break; `// for-loop of` t
14   **else**
15   **for** $\tau \leftarrow t + 1$ **to** $\tau_{\max}$ **do**
16                 $\overline{\boldsymbol{b}}_{n,t+1}^{*}, \overline{\boldsymbol{b}}_{n,t+2}^{*}, \ldots, \overline{\boldsymbol{b}}_{n,\tau}^{*} \leftarrow \arg\max_{\boldsymbol{b}_{n,t+1}, \boldsymbol{b}_{n,t+2}, \ldots, \boldsymbol{b}_{n,\tau} \in \Delta^{d-1}} w_{n,t} \prod_{t'=t+1}^{\tau} \overline{w}_{n,t'}$, where $w_{n,t} = w(\boldsymbol{b}_{n,t}, \boldsymbol{b}_{n,t+1}, \boldsymbol{x}_{n,t}, \nu_{n,t-1})$, $\overline{w}_{n,t'} = w\left(\boldsymbol{b}_{n,t'}, \boldsymbol{b}_{n,t'+1}, \boldsymbol{1}, s_{n,t'-1} w_{n,t'-1} \prod_{t''=t+1}^{t'-1} \overline{w}_{n,t''}\right)$, and $\boldsymbol{b}_{n,\tau+1} \stackrel{\text{def}}{=} \boldsymbol{b}_{n+1}$; `// from` (29)
17   **if** $\left(v_{-1}^{(j)}\right)_{n,t'} \leq \overline{q}_{n,t'}^{(j)\,*} \leq \left(v_{1}^{(j)}\right)_{n,t'}, \forall j \in \{1, 2, \ldots, d\}, \forall t' \in \{t, t+1, \ldots, \tau\}$ `// where` $\overline{q}_{n,t'}^{(j)\,*}$`, calculable from` (9) `and` (11)`, is the order size of asset` j `when rebalancing a portfolio from` $\overline{b}_{n,t'}^{*}$ `to` $\overline{b}_{n,t'+1}^{*}$ `(`$\overline{b}_{n,t}^{*} \stackrel{\text{def}}{=} b_{n,t}$ `and` $\overline{b}_{n,\tau+1}^{*} \stackrel{\text{def}}{=} b_{n+1}$`)`
18   **then**
19   break; `// for-loop of` τ
20   **end**
21   **end**
22   rebalance a portfolio from $\boldsymbol{b}_{n,t}$ to $\overline{\boldsymbol{b}}_{n,t+1}^{*}$;
23   **end**
24   **end**
25 **end**

(ii) The current LOBs of all assets in the portfolio are taken into account at every intraday trading time t until either the time that satisfies $t = \tau_{\max}$ (see the 3rd line of Algorithm 2) or the time that satisfies $\left(v_{-1}^{(j)}\right)_{n,t} \leq q_{n,t}^{(j)} \leq \left(v_{1}^{(j)}\right)_{n,t}, \forall j$ (i.e. whether trading all assets in the current best ask or best bid price is possible or not is checked on every t; see the 10th line of Algorithm 2), whichever happens first.

(iii) If any of the inequalities at the 10th line is false, then the minimum optimal number of intraday tradings $\tau_{\min}^{*}$ is obtained by considering the current LOBs and by using Algorithm 1 (see between the 15th line and the 21st line of Algorithm 2).

(iv) Among the suboptimal path $\overline{\boldsymbol{b}}^*_{n,t+1}, \overline{\boldsymbol{b}}^*_{n,t+2}, \ldots, \overline{\boldsymbol{b}}^*_{n,\tau}$, calculated at the 16th line, only one component $\overline{\boldsymbol{b}}^*_{n,t+1}$ is used for the rebalancing at time t (see the 22nd line of Algorithm 2), but the other components of the suboptimal path $\overline{\boldsymbol{b}}^*_{n,t+2}, \overline{\boldsymbol{b}}^*_{n,t+3}, \ldots, \overline{\boldsymbol{b}}^*_{n,\tau}$ are ignored. This is because new LOBs will be given at time $t+1$ (see the 8th line of Algorithm 2).

(v) The portfolio vector $\overline{\boldsymbol{b}}^*_{n,t+1}$ at time t will be the portfolio vector $\boldsymbol{b}_{n,t}$ at the subsequent time (see the 22nd line of Algorithm 2).

However, Algorithm 2 ignores the risk of intraday price volatility as Alfonsi *et al.* (2008, 2010) did. I.e. the volatility of the random intraday market vectors $\boldsymbol{X}_{n,t+1}, \boldsymbol{X}_{n,t+2}, \ldots, \boldsymbol{X}_{n,\tau}$ is not considered by the assumption that traders are risk-neutral.

6. Simulations (backtesting)

Monte Carlo (MC) simulations consisting of independent trials of random stock selection—each stock has an equal chance of being selected—were conducted to compare the performance between OPS without intraday trading and OPS with the proposed method. To be specific, the number of MC trials is 100, and the number of selected stocks is 30 ($d = 30$). These values are the same as (Ha 2017). 100 is a relatively small number for MC simulations, but the heavy computation to solve the optimisation problem in (29) restricts the number. Also, the numerical results in this section depend on the number. As the greater number of MC simulations is chosen, the more accurate numerical results are obtained. The MATLAB codes of the following experiments have been uploaded on `http://www.mathworks.com/matlabcentral/fileexchange/62503` to avoid any potential ambiguity of the MC simulations. You may leave comments on the web page; any feedback or bug report is welcome.

6.1. *Assumptions for simplicity*

The following assumptions were made for simplicity.

- Assets are arbitrarily divisible (i.e. $q^{(j)}_{n,t} \in \mathbb{R}$ instead of $q^{(j)}_{n,t} \in \mathbb{Z}$) to avoid mixed-integer nonlinear programming.[1]
- Hidden limit orders (HLOs), invisible in limit order books, are never submitted.[2]
- The execution of market orders by OPS at the current time t does not affect the LOBs at the next time $t+1$ (this contradicts the real world but makes this backtesting feasible).
- The computation time to calculate $\boldsymbol{b}_{n,t+1}$ is zero (neither the running time between the 9th line and the 11th line in Algorithm 2 nor that between the 9th line and the 21st line in Algorithm 2 is considered).

Trading at the market opening (9:30 a.m.) was not conducted in this experiment because bid-ask spreads are much higher at the open than mid-day or close as shown in Figure 6 (Figure 6 corresponds to the empirical analysis that bid-ask spreads decrease and level out after about the first 15–30 minutes for large cap stocks and after about 30–60 minutes for small cap stocks by Kissell (2013, pp. 67–69)). This is because the higher bid-ask spreads cause the greater MICs: $\sum_{j=1}^{d} \left(\bar{p}\left(q^{(j)}_{n,t}\right) - m^{(j)}_{n,t} \right) q^{(j)}_{n,t}$ in (8), as well as the greater proportional TCs:

[1] Softwares of mixed-integer nonlinear programming are listed in (Bussieck and Vigerske 2011).

[2] The historical quantity of HLOs can be measured by hidden volume rate, the total volume of trades against hidden orders divided by the total volume of all trades. The mean of this value between 2 Jan 2014 and 30 Sep 2016 is {13.1%, 13.5%} in the case of stocks, not exchange traded products, traded on {NYSE, NASDAQ}, respectively. The daily data of hidden volume rate is downloadable at the homepage of U.S. Securities and Exchange Commission (URL: `http://www.sec.gov/opa/data/market-structure/marketstructuredata-by-exchange.html`).

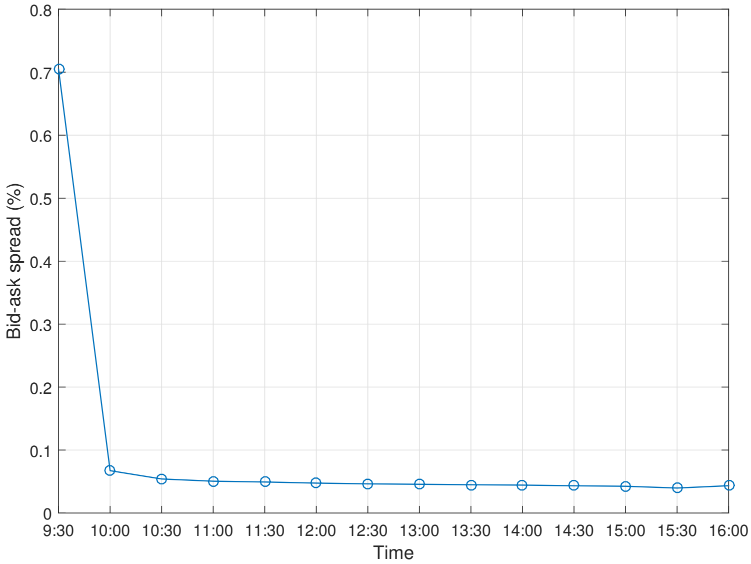


Figure 6. The median values of intraday bid-ask spreads of NASDAQ 100 Index Components (between 1 Jan 2008 and 31 Mar 2016).

$c_p \sum_{j=1}^{d} \left(\bar{p}\left(q_{n,t}^{(j)}\right) q_{n,t}^{(j)}\right)^+ + c_s \sum_{j=1}^{d} \left(-\bar{p}\left(q_{n,t}^{(j)}\right) q_{n,t}^{(j)}\right)^+$ in (8), which in turn deteriorates the performance of OPS methods. As a result, in this experiment, trading starts at 10:00 a.m., and this moment corresponds to time 0 ($t = 0$; see Figure 2).

Trading interval was fixed to 30 minutes under the assumption that LOBs reverts to its normal shape within 30 minutes after the execution of market orders by OPS. This is because Xu *et al.* (2017) empirically showed that the intensity (or rate) of limit order submissions gradually decreases to its normal level within 30 minutes after the execution of market orders in the case of the Shanghai Stock Exchange. Consequently, $\tau_{\max}$ is determined as 12 because trading starts at 10:00 a.m. instead of 9:30 a.m. and because NASDAQ regular market hours ends at 4:00 p.m.[1] (the maximum number of intraday tradings is 12, not 13, since the portfolio rebalancing from $\boldsymbol{b}_{n,\tau}$ to $\boldsymbol{b}_{n+1}$ is not counted as an intraday trading).

6.2. *The source of backtesting data*

10-level (10 levels of the ask side and 10 levels of the bid side, respectively) historical LOB data[2] of NASDAQ 100 Index Components[3] (30 components are randomly selected among the 100 components at each MC trial) between 1 Jan 2008 and 31 Mar 2016 (total 2076 trading days) was downloaded from Limit Order Book System: The Efficient Reconstructor (LOBSTER),[4] and the LOB data was sampled with the period of 30 minutes during NASDAQ regular market hours.

[1] $\tau_{\max}$ is 6 instead of 12 for the following NASDAQ early closing dates (NASDAQ closes at 1:00 p.m.): 3 Jul 2008, 28 Nov 2008, 24 Dec 2008, 27 Nov 2009, 24 Dec 2009, 26 Nov 2010, 25 Nov 2011, 3 Jul 2012, 23 Nov 2012, 24 Dec 2012, 3 Jul 2013, 29 Nov 2013, 24 Dec 2013, 3 Jul 2014, 28 Nov 2014, 24 Dec 2014, 27 Nov 2015, and 24 Dec 2015.

[2] If accessing LOB data at greater than level 10 is required, ask price and volume at level $i \in \mathbb{Z}_{>10}$ are estimated as $p_i = p_{10} + \frac{p_{10} - p_{-1}}{10}(i - 10)$, $v_i = \frac{\sum_{k=1}^{10} v_k}{10}$, respectively. Similarly, if accessing LOB data at less than level -10 is required, bid price and volume at level $i \in \mathbb{Z}_{<-10}$ are estimated as $p_i = p_{-10} + \frac{p_{-10} - p_1}{10}(-i - 10)$, $v_i = \frac{\sum_{k=1}^{10} v_{-k}}{10}$, respectively.

[3] Historical, not current, NASDAQ 100 Index Components on 1 Jan 2008 was downloaded from http://siblisresearch.com/data/historical-components-nasdaq/. In addition, the number of stock candidates used in this experiment is 83 because 17 companies were delisted before 31 Mar 2016 (ALTR, AMLN, BEAS, BRCM, CEPH, DELL, FMCN, GENZ, JAVA, LEAP, NIHD, PETM, SIAL, TLAB, and VMED were acquired by other companies, and FWLT and MIICF were voluntarily delisted from NASDAQ). Therefore, the number of possible portfolio combinations is $\binom{83}{30} = 3.5 \times 10^{22}$, and the portion of $\frac{100}{3.5 \times 10^{22}} = 2.9 \times 10^{-21}$ is covered by the MC simulations.

[4] LOBSTER (https://lobsterdata.com/) has LOB data from 27 Jun 2007 to the present, and the LOB data of LOBSTER does not include hidden LOBs (Huang and Polak 2011, Table 1).

Table 2. A list of online portfolio selection (OPS) strategies.

Category	Author(s)	OPS strategy	Abbreviation
Follow the winner	Cover (1991)	Universal portfolio	UP
	Helmbold *et al.* (1998)	Exponential gradient	EG
	Borodin *et al.* (2000)	Markov of order zero	M0, T0[a]
	Agarwal *et al.* (2006)	Online Newton step	ONS
	Kozat and Singer (2011)	Universal semi-constant rebalanced portfolio	USCRP
Follow the loser	Borodin *et al.* (2004)	Anti-correlation	ANTICOR, ANTICOR_ANTICOR[b]
	Li and Hoi (2012)	Online moving average reversion	OLMAR1, OLMAR2[c]
	Li *et al.* (2012)	Passive aggressive mean reversion	PAMR, PAMR1, PAMR2[d]
	Li *et al.* (2013)	Confidence weighted mean reversion	CWMR_VAR, CWMR_STDEV[e]
Pattern matching	Györfi *et al.* (2006)	Nonparametric kernel-based log-optimal	BK
	Györfi *et al.* (2008)	Nonparametric nearest neighbour log-optimal	BNN
	Li *et al.* (2011)	Correlation-driven nonparametric learning	CORN, CORNU, CORNK[f]

[a]T0 considers the historical market vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ fully, while M0 does not.

[b]ANTICOR_ANTICOR is the twice compounded algorithm of ANTICOR.

[c]OLMAR1 uses a simple (unweighted) moving average, while OLMAR2 uses an exponential (exponentially weighted) moving average.

[d]PAMR1 added a slack variable ξ to the objective function of PAMR, and PAMR2 added ξ^2.

[e]CWMR_STDEV is a modified algorithm of CWMR_VAR to obtain convex constraints (Crammer *et al.* 2008).

[f]CORN: each expert has a weight proportional to its historical performance; CORNU: all experts have the same weight; CORNK: only the K-best experts have weights.

When calculating the relative price of asset j between time τ after the end of the $(n-1)$-th day and the end of the n-th day, cash dividends, stock dividends, and stock splits were considered as

$$x_{n,0}^{(j)} = \frac{m_{n,0}^{(j)} \dfrac{a_{n,\tau_{\max}}^{(j)}}{g_{n,\tau_{\max}}^{(j)}}}{m_{n-1,\tau}^{(j)} \dfrac{a_{n-1,\tau_{\max}}^{(j)}}{g_{n-1,\tau_{\max}}^{(j)}}}, \tag{38}$$

where $m_{n,t}^{(j)}$ is the mid price of asset j at time t after the end of the n-th day from LOBSTER, and $\left\{g_{n,\tau_{\max}}^{(j)},\ a_{n,\tau_{\max}}^{(j)}\right\}$ (the subscript $n, \tau_{\max}$ indicates time $\tau_{\max}$ after the end of the n-th day) is the {closing, adjusted closing} price of asset j of the $(n+1)$-th day, not n-th day, (a day ends after or at the market opening in this chapter; see Figure 2) from Yahoo Finance.

6.3. *Performance comparison among online portfolio selection methods without proposed intraday trading*

The performance (annualised return) of the OPS methods in Table 2 without intraday trading is compared in the case of zero TCs ($c_p = 0$, $c_s = 0$, and zero MICs) as shown in Table 3 and Figure 7,[1] where all the OPS methods rebalanced a portfolio at 10:00 a.m. on every U.S. trading day. In particular, the unpaired two-sample t-tests with unequal variances (simply called t-tests in the remainder of this chapter), whose null hypothesis is that the data in two groups comes from independent random samples from normal distributions with equal means but different variances, were performed to compare the performance between a buy-and-hold (B&H) strategy with the initial

[1] MATLAB programs of OPS by Li and Hoi (2015, Appendix A) were utilised to obtain Table 3 and Figure 7.

Table 3. Statistics of annualised returns of different online portfolio selection methods without intraday trading ($c_p = 0$, $c_s = 0$, and zero market impact costs).

	B&H	UP	EG	ONS	USCRP	ANTICOR	ANTICOR_ANTICOR	OLMAR1	OLMAR2	PAMR	PAMR1
P-value of JB test	0.956	0.450	0.530	0.077	0.570	0.330	0.766	0.048	0.034	0.332	0.332
Standard deviation (%)	1.26	1.50	1.51	11.43	1.32	8.29	11.93	17.37	17.18	16.58	16.58
Mean (%)	9.3	11.4	11.4	23.1	10.4	8.4	6.6	6.2	9.6	9.2	9.2
Difference of means[a] (%)	-	2.11***	2.10***	13.81***	1.08***	−0.88	−2.75**	−3.11*	0.24	−0.12	−0.12
P-value of t-test	-	2.0×10^{-21}	3.4×10^{-21}	3.3×10^{-21}	1.3×10^{-8}	2.9×10^{-1}	2.4×10^{-2}	7.8×10^{-2}	8.9×10^{-1}	9.4×10^{-1}	9.4×10^{-1}

	PAMR2	CWMR_VAR	CWMR_STDEV	BK	BNN	CORN	CORNU	CORNK	M0	T0
P-value of JB test	0.357	0.233	0.231	0.002	0.127	0.672	0.629	0.729	0.445	0.533
Standard deviation (%)	16.53	16.07	15.99	5.39	8.05	9.46	6.34	8.90	3.36	1.53
Mean (%)	9.5	11.1	11.3	8.6	12.2	5.3	10.0	6.0	12.8	11.5
Difference of means[a] (%)	0.16	1.80	1.96	−0.73	2.85***	−3.99***	0.68	−3.30***	3.43***	2.18***
P-value of t-test	9.2×10^{-1}	2.7×10^{-1}	2.2×10^{-1}	1.9×10^{-1}	7.1×10^{-4}	6.0×10^{-5}	3.0×10^{-1}	3.8×10^{-4}	1.3×10^{-16}	3.9×10^{-22}

[a]Difference equals average annualised return of the corresponding OPS method minus that of buy-and-hold (B&H). $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

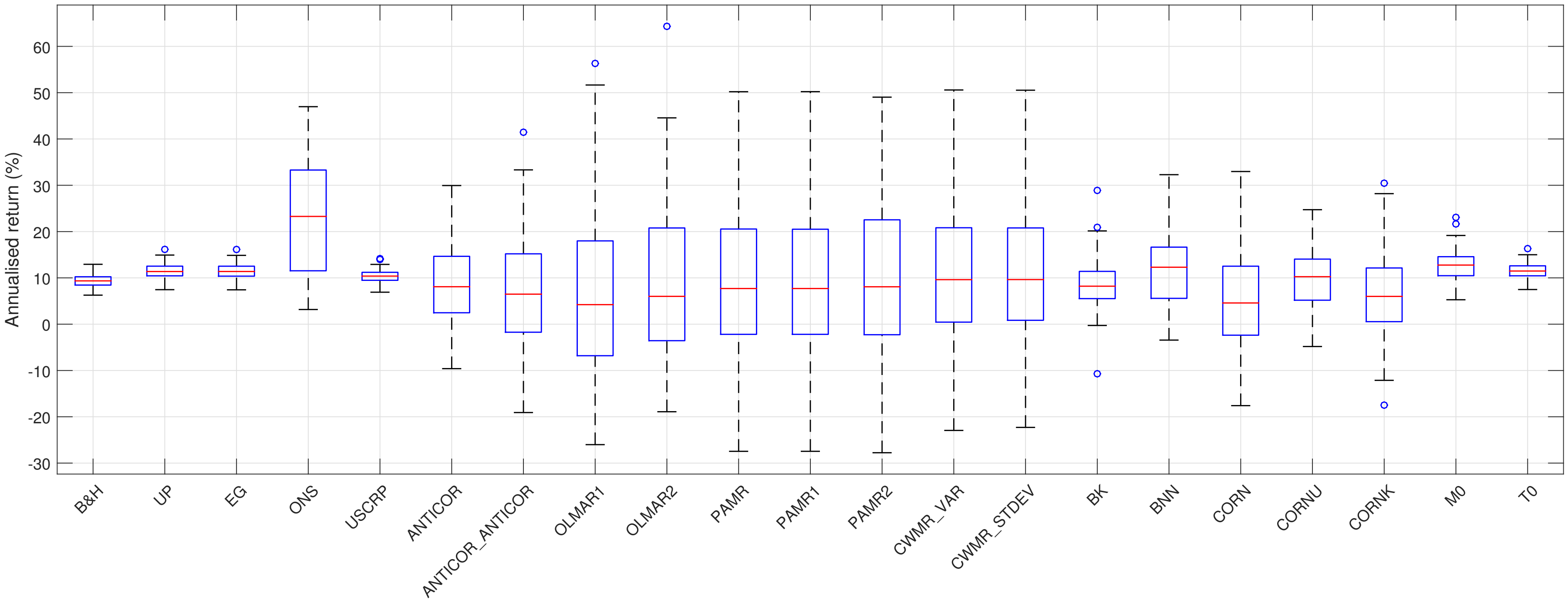


Figure 7. Box plots of annualised returns of different online portfolio selection methods without intraday trading ($c_p = 0$, $c_s = 0$, and zero market impact costs).

portfolio $\boldsymbol{b}_1$ and the OPS methods. Also, the normality assumption of the t-test was confirmed by the Jarque–Bera (JB) test with the significance level of 0.05 as shown in Table 3 except OLMAR1, OLMAR2, and BK. Standard deviation in Table 3 can be interpreted as the sensitivity of the annualised return to the random stock selection. USCRP and OLMAR1 are the least and most sensitive, respectively.

The t-tests answer whether the performance difference is fundamental or whether it is due to random fluctuations (Simon 2013, p. 631). If a p-value of the t-test is less than a significance level, the performance difference is fundamental.[1] All the OPS methods of *follow the winner* and *others* are highly fundamentally superior to B&H as shown in Table 3. On the contrary, all the OPS methods of *follow the loser* and *pattern matching* except BNN are not fundamentally superior to B&H, and some of them are fundamentally inferior to B&H (i.e. the differences of means are negative with low p-values in Table 3). However, it should not be interpreted that the OPS methods of *follow the loser* and *pattern matching* are always inferior to those of *follow the winner*. In other words, the NASDAQ bull market between 2009 and 2016 (see figure 10) was unfavourable to the OPS methods of *follow the loser*.

The performance of each OPS method without TCs and intraday trading in Table 3 and Figure 7 can be considered as the upper limit of that with TCs and intraday trading. This is because OPS carries out long-term investment for every period by using stock prices at $t = 0$ (OPS works only at the end of every trading day as shown in Figure 2), while the proposed method of intraday trading absorbs the shock to the market whenever OPS rebalances a portfolio. In other words, the proposed method minimises the performance gap between OPS without TCs and OPS with TCs, not making additional profits.

6.4. *Performance comparison of online portfolio selection methods between without and with proposed method*

T0, UP, and ONS were selected as the best three OPS methods by referring to the p-values of the t-test in Table 3 for the comparison of each OPS method between without and with the proposed method (PM; Algorithm 2). All of the three methods increase the relative weights of more successful assets in the past periods (these are called *follow the winner* algorithms) as the proportion of asset j or the portfolio vector of each algorithm in the next period is

- T0: $b_{n+1}^{(j)} = \dfrac{c_n^{(j)} + \beta}{d\beta + \sum_{j'=1}^{d} c_n^{(j')}}$, where $c_n^{(j)} = c_{n-1}^{(j)} + \log_2\left(1 + x_n^{(j)}\right)$, and $\beta \in [0, \infty)$ is a parameter;
- UP: $\boldsymbol{b}_{n+1} = \dfrac{\int_{\Delta^{d-1}} \boldsymbol{b} s_n(\boldsymbol{b}, \boldsymbol{x}_{1:n}) d\boldsymbol{b}}{\int_{\Delta^{d-1}} s_n(\boldsymbol{b}, \boldsymbol{x}_{1:n}) d\boldsymbol{b}}$, where, $s_n(\boldsymbol{b}, \boldsymbol{x}_{1:n}) = s_0 \prod_{i=1}^{n} \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle$ is wealth at the end of the n-th period with an initial wealth s_0;
- ONS: $\boldsymbol{b}_{n+1} = \underset{\boldsymbol{b} \in \Delta^{d-1}}{\arg\max} \left(\sum_{i=1}^{n} \ln\langle \boldsymbol{b}, \boldsymbol{x}_i \rangle - \dfrac{\beta}{2} \|\boldsymbol{b}\|^2 \right)$, where $\beta \in [0, \infty)$ is a trade-off parameter between the *follow the winner* term $\sum_{i=1}^{n} \ln\langle \boldsymbol{b}, \boldsymbol{x}_i \rangle$ and the regularisation term $\|\boldsymbol{b}\|^2$.

TCs, consisting of proportional TCs and MICs, were calculated whenever a portfolio was rebalanced for the both cases (without and with the PM) to measure the performance difference. TCs were calculated at 10:00 a.m. on every trading day in the case of OPS without the PM, whereas they were calculated not only at 10:00 a.m. but also between 10:30 a.m. and 4:00 p.m. in the period of 30 minutes on every trading day in the case of OPS with the PM.

The range of the proportional TC rate was set as $c_p = 0$ and $0.00218\% \leq c_s \leq 0.5\%$. This is

[1] The p-value of the t-test is interpreted as the probability that a difference in the mean values would be obtained, given that the population means of two methods are equivalent. The p-value is not equal to the probability that the population means are equivalent (Simon 2013, p. 635).

because the securities transaction tax rates in most of the G20 countries vary between 0.1% and 0.5% (Matheson 2011), and those in the United States in 2016 are $c_p = 0$ and $c_s = 0.218$ basis points.[1] However, stock brokerage commissions were ignored by an assumption that institutional investors, who pay tiny commissions, rather than individual investors, are the main users of OPS.

[1] Order making fiscal year 2016 annual adjustments to transaction fee rates, U.S. Securities and Exchange Commission [Release No. 34-76848/7 Jan 2016].

Table 4. Statistics of annualised returns of different online portfolio selection methods without and with proposed method (PM) ($c_p = 0$, $s_0 = 10^5$ USD)

c_s (%)	B&H	0.00218 T0 w/o PM	0.00218 T0 w/ PM	0.00218 UP w/o PM	0.00218 UP w/ PM	0.00218 ONS w/o PM	0.00218 ONS w/ PM	0.16667 T0 w/o PM	0.16667 T0 w/ PM	0.16667 UP w/o PM	0.16667 UP w/ PM	0.16667 ONS w/o PM	0.16667 ONS w/ PM
P-value of JB test	0.956	0.594	0.617	0.509	0.503	0.076	0.082	0.603	0.585	0.519	0.514	0.073	0.079
Standard deviation (%)	1.26	1.53	1.53	1.50	1.50	10.93	11.22	1.53	1.53	1.50	1.50	10.88	11.12
Mean (%)	9.3	11.3	11.3	11.2	11.2	21.3	22.1	11.0	11.0	11.0	11.0	20.1	20.7
Difference of means[a] (%)	-	1.97***	1.97***	1.92***	1.92***	11.99***	12.73***	1.69***	1.69***	1.65***	1.65***	10.72***	11.36***
P-value of t-test	-	4.6×10^{-19}	4.9×10^{-19}	1.3×10^{-18}	1.3×10^{-18}	8.9×10^{-19}	1.3×10^{-19}	4.9×10^{-15}	4.0×10^{-15}	8.2×10^{-15}	8.2×10^{-15}	2.4×10^{-16}	4.1×10^{-17}
Difference of means[b] (%)	-	-0.00		0.00		0.73		0.01		0.00		0.63	
P-value of t-test	-	0.990		0.999		0.641		0.976		0.999		0.684	

c_s (%)	0.33333 T0 w/o PM	0.33333 T0 w/ PM	0.33333 UP w/o PM	0.33333 UP w/ PM	0.33333 ONS w/o PM	0.33333 ONS w/ PM	0.5 T0 w/o PM	0.5 T0 w/ PM	0.5 UP w/o PM	0.5 UP w/ PM	0.5 ONS w/o PM	0.5 ONS w/ PM
P-value of JB test	0.612	0.595	0.529	0.524	0.070	0.075	0.621	0.606	0.539	0.534	0.067	0.073
Standard deviation (%)	1.53	1.53	1.50	1.50	10.83	11.08	1.53	1.53	1.50	1.50	10.78	11.02
Mean (%)	10.7	10.7	10.7	10.7	18.8	19.4	10.4	10.4	10.4	10.4	17.5	18.1
Difference of means[a] (%)	1.40***	1.40***	1.38***	1.38***	9.45***	10.08***	1.11***	1.11***	1.11***	1.11***	8.18***	8.78***
P-value of t-test	3.0×10^{-11}	2.6×10^{-11}	3.2×10^{-11}	3.2×10^{-11}	7.3×10^{-14}	1.1×10^{-14}	7.3×10^{-8}	6.3×10^{-8}	5.4×10^{-8}	5.3×10^{-8}	2.0×10^{-11}	3.2×10^{-12}
Difference of means[b] (%)	0.01		0.00		0.63		0.01		0.00		0.60	
P-value of t-test	0.977		0.999		0.686		0.977		0.999		0.699	

[a]Difference equals average annualised return of corresponding OPS method minus that of buy-and-hold (B&H). $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

[b]Difference equals average annualised return of corresponding OPS method with proposed method (PM) minus that without PM. $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

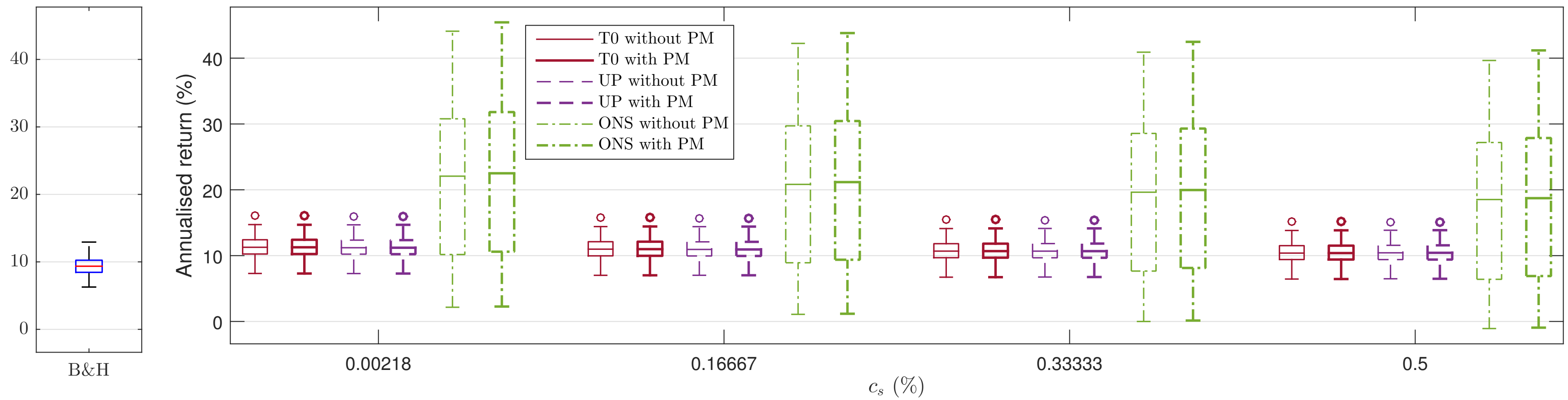


Figure 8. Box plots of annualised returns of different online portfolio selection methods without and with proposed method (PM) ($c_p = 0$, $s_0 = 10^5$ USD).

Table 5. Statistics of annualised returns of different online portfolio selection methods without and with proposed method (PM) ($c_p = 0$, $s_0 = 10^6$ USD)

c_s (%)	BnH	0.00218 T0 w/o PM	0.00218 T0 w/ PM	0.00218 UP w/o PM	0.00218 UP w/ PM	0.00218 ONS w/o PM	0.00218 ONS w/ PM	0.16667 T0 w/o PM	0.16667 T0 w/ PM	0.16667 UP w/o PM	0.16667 UP w/ PM	0.16667 ONS w/o PM	0.16667 ONS w/ PM
P-value of JB test	0.956	0.643	0.552	0.511	0.487	0.059	0.090	0.651	0.564	0.520	0.503	0.057	0.086
Standard deviation (%)	1.26	1.53	1.54	1.50	1.51	10.12	11.23	1.53	1.53	1.50	1.50	10.10	11.17
Mean (%)	9.3	11.3	11.3	11.3	11.3	18.6	22.1	11.0	11.0	11.0	11.0	17.5	20.8
Difference of means[a] (%)	-	1.95***	2.00***	1.93***	1.93***	9.24***	12.74***	1.66***	1.71***	1.66***	1.66***	8.12***	11.51***
P-value of t-test	-	9.2×10^{-19}	2.0×10^{-19}	8.5×10^{-19}	8.8×10^{-19}	9.8×10^{-15}	1.3×10^{-19}	9.2×10^{-15}	2.4×10^{-15}	5.5×10^{-15}	5.9×10^{-15}	2.2×10^{-12}	2.5×10^{-17}
Difference of means[b] (%)	-	0.05		0.00		3.50**		0.05		-0.00		3.39**	
P-value of t-test	-	0.810		0.995		0.022		0.825		1.000		0.025	

c_s (%)	0.33333 T0 w/o PM	0.33333 T0 w/ PM	0.33333 UP w/o PM	0.33333 UP w/ PM	0.33333 ONS w/o PM	0.33333 ONS w/ PM	0.5 T0 w/o PM	0.5 T0 w/ PM	0.5 UP w/o PM	0.5 UP w/ PM	0.5 ONS w/o PM	0.5 ONS w/ PM
P-value of JB test	0.659	0.578	0.530	0.513	0.055	0.082	0.667	0.591	0.540	0.524	0.054	0.079
Standard deviation (%)	1.53	1.53	1.50	1.50	10.08	11.11	1.52	1.53	1.50	1.50	10.06	11.06
Mean (%)	10.7	10.8	10.7	10.7	16.3	19.6	10.4	10.5	10.5	10.4	15.2	18.4
Difference of means[a] (%)	1.37***	1.42***	1.39***	1.39***	6.99***	10.27***	1.09***	1.12***	1.12***	1.11***	5.86***	9.02***
P-value of t-test	5.4×10^{-11}	1.7×10^{-11}	2.2×10^{-11}	2.4×10^{-11}	4.9×10^{-10}	5.5×10^{-15}	1.2×10^{-7}	4.9×10^{-8}	3.9×10^{-8}	4.4×10^{-8}	8.3×10^{-8}	1.2×10^{-12}
Difference of means[b] (%)	0.04		-0.00		3.28**		0.04		-0.00		3.16**	
P-value of t-test	0.841		0.993		0.030		0.856		0.987		0.036	

[a]Difference equals average annualised return of corresponding OPS method minus that of buy-and-hold (B&H). $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

[b]Difference equals average annualised return of corresponding OPS method with proposed method (PM) minus that without PM. $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

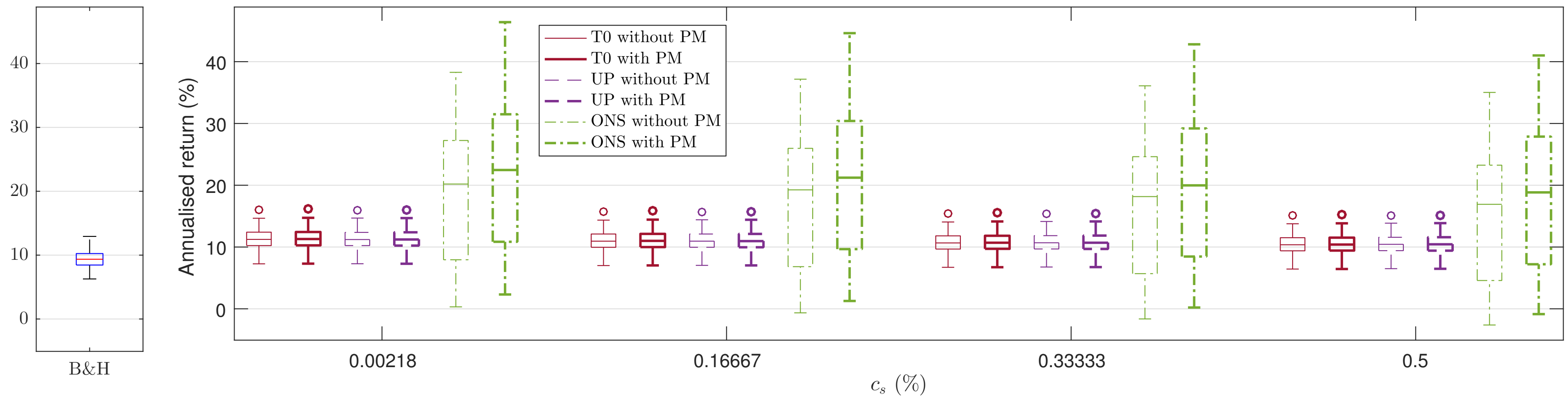


Figure 9. Box plots of annualised returns of different online portfolio selection methods without and with proposed method (PM) ($c_p = 0$, $s_0 = 10^6$ USD).

If initial wealth s_0 is as small as USD 100,000, the performance difference between OPS without and with the PM is not statistically significant, as shown in Table 4 and Figure 8. The statistical significance can be determined by both the sign and p-value of the difference of means of annualised return between without and with the PM. A positive difference with low p-value indicates the PM is useful. However, the lowest p-value (when using ONS at $c_s = 0.00218\%$) is as high as 0.641 as shown in Table 4, which means the PM is not useful for small-sized funds.

However, the PM is useful when initial wealth s_0 is as large as USD 1,000,000 as shown in Table 5 and Figure 9. In particular, the performance difference between ONS without and with the PM is statistically significant. In addition, the performance difference between without and with the PM varies by an OPS method. ONS makes the greater performance gap than T0 and UP. This is because ONS is a more dynamic investment strategy than T0 and UP (this will be shown in Figure 11). I.e. the PM has the more opportunities to reduce TCs when OPS attempts to cause the higher TCs.

The performance difference between without and with the PM (the difference of means in Table 4 and Table 5) is more significant when c_s is small. This is because proportional TCs, $c_p \sum_{j=1}^{d} \left(\bar{p}\left(q_{n,t}^{(j)}\right) q_{n,t}^{(j)}\right)^{+} + c_s \sum_{j=1}^{d} \left(-\bar{p}\left(q_{n,t}^{(j)}\right) q_{n,t}^{(j)}\right)^{+}$ in (8), are dominant compared to MICs, $\sum_{j=1}^{d} \left(\bar{p}\left(q_{n,t}^{(j)}\right) - m_{n,t}^{(j)}\right) q_{n,t}^{(j)}$ in (8), when c_p or c_s is large. Even if a large market order is divided into consecutive intraday market orders by the PM, proportional TCs with the PM are as large as those without the PM. Consequently, the PM is more useful with lower c_p and c_s.

6.5. *Graphical comparisons*

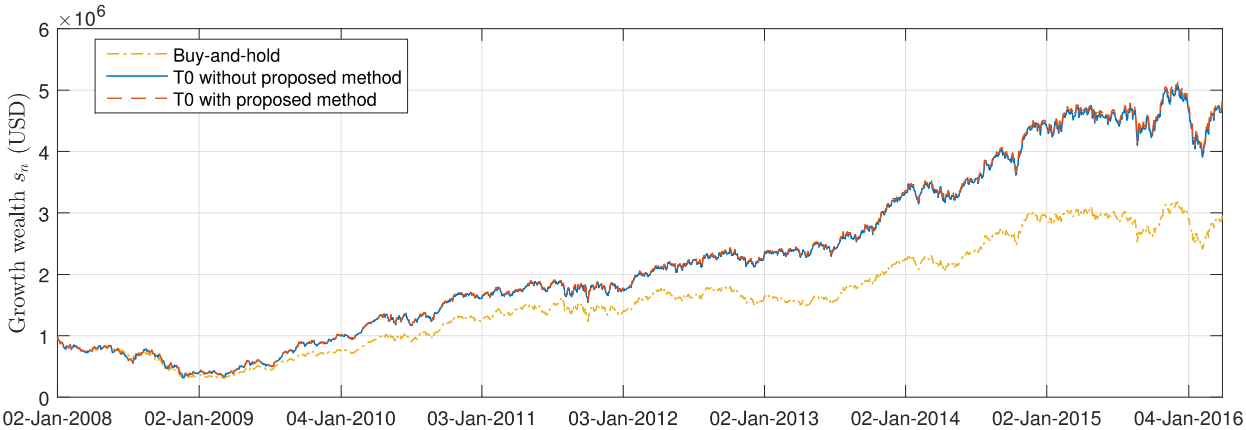
Figure 10 shows growth wealth s_n of a portfolio of six stocks with initial wealth of a million US dollars. The PM of intraday trading has little value in the case of T0 and UP as shown in Figure 10(a) and Figure 10(b). However, The PM works well when ONS is used as shown in Figure 10(c). These correspond to the performance difference between without and with the PM in Table 5.

Figure 11 shows the proportion of portfolio (the portfolio vector $\boldsymbol{b}_n$) that made the growth wealth in Figure 10, in the form of area plots ($\boldsymbol{b}_n$ is independent of the usage of an intraday trading algorithm). The portfolio vector of B&H changes over time, as shown in Figure 11(a), as the prices of assets change over time. T0 generates almost constant portfolio $\boldsymbol{b}_n = \boldsymbol{b}_1$ as shown in Figure 11(b), and UP generates rougher portfolio weights over time than those of T0, but smoother changes than those of B&H, as shown in Figure 11(c). ONS makes the most abrupt changes of portfolio weights over time as shown in Figure 11(d).

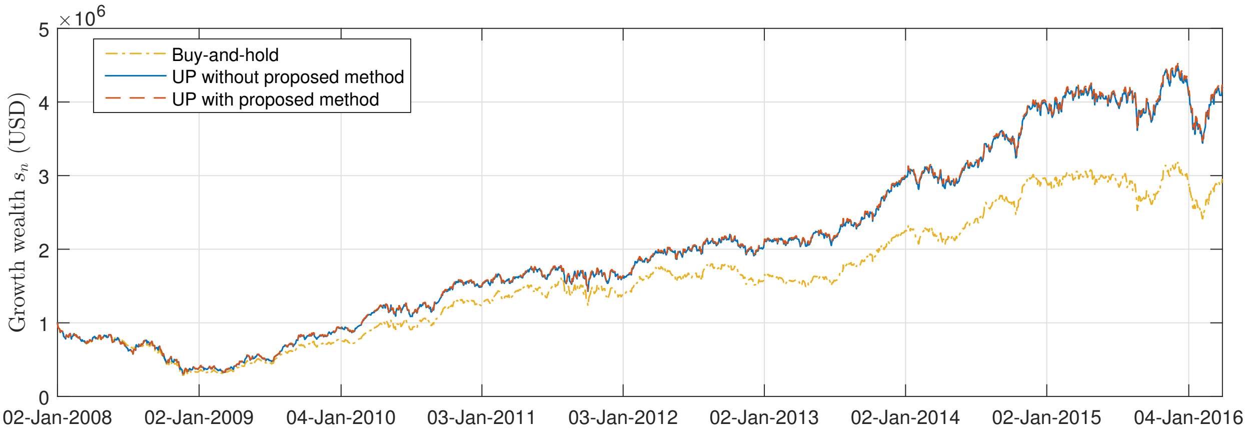
Figure 12 shows how much the PM can decrease TCs, consisting of both proportional TCs and MICs, when following the growth wealth in Figure 10. Its TC reduction is not significant in the case of T0 and UP as shown in Figure 12(a) and Figure 12(b) since both T0 and UP do not trade too much. On the contrary, it saves a lot of TCs for a high-volume trading algorithm like ONS as shown in Figure 12(c).

6.6. *Computation time*

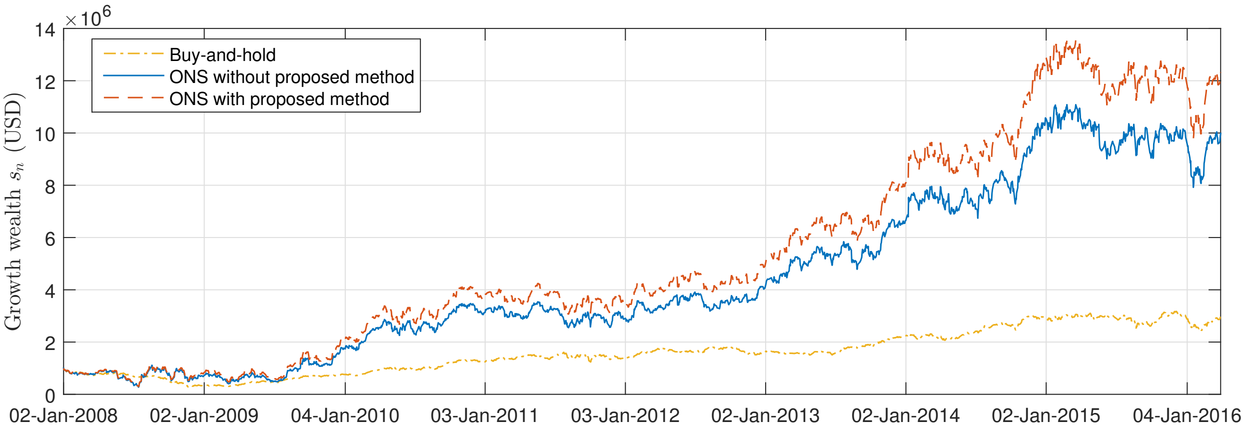
Mean computation time of the PM depends on an OPS strategy, initial wealth s_0, and intraday trading time t as shown in Figure 13. Firstly, the PM requires more computation time for the higher-volume trading algorithms (e.g. ONS) than the lower-volume trading algorithms (e.g. T0 and UP), as shown in both Figure 13(a) and Figure 13(b). Secondly, the PM requires more computation time for the bigger-sized funds (see Figure 13(b)) than the smaller-sized funds (see Figure 13(a)). Thirdly, the computation time changes over time in a day. To be specific, the computation time of the PM with ONS decreases as time goes by during the NASDAQ trading hour, as shown in Figure 13(b). This is because Algorithm 2 has the iteration for $(\tau_{\max} - t)$ times for each t to calculate



(a) T0.



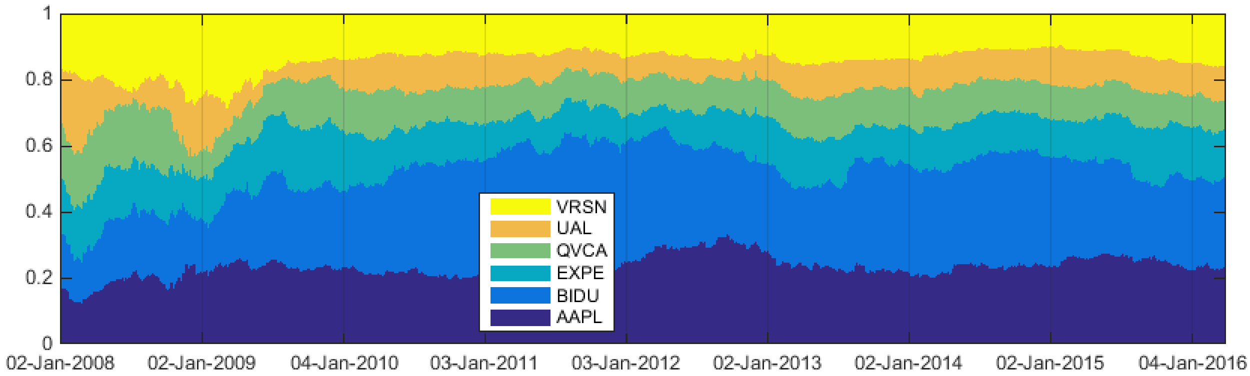
(b) UP.



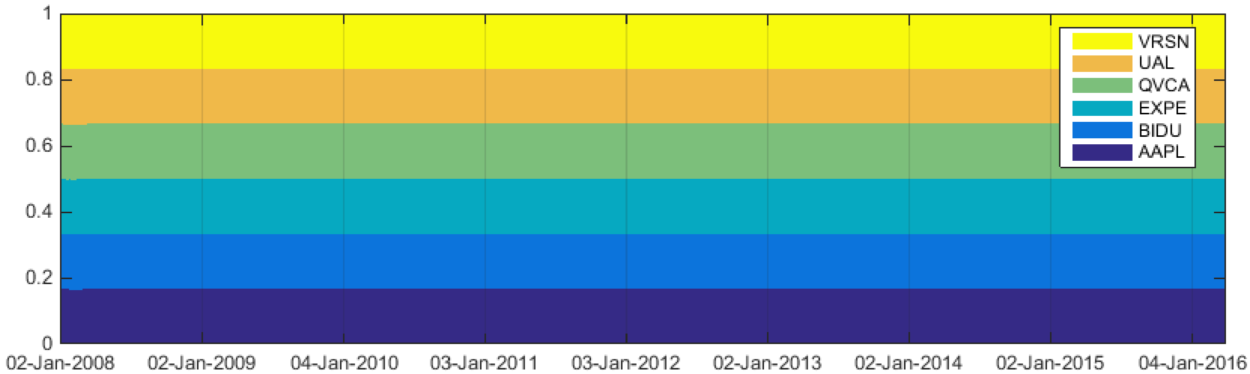
(c) ONS.

Figure 10. Growth wealth over time s_n when the portfolio consists of AAPL, BIDU, EXPE, QVCA, UAL, and VRSN ($s_0 = 10^6$ USD, $c_p = 0$, $c_s = 0.00218\%$).

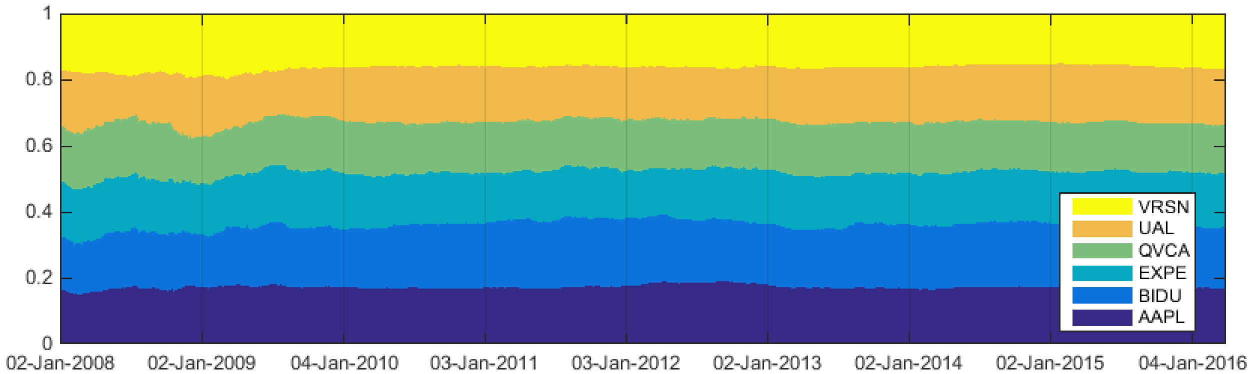
the minimum optimal number of intraday tradings. Consequently, the time complexity of the PM is $O(\tau_{\max})$.



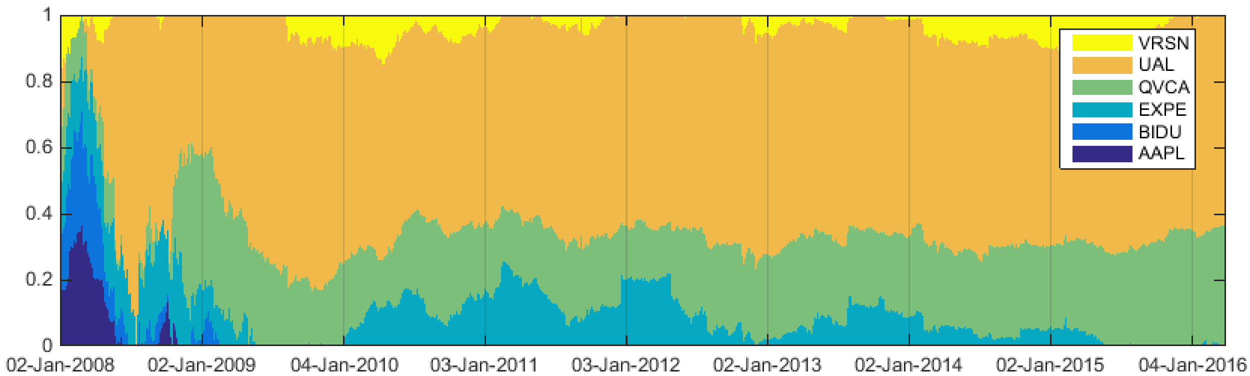
(a) Buy-and-hold (B&H).



(b) T0.

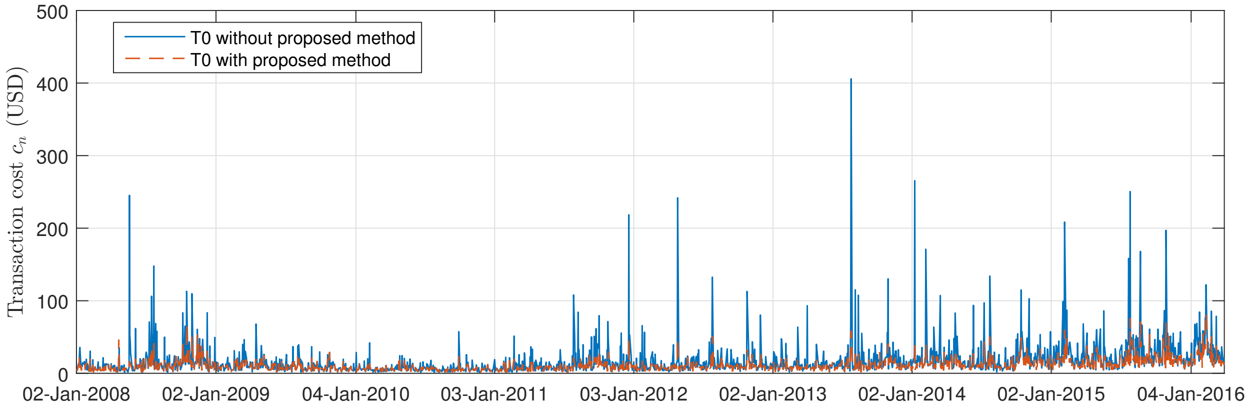


(c) UP.

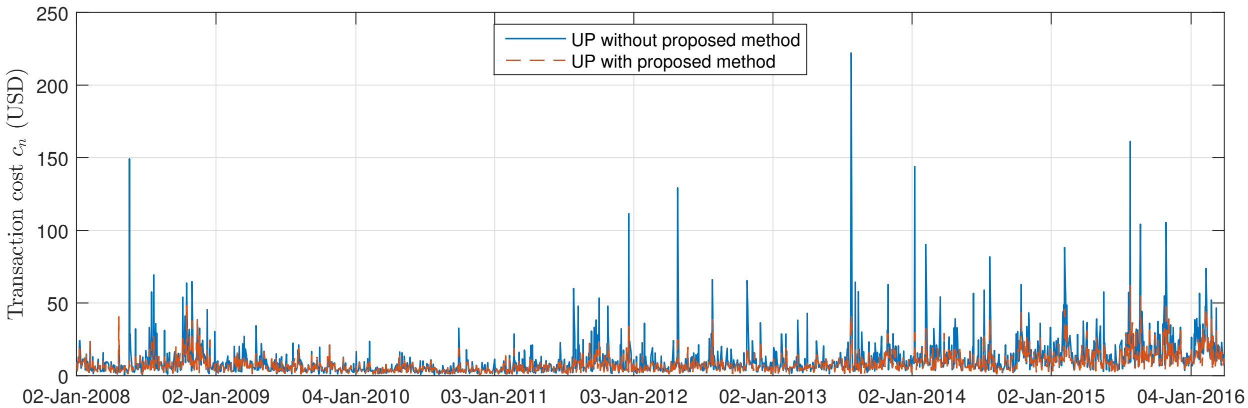


(d) ONS.

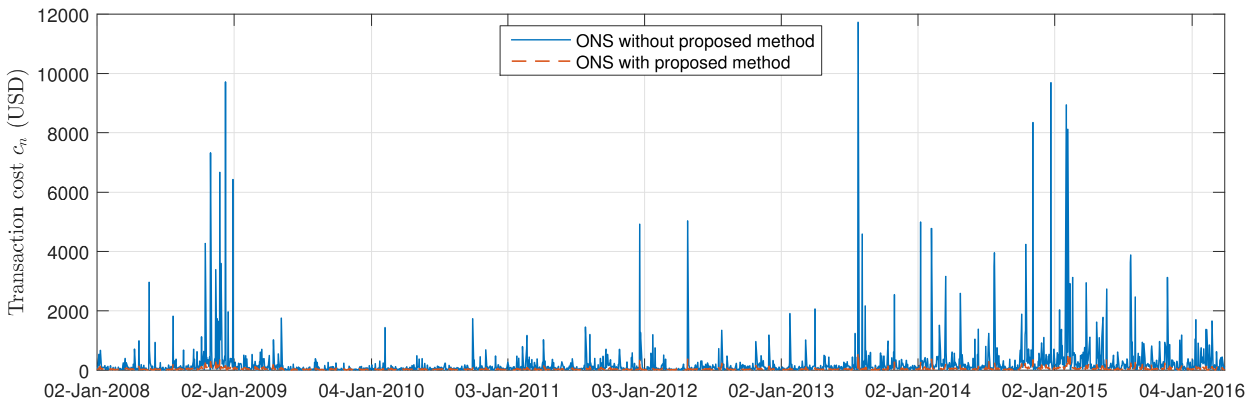
Figure 11. The proportion of portfolio over time $\boldsymbol{b}_n$.



(a) T0.



(b) UP.

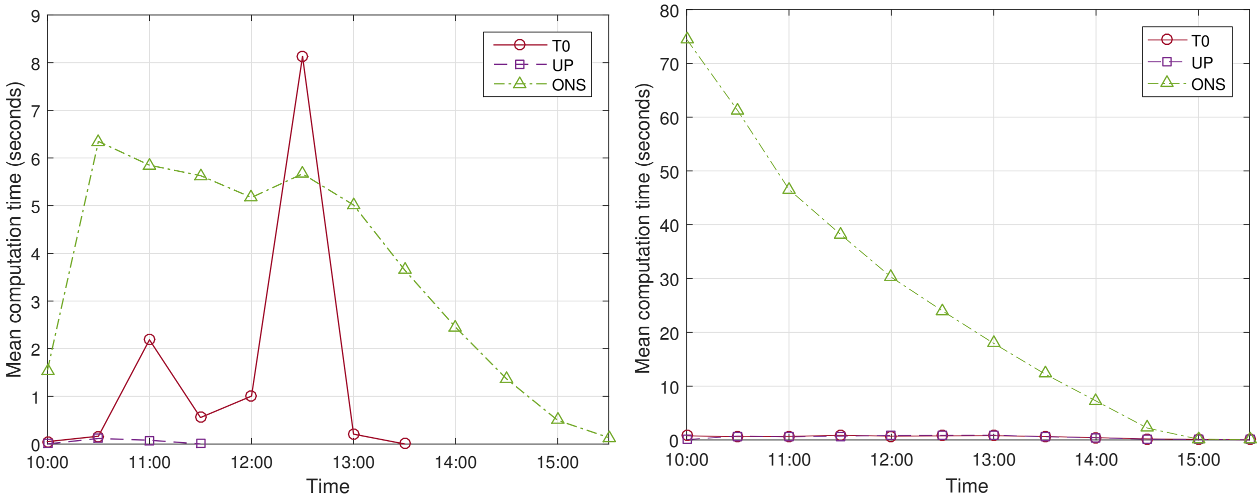


(c) ONS.

Figure 12. Transaction costs, consisting of proportional costs and market impact costs, over time c_n ($s_0 = 10^6$ USD, $c_p = 0$, and $c_s = 0.00218\%$).

7. Conclusion

A mathematical framework to minimise overall TCs, consisting of proportional TCs and MICs, when rebalancing a multi-asset portfolio has been proposed. It considers real-time LOBs and splits very large market orders of the portfolio into small sequential market orders. As a result, it cushions the shock when rebalancing large-sized funds (the backtesting results in Section 6 have demonstrated that the PM is effective for the large capital investment). Moreover, the proposed intraday trading algorithm is applicable to any portfolio rebalancing strategy as well as all the OPS methods regardless of its capital size. However, the heavy computation, analysed in Section 6.6,



(a) s_0 is USD 10^5 (b) s_0 is USD 10^6.

Figure 13. The mean computation time of the proposed method (trading at 16:00 is not counted as intraday trading). The latter parts of T0 and UP are missing as they end intraday trading before 15:30.

should be reduced for real-time algorithmic trading.

References

Agarwal, A., Hazan, E., Kale, S. and Schapire, R.E., Algorithms for portfolio management based on the Newton method. In *Proceedings of the the 23rd International Conference on Machine Learning*, pp. 9--16, 2006.

Agliardi, R. and Gençay, R., Hedging through a limit order book with varying liquidity. *Journal of Derivatives*, 2014, **22**, 32--49.

Alfonsi, A., Fruth, A. and Schied, A., Constrained portfolio liquidation in a limit order book model. In *Proceedings of the Advances in mathematics of finance*, Vol. 83 of *Banach Center publications*, pp. 9--25, 2008.

Alfonsi, A., Fruth, A. and Schied, A., Optimal execution strategies in limit order books with general shape functions. *Quantitative Finance*, 2010, **10**, 143--157.

Almgren, R. and Chriss, N., Optimal execution of portfolio transactions. *Journal of Risk*, 2001, **3**, 5--40.

Avellaneda, M. and Lee, J.H., Statistical arbitrage in the US equities market. *Quantitative Finance*, 2010, **10**, 761--782.

Bendtsen, M. and Peña, J.M., Gated Bayesian networks for algorithmic trading. *International Journal of Approximate Reasoning*, 2016, **69**, 58--80.

Birge, J.R. and Louveaux, F., *Introduction to stochastic programming* (2nd edn), 2011, Springer.

Borodin, A., El-Yaniv, R. and Gogan, V., On the competitive theory and practice of portfolio selection. In *LATIN 2000: Theoretical Informatics*, Vol. 1776 of *Lecture Notes in Computer Science*, pp. 173--196, 2000, Springer.

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

Bussieck, M.R. and Vigerske, S., MINLP solver software. In *Wiley Encyclopedia of Operations Research and Management Science*, 2011, Wiley.

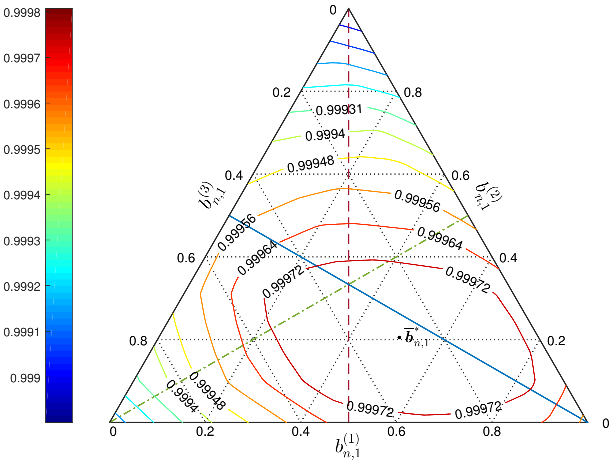
Caldeira, J. and Moura, G.V., Selection of a portfolio of pairs based on cointegration: A statistical arbitrage strategy. *Brazilian Review of Finance*, 2013, **11**, 49--80.

Cont, R., Stoikov, S. and Talreja, R., A stochastic model for order book dynamics. *Operations Research*, 2010, **58**, 549--563.

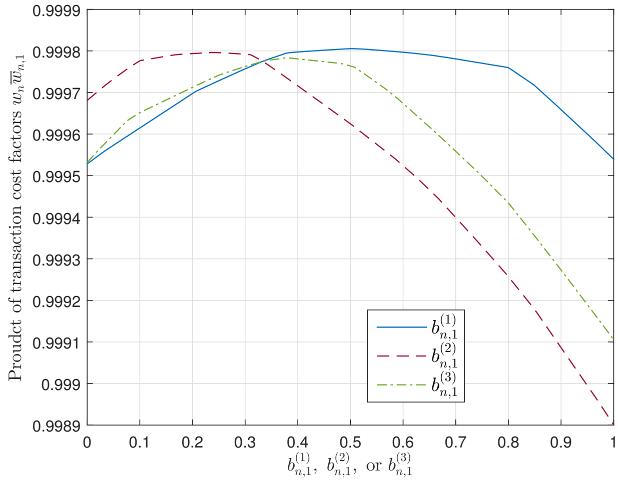
Cover, T.M., Universal portfolios. *Mathematical finance*, 1991, **1**, 1--29.

Crammer, K., Dredze, M. and Pereira, F., Exact convex confidence-weighted learning. In *Proceedings of the Advances in Neural Information Processing Systems*, 2008.

Damodaran, A., *Investment philosophies: Successful strategies and the investors who made them work* (2nd edn), 2012, Wiley.
Das, P., Online convex optimization and its application to online portfolio selection. PhD thesis, University of Minnesota, 2014.
Engle, R.F. and Sokalska, M.E., Forecasting intraday volatility in the us equity market. multiplicative component garch. *Journal of Financial Econometrics*, 2012, **10**, 54--83.
Guéant, O., Lehalle, C.A. and Fernandez-Tapia, J., Optimal portfolio liquidation with limit orders. *SIAM Journal on Financial Mathematics*, 2012, **3**, 740--764.
Györfi, L., Lugosi, G. and Udina, F., Nonparametric kernel-based sequential investment strategies. *Mathematical Finance*, 2006, **16**, 337--357.
Györfi, L., Udina, F. and Walk, H., Nonparametric nearest neighbor based empirical portfolio selection strategies. *Statistics & Decisions*, 2008, **26**, 145--157.
Györfi, L. and Vajda, I., Growth optimal investment with transaction costs. In *Algorithmic Learning Theory*, pp. 108--122, 2008, Springer.
Ha, Y., Review of online portfolio selection: Performance comparison with transaction costs including market impact costs. Working paper, available at http://ssrn.com/abstract=2763202, 2017.
Helmbold, D.P., Schapire, R.E., Singer, Y. and Warmuth, M.K., On-line portfolio selection using multiplicative updates. *Mathematical Finance*, 1998, **8**, 325--347.
Huang, R. and Polak, T., LOBSTER: Limit order book reconstruction system. Working paper, available at http://ssrn.com/abstract=1977207, 2011.
Kissell, R., *The science of algorithmic trading and portfolio management*, 2013, Academic Press.
Kissell, R., Glantz, M. and Malamut, R., *Optimal trading strategies: Quantitative approaches for managing market impact and trading risk*, 2003, AMACOM.
Kissell, R., Glantz, M. and Malamut, R., A practical framework for estimating transaction costs and developing optimal trading strategies to achieve best execution. *Finance Research Letters*, 2004, **1**, 35--46.
Kozat, S.S. and Singer, A.C., Universal semiconstant rebalanced portfolios. *Mathematical Finance*, 2011, **21**, 293--311.
Krauss, C., Do, X.A. and Huck, N., Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, 2017, **259**, 689--702.
Li, B. and Hoi, S.C., On-line portfolio selection with moving average reversion. In *Proceedings of the the 29th International Conference on Machine Learning*, 2012.
Li, B. and Hoi, S.C., Online portfolio selection: A survey. *ACM Computing Surveys*, 2014, **46**, 35.
Li, B., Hoi, S.C. and Gopalkrishnan, V., Corn: Correlation-driven nonparametric learning approach for portfolio selection. *ACM Transactions on Intelligent Systems and Technology*, 2011, **2**.
Li, B., Hoi, S.C., Zhao, P. and Gopalkrishnan, V., Confidence weighted mean reversion strategy for online portfolio selection. *ACM Transactions on Knowledge Discovery from Data*, 2013, **7**, 4.
Li, B. and Hoi, S.C.H., *Online portfolio selection: Principles and algorithms*, 2015, CRC Press.
Li, B., Zhao, P., Hoi, S.C. and Gopalkrishnan, V., PAMR: Passive aggressive mean reversion strategy for portfolio selection. *Machine Learning*, 2012, **87**, 221--258.
Lorenz, J.M., Optimal trading algorithms: Portfolio transactions, multiperiod portfolio selection, and competitive online search. PhD thesis, ETH Zurich, 2008.
Matheson, T., Taxing financial transactions: Issues and evidence. Working paper 11/54, International Monetary Fund, 2011.
Mousavi, S., Esfahanipour, A. and Zarandi, M.H.F., A novel approach to dynamic portfolio trading system using multitree genetic programming. *Knowledge-Based Systems*, 2014, **66**, 68--81.
Olsson, R., Portfolio management under transaction costs: Model development and Swedish evidence. PhD thesis, Umeå University, 2005.
Simon, D., *Evolutionary optimization algorithms*, 2013, Wiley.
Tan, Z., Quek, C. and Cheng, P.Y., Stock trading with cycles: A financial application of ANFIS and reinforcement learning. *Expert Systems with Applications*, 2011, **38**, 4741--4755.
Valle, C., Meade, N. and Beasley, J., Absolute return portfolios. *Omega*, 2014a, **45**, 20--41.
Valle, C., Meade, N. and Beasley, J., Market neutral portfolios. *Optimization Letters*, 2014b, **8**, 1961--1984.
Valle, C., Meade, N. and Beasley, J., Factor neutral portfolios. *OR Spectrum*, 2015, **37**, 843--867.
Xu, H.C., Chen, W., Xiong, X., Zhang, W., Zhou, W.X. and Stanley, H.E., Limit-order book resiliency after effective market orders: Spread, depth and intensity. *Journal of Statistical Mechanics: Theory and Experiment*, 2017.



(a) Ternary contour plot of $w_n\overline{w}_{n,1}$.



(b) 1D plots of $w_n\overline{w}_{n,1}$ (each line matches each straight line in the left plot).

Figure A1. Ternary contour plots and 1D plots of the product of transaction cost factors $w_n\overline{w}_{n,1}$ with the variable $\boldsymbol{b}_{n,1}$ and the fixed values: $\boldsymbol{b}_n = [1/3\ 1/3\ 1/3]^{\mathrm{T}}$, $\boldsymbol{x}_{n,0} = [0.6\ 0.9\ 1.4]^{\mathrm{T}}$, $\nu_{n-1,\tau} = 10^6$ USD, $\boldsymbol{b}_{n+1} = [0.8\ 0.8\ 0.1]^{\mathrm{T}}$, $c_p = 0$, and $c_s = 0.00218\%$. 10-level limit order book data of AAPL ($b^{(1)}$), AMZN ($b^{(2)}$), and GOOG ($b^{(3)}$) on 21 Jun 2012 at 16:00:00 was used.

Appendix A: An example of unimodality of $w_n\overline{w}_{n,1}$

The product of TCFs $w_n\overline{w}_{n,1}$ in (22) can be rewritten as a function of $\boldsymbol{b}_{n,1}$ as

$$\begin{aligned} w_n\overline{w}_{n,1} =& w(\boldsymbol{b}_n, \boldsymbol{b}_{n,1}, \boldsymbol{x}_{n,0}, \nu_{n-1,\tau})w(\boldsymbol{b}_{n,1}, \boldsymbol{b}_{n+1}, \boldsymbol{1}, s_n w_n) \\ =& w(\boldsymbol{b}_n, \boldsymbol{b}_{n,1}, \boldsymbol{x}_{n,0}, \nu_{n-1,\tau})w(\boldsymbol{b}_{n,1}, \boldsymbol{b}_{n+1}, \boldsymbol{1}, \nu_{n-1,\tau}\langle \boldsymbol{b}_n, \boldsymbol{x}_{n,0}\rangle w_n) \\ =& w(\boldsymbol{b}_n, \boldsymbol{b}_{n,1}, \boldsymbol{x}_{n,0}, \nu_{n-1,\tau})w(\boldsymbol{b}_{n,1}, \boldsymbol{b}_{n+1}, \boldsymbol{1}, \nu_{n-1,\tau}\langle \boldsymbol{b}_n, \boldsymbol{x}_{n,0}\rangle w(\boldsymbol{b}_n, \boldsymbol{b}_{n,1}, \boldsymbol{x}_{n,0}, \nu_{n-1,\tau})), \end{aligned} \tag{A1}$$

and it is plotted as a unimodal function of $\boldsymbol{b}_{n,1}$ (i.e. $w_n\overline{w}_{n,1}$ strictly decreases as $\boldsymbol{b}_{n,1}$ goes away from the maximum point $\overline{\boldsymbol{b}}^*_{n,1}$) as shown in Figure A1. Therefore, $w_n\overline{w}_{n,1}$ is a unimodal function of $\boldsymbol{b}_{n,1} \in \Delta^{d-1}$. However, this is only one example from the given values ($\boldsymbol{b}_n$, $\boldsymbol{x}_{n,0}$, $\nu_{n-1,\tau}$, $\boldsymbol{b}_{n+1}$, c_p, and c_s) and the LOBs of the three stocks. The mathematical proof of the unimodality of $w_n\overline{w}_{n,1}$ is not provided in this chapter.

Conclusion

This thesis has proposed three different algorithms of machine learning in quantitative finance. Although they have different topics (multi-output nonlinear regression in Chapter 1, transaction cost factor and online portfolio selection in Chapter 2, and intraday algorithmic trading in Chapter 3), the common contributions of Chapter 2 and Chapter 3 are to propose how to avoid trading illiquid stocks for investors' wealth maximisation.

In Chapter 1, the computation of multi-output relevance vector regression has been accelerated by the proposed method that uses matrix normal distribution to model correlated outputs (the existing method uses multivariate normal distribution). However, the posposed method has the disadvantage: its regression accuracy is less than the existing method.

In Chapter 2, the method how to quantify stock's liquidity from limit order book data when rebalancing a portfolio has been proposed. Also, it was applied to one of online portfolio selection (OPS) methods to reduce market impact costs (MICs), and it showed better performance in the backtesting. Moreover, the proposed liquidity measure can be applied to not only OPS but also mean-variance portfolio selection (MVPS). This is because it can gauge the reduced expected return of MVPS by the limited liquidity when MVPS rebalances a portfolio.

In Chapter 3, the intraday trading algorithm to minimise MICs as well as proportional transaction costs has been proposed by using the liquidity quantification method proposed in Chapter 2. The backtesting results in Chapter 3 have shown that the proposed method reduces MICs for OPS when rebalancing a portfolio. The proposed algorithm has pros and cons: it is applicable to not only OPS but also MVPS, but its inefficient computation should be reformed in the future for real-time trading.

OPS algorithms, dealt with in both Chapter 2 and Chapter 3, which directly optimises a portfolio in terms of long-term investment can be explained econometrically rather than algorithmically. They are classified as i) follow-the-winner (increasing the relative weights of more successful assets in the past periods), ii) follow-the-loser (decreasing the relative weights of more successful assets in the past periods), and iii) pattern matching (finding similar stock return time series in the past and predicting stock returns in the future) by Li and Hoi (2014). The assumption of follow-the-winner and follow-the-loser is that stock returns are modelled as autoregressive time series:

$$r_n = \alpha + \sum_{i=1}^{h} \beta_i r_{n-i} + \epsilon_n, \tag{1}$$

where r_n is excess log return (log return minus risk-free rate) of the n-th period, α is the intercept, h is the maximum lag, and ϵ_n is error term of the n-th period. The follow-the-winner OPS algorithms assume that the coefficient β_i is positive, while the follow-the-lower OPS algorithms assume that it is negative. In a similar way, the assumption of these two OPS categories can be modelled as time series momentum, proposed by Moskowitz *et al.* (2012):

$$\frac{r_n}{\sigma_{n-1}} = \alpha + \beta_h \frac{r_{n-h}}{\sigma_{n-h-1}} + \epsilon_n, \tag{2}$$

or returns signal momentum, proposed by Papailias *et al.* (2017):

$$\frac{r_n}{\sigma_{n-1}} = \alpha + \frac{\beta_h}{h} \sum_{i=1}^{h} \mathbb{1}_{\{r_{n-i} \geq 0\}} + \epsilon_n, \tag{3}$$

where σ_{n-1} is ex-ante volatility, and $\mathbb{1}$ is the indicator function. Both the momentum methods assume that returns scaled by their ex ante volatility depend on previous returns with proper coefficient β_h (follow-the-winner corresponds to positive β_h, whereas follow-the-loser corresponds to negative β_h).

References

Li, B. and Hoi, S.C., Online portfolio selection: A survey. *ACM Computing Surveys*, 2014, **46**, 35.

Moskowitz, T.J., Ooi, Y.H. and Pedersen, L.H., Time series momentum. *Journal of Financial Economics*, 2012, **104**, 228--250.

Papailias, F., Liu, J. and Thomakos, D.D., Returns signal momentum. Working paper, available at http://ssrn.com/abstract=2971444, 2017.

Acknowledgements

The author received Adam Smith Business School PhD Studentship (GBP 44,018 and tuition exemption), and the financial support of EUR 4,798 (excluding VAT) from the Department of Economics at the University of Glasgow to purchase the Limit Order Book System: The Efficient Reconstructor (LOBSTER) data is acknowledged. In addition, the High Performance Compute Cluster of IT Services at the University of Glasgow reduced the total time required to obtain the Monte Carlo simulation results in Chapter 2 and Chapter 3.