A Quantum Online Portfolio Optimization Algorithm

Debbie Lim[*], Patrick Rebentrost[†]

Abstract

Portfolio optimization plays a central role in finance to obtain optimal portfolio allocations that aim to achieve certain investment goals. Over the years, many works have investigated different variants of portfolio optimization. Portfolio optimization also provides a rich area to study the application of quantum computers to obtain advantages over classical computers. In this work, we give a sampling version of an existing classical online portfolio optimization algorithm by Helmbold *et al.*, for which we in turn develop a quantum version. The quantum advantage is achieved by using techniques such as quantum state preparation and inner product estimation. Our quantum algorithm provides a quadratic speedup in the time complexity, in terms of n, where n is the number of assets in the portfolio. The transaction cost of both of our classical and quantum algorithms is independent of n which is especially useful for practical applications with a large number of assets.

1 Introduction

1.1 Online optimization

Online optimization is a branch of optimization, where the input data is revealed over time and decisions have to made while having incomplete knowledge about the input data. At every time step, a loss function will be given based on the decisions made so far. A feature of online optimization is that the sequential input can be given in an adversarial manner; the provable guarantees hold even if the input is chosen by an adversary who knows our strategy. Online convex optimization studies the problem of optimizing a convex function over a convex set in an online fashion. The popular first-order algorithms for online convex optimization include variants of gradient descent, mirror descent and coordinate descent [1, 2, 3, 4, 5].

Apart from the commonly known gradient descent method, the multiplicative weight update method is another alternative to solving optimization problems. The multiplicative weight update method is a primal-dual algorithm proposed by Arora and Kale [6], which assigns an initial weight to each expert and at every iteration, updates the weights according to the experts' performances. This algorithm can also be extended to the online convex optimization framework when the convex set is the n-dimensional simplex. The multiplicative weight update method is one of the second-order methods in online convex optimization besides the Newton's method [7], which iteratively finds the roots of a differentiable function. Some applications of the multiplicative weight update method include solving linear programs and semidefinite programs [8], learning algorithms [9], and portfolio selection [10].

In zeroth-order online convex optimization (bandit convex optimization), the feedback is in the form of a real number (instead of a loss function), thereby being less informative. The first algorithm for bandit convex optimization was proposed by Flaxman *et al.* [11]. Subsequently, many follow up works [12, 13, 14] have been done to improve the regret bound.

[*] Centre for Quantum Technologies, limhueychih@gmail.com

[†] Centre for Quantum Technologies, cqtfpr@nus.edu.sg

1.2 Portfolio optimization

Portfolio optimization is a standard problem in mathematical finance. The first formalization, the Markowitz (mean-variance) model, is proposed by Nobel prize winner, Harry Markowitz [15]. It is a single-period unconstrained quadratic programming problem, which either maximizes the portfolio return for a given level of risk or minimizes the risk for a given return. However, there are several caveats concerning the implementation of this model. Among them, the model relies on the knowledge of the mean and covariance matrix of the asset returns. Besides that, the model suffers from error maximization, i.e., a small change in the inputs can result in a large change in the portfolio [16]. Consequently, many refinements have been proposed to make the model more realistic [17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28].

Reference [10] by Helmbold *et al.* is a seminal paper discussing a (classical) online algorithm for portfolio selection based on the multiplicative weight update rule. The update rule was derived using a framework introduced by Reference [29] for online regression. The authors adapted this framework to the online portfolio selection setting and the resulting algorithm uses linear (in the number of assets) time and space to update the portfolio vector at each time step. A survey on (classical) online portfolio selection was done by Reference [30] from an online machine learning perspective. The survey paper expressed online portfolio selection as a sequential decision problem and included various classes of related algorithms, such as follow the winner, follow the loser, pattern-matching-base approaches and meta-learning algorithms.

1.3 Our work

Our main contribution is an online quantum algorithm for portfolio selection. We show that the online portfolio selection algorithm proposed by Helmbold *et al.* [10] can be quantized. We adopt a step by step approach to demonstrate how we arrive at the quantum algorithm. We start from Algorithm 1, the slightly extended version of the classical online portfolio optimization algorithm from Reference [10] which includes a transaction cost (Corollary 1). Next, we implement a sampling procedure in Algorithm 2 which renders the transaction cost independent of n (Theorem 2). Subsequently, we build on Algorithm 2 but use an inner product estimation procedure to compute the portfolio vectors in Algorithm 3 (Corollary 2). Lastly, we use quantum inner product estimation to replace its classical counterpart and use quantum state preparation to prepare the portfolio vector when devising our quantum online portfolio optimization algorithm, Algorithm 4 (Theorem 4).

We summarize our results in the table below:

Name	Alg.	Regret	Run time	Transaction cost
Online	1	$\frac{1}{r_{\min}}\sqrt{\frac{\log n}{2T}}$	$O(Tn)$	$O(TnC)$
Sampling-based Online	2	$\frac{2}{r_{\min}}\sqrt{\frac{\log \frac{n}{\delta}}{2T}}$	$O(Tn)$	$O(TC)$
Approximate Sampling-based Online	3	$\frac{6}{r_{\min}}\sqrt{\frac{\log \frac{n}{\delta}}{2T}}$	$O\left(Tn + \frac{T^2}{r_{\min}}\log\left(\frac{1}{\delta}\right)\right)$	$O(TC)$
Quantum Online	4	$\frac{20}{r_{\min}}\sqrt{\frac{\log \frac{n}{\delta}}{2T}}$	$\tilde{O}\left(\frac{T^3\sqrt{n}}{r_{\min}}\log\left(\frac{1}{\delta}\right)\right)$	$O(TC)$

Table 1: Summary of results. Throughout this work, n is the number of assets, T is the total number of time steps, and $r_{\min}$ is the lower bound for the price relatives (see Assumption 3). In addition, C is the transaction cost (see Assumption 2) and 3δ is an upper bound on the probability of failure.

The regret bound achieved by Algorithm 4 is larger than than that of Algorithm 1 only by a small factor, and the algorithm provides a quadratic speedup in the run time in terms of n, the number of assets in the portfolio. The speedup is due to quantum state preparation and inner product estimation, which in turn use amplitude amplification. In addition, the algorithm does not have to store the portfolio vectors $w^{(t)}$ explicitly for every time step t. Instead, the portfolio vectors can be computed efficiently via unitaries that perform arithmetic operations. Moreover, the transaction cost of our algorithm is independent of n, which is especially useful for practical applications with a large number of assets in the portfolio.

1.4 Related work

References [31, 32, 33, 34] discuss the state-of-the-art, potential and challenges of quantum computing in finance. Rosenberg *et al.* [35] discuss a non-convex discrete portfolio optimization problem, in the context of D-Wave's quantum annealer. Their problem formulation aims to maximize the expected return while minimizing the risk and transaction costs. They numerically showed that the quantum annealer in principle could solve this problem with high probability and this success probability can be increased by making adjustments to the annealer. In Ref. [36], the authors proposed a quantum algorithm for the unconstrained portfolio optimization problem. The algorithm uses quantum linear system solvers [37, 38] to obtain speedups for portfolio optimization problems that can be reduced to unconstrained quadratic programs, which in turn are reducible to a single linear system. Subsequently, Ref. [39] gave a quantum algorithm for the general constrained portfolio optimization problem with an arbitrary number of nonnegativity and budget constraints, resulting in a polynomial speedup in terms of the number of assets, as compared to the best known classical algorithm when only a moderately accurate solution is required.

In terms of practical implementation, Reference [40] evaluated the experimental performance of using the Quantum Approximate Optimization Algorithm and the Quantum Alternating Operator Ansatz to solve a discrete portfolio optimization problem for a multi-period portfolio rebalancing setting. Subsequently, Ref. [41] numerically showed that the Quantum Walk Optimization Algorithm is capable of achieving a significantly better performance. In the noisy intermediate-scale quantum (NISQ) setting, the work by Ref. [42] proposed a hybrid algorithm for end-to-end execution of small scale portfolio optimization problems on near-term devices. Their algorithm uses techniques such as mid-circuit measurement, quantum conditional logic, and qubit reset/reuse, and also improved on the existing eigenvalue inversion component of HHL.

Online optimization has been considered in the quantum setting. Boosting is an approach to improve the performance of a weak learning algorithm in terms of its accuracy. Quantum boosting was discussed in Reference [43] to improve the time complexity of the widely used classical AdaBoost proposed by Ref. [44]. Subsequently, a follow-up work by Ref. [45] was done to provide a significantly faster and simpler quantum boosting algorithm. The Hedge algorithm proposed by Freund and Schapire uses the multiplicative weight update method to adaptively allocate mixed strategies to solve an adversarial online optimization problem. The Sparsitron by Reference [46] which is based on the Hedge algorithm, is a machine learning algorithm for undirected graphical models. Quantum versions of both the Hedge algorithm and the Sparsitron were discussed in Ref. [47]. In zeroth-order optimization, there are instances where quantum advantage have been proven. For example, in the multi-armed bandits setting, Ref.[48] proposed a quantum algorithm that provides an exponential speedup in terms of the time T in the regret bound as compared to the well known classical lower bounds [49, 50]. In bandit convex optimization, Ref. [51] gave an quantum algorithm that achieves a regret bound that is independent of n, the dimension. This outperforms the best known optimal classical algorithm [52].

2 Preliminaries

2.1 Notations

We use $[n]$ to represent the set $\{1, \cdots, n\}$, where $n \in \mathbb{Z}_+$ and denote the i-th entry of a vector $v \in \mathbb{R}^n$ as v_i for $i \in [n]$. If a vector has a time dependency we denote it as $v^{(t)}$. Let $\mathbf{e}_i$ be the vector of all zeros with a 1 in the i-th position. The ℓ_1-norm of a vector $v \in \mathbb{R}^n$ is defined as $\|v\|_1 := \sum_{i=1}^n |v_i|$. We use $\bar{0}$ to denote the all zeros vector and use $|\bar{0}\rangle$ to denote the state $|0\rangle \otimes \cdots \otimes |0\rangle$, where the number of qubits is clear from the context. The maximum entry in absolute value of a vector $v \in \mathbb{R}^n$ is denoted as $\|v\|_{\max} = \max\limits_{i \in [n]} |v_i|$ and we denote the maximum entry of a vector $v \in \mathbb{R}^n$ as $v_{\max} = \max\limits_{i \in [n]} v_i$. For $v \in \mathbb{R}^n$ and $k \in \mathbb{R}$, $k^v \in \mathbb{R}^n$ is the element-wise exponential v, i.e. $(k^v)_i = k^{v_j}$. We write the natural logarithm (base e) as log. We use $\tilde{O}(\cdot)$ to hide the polylog factor, i.e., $\tilde{O}(f(n)) = O(f(n) \cdot \text{polylog}(f(n)))$. We sometimes use $O(1)$ to denote a constant.

2.2 The computational model

We refer to the run time of a classical/quantum computation as the number of basic gates performed. We assume a classical arithmetic model, which allows us to ignore issues arising from the fixed-point representation of real numbers. The basic arithmetic operations take constant time. In the quantum setting, we assume a quantum circuit model. Each quantum gate in the circuit represents an elementary operation, and the application of each quantum gate takes constant time. The time complexity of a given unitary operator U is the minimum number of basic quantum gates required to prepare U. In addition, we assume a quantum arithmetic model, which is equivalent to the classical model in that arithmetic operations take constant time. Our quantum algorithm assumes quantum query access to certain vectors. For the oracles, the representation of real numbers to finite precision is also not taken into account. Given a vector $v \in \mathbb{R}^n$, we say we have quantum query access to this vector if we have access to the operation O_v which performs

$$O_v |i\rangle |\bar{0}\rangle \quad = \quad |i\rangle |v_i\rangle . \tag{1}$$

The second register is assumed to contain sufficient qubits to make all the subsequent computations accurate, in analogy to the sufficient bits that a classical algorithm assumes to run correctly.

3 The online portfolio optimization framework

Consider T discrete time steps and n assets, and the setting as in Ref. [10]. A portfolio of these n assets at time $t \in [T]$ is described by a vector $w^{(t)}$ such that for each $i \in [n]$, $w_i^{(t)} \geqslant 0$ and $\sum\limits_{i=1}^{n} w_i^{(t)} = 1$. Here, we make the no-shortselling assumption, see also below. Each asset has a price as a function of time and in this work we consider the time series of closing prices. The original paper uses the opening prices, and we assume that the closing price at t is the same as the opening price at $t+1$. Define the day-to-day return as

$$R_i^{(t)} := \frac{\text{closing price of asset } i \text{ on day } t}{\text{closing price of asset } i \text{ on day } t-1}. \tag{2}$$

In this work, the performance of the assets is reflected in a price relative vector $\rho^{(t)} \in \mathbb{R}_+^n$, where for all $i \in [n]$, $\rho_i^{(t)}$ is the ratio

$$\rho_i^{(t)} := \frac{R_i^{(t)}}{R_{\max}^{(t)}}, \tag{3}$$

where $R^{(t)}_{\max} = \max\limits_{j \in [n]} R^{(t)}_j$. We assume that $\rho^{(t)}$ has at least one non-zero entry and by definition $0 \leqslant \rho^{(t)}_i \leqslant 1$ for all $i \in [n]$ and $t \in [T]$. Given w and ρ, an investor's wealth changes by a factor of

$$w \cdot \rho \;\; = \;\; \sum_{i=1}^{n} w_i \cdot \rho_i \tag{4}$$

from one trading day to the next. In the online portfolio selection setting, the learning algorithm has access to the price relative vectors $\rho^{(1)}, \cdots, \rho^{(t)}$ at the end of trading day t. The algorithm then selects the portfolio $w^{(t+1)}$ for the next day. On the end of each trading day t, $\rho^{(t)}$ is revealed and the investor's wealth changes by a factor of $w^{(t)} \cdot \rho^{(t)}$. As time progresses, $\rho^{(1)}, \cdots, \rho^{(T)}$ will be revealed and $w^{(1)}, \cdots, w^{(T)}$ will be selected. From the start of trading day 1 through the start of trading day $(T+1)$, the wealth changes by a factor of

$$S \;\; = \;\; \prod_{t=1}^{T} w^{(t)} \cdot \rho^{(t)}. \tag{5}$$

Similar to the analysis in Reference [10], we will deal with the normalized logarithm of S:

$$LS \;\; = \;\; \frac{1}{T} \sum_{t=1}^{T} \log\left(w^{(t)} \cdot \rho^{(t)}\right), \tag{6}$$

since wealth often grows or decays geometrically in typical markets.

On the other hand, consider the "offline gain", $LS^* = \frac{1}{T} \max\limits_{\{w:\|w\|_1=1\}} \sum\limits_{t=1}^{T} \log\left(w \cdot \rho^{(t)}\right)$, which is the maximum gain in wealth achievable when choosing the same portfolio $w^* = \operatorname*{arg\,max}\limits_{\{w:\|w\|_1=1\}} S(w) = \operatorname*{arg\,max}\limits_{\{w:\|w\|_1=1\}} LS(w)$ for all trading days $t \in [T]$. The difference of offline loss and the loss of some sequence of normalized $w^{(t)}$ is called regret. Formally, it is $LS^* - LS$ and can be naively bounded as $LS^* - LS \leqslant \log\left(\frac{1}{r_{\min}}\right)$. The bound follows from

$$\begin{aligned} LS^* - LS \;\; &= \;\; \max_{\{w:\|w\|_1=1\}} \frac{1}{T} \sum_{t=1}^{T} \log(w \cdot \rho^{(t)}) - \frac{1}{T} \sum_{t=1}^{T} \log(w^{(t)} \cdot \rho^{(t)}) &(7)\\ &\leqslant \;\; \frac{1}{T} \sum_{t=1}^{T} \log\left(\|\rho^{(t)}\|_{\max}\right) - \frac{1}{T} \sum_{t=1}^{T} \log\left(\|w^{(t)}\|_1 r_{\min}\right) &(8)\\ &= \;\; \log\left(\frac{1}{r_{\min}}\right). &(9) \end{aligned}$$

This bound does not decrease with T. A main result of the work by Reference [10] is a sequence of $w^{(t)}$ which shows a regret bound of about $1/\sqrt{T}$.

We would like to emphasize again the following assumptions.

Assumption 1. *We assume that there is no short-selling throughout the trading period. Therefore,* $w^{(t)}_i \geqslant 0$ *for all* $t \in [T]$ *and* $i \in [n]$.

To model the cost of trading, we assume a fixed transaction cost per investment. This cost will highlight the difference between the standard and the sampling algorithm. It was mentioned as a possible extension in Reference [10].

Assumption 2. *We assume that a transaction cost of* $C \geqslant 0$ *is incurred when investing in a single asset. Here, this transaction cost is independent of the amount of asset that is bought.*

The following assumption is as in the classical work and simplifies the analysis of the regret bound. Reference [10] also relaxes this assumption and provides a different regret bound for the relaxed setting.

Assumption 3. *We assume a known lower bound $r_{\min} \in (0,1]$ for the price relatives, i.e., $r_{\min} \leqslant \rho_i^{(t)} \leqslant 1$ for all $i \in [n]$ and $t \in [T]$.*

3.1 Helmbold *et al.*'s algorithm

In Reference [10], the authors provide a online algorithm for portfolio optimization. Given a current portfolio $w^{(t)}$, consider the following optimization problem

$$\max_{\{w^{(t+1)}:\|w^{(t+1)}\|_1=1\}} \eta \left(\log\left(w^{(t)} \cdot \rho^{(t)}\right) + \frac{\rho^{(t)} \cdot \left(w^{(t+1)} - w^{(t)}\right)}{w^{(t)} \cdot \rho^{(t)}} \right) - d\left(w^{(t+1)}, w^{(t)}\right). \tag{10}$$

The problem is to pick a portfolio vector $w^{(t+1)}$ that maximizes the gain and at the same time, is close to the portfolio vector picked in the previous iteration. Here, η is the "learning rate" and $d\left(w^{(t+1)}, w^{(t)}\right)$ is a distance measure between $w^{(t+1)}$ and $w^{(t)}$. We formally define the update rule which is the solution to Eq. (10) when the relative entropy is used as the distance measure.

Definition 1 (Exponentiated gradient $EG(\eta)$ update [10])**.** *With $\eta > 0$, $w \in \mathbb{R}^n$ and $\rho \in \mathbb{R}^n$, we define by $EG(\eta, w, \rho) \in \mathbb{R}^n$ the mapping which performs the following weight update for all $i \in [n]$:*

$$(\eta, w, \rho, i) \to \frac{w_i \exp\left(\eta \frac{\rho_i}{w \cdot \rho}\right)}{Z}, \tag{11}$$

where $Z = \sum_{j=1}^{n} w_j \exp\left(\eta \frac{\rho_j}{w \cdot \rho}\right)$. We use the shorthand notation $EG(\eta)$ if the other inputs are clear from the context.

In order to solve the portfolio selection problem, Reference [10] gave Algorithm 1, which uses linear time and space (in n) to update $w^{(t)}$ for each t. We present a slightly extended version of their algorithm by including the transaction cost for investing in an asset.

Algorithm 1 Online Portfolio Optimization Algorithm

```
Input: $n$, $\eta$, $T$, $C$.
Initialize $w^{(1)} = \left(\frac{1}{n}, \cdots, \frac{1}{n}\right) \in \mathbb{R}^n$.
for $t = 1$ to $T$ do
    Invest in all assets according to $w^{(t)}$, with cost $C$ for each asset.
    Wait until end of day.
    Receive price relative vector $\rho^{(t)}$.
    for $i = 1$ to $n$ do
        $w_i^{(t+1)} \leftarrow EG(\eta, w^{(t)}, \rho^{(t)})_i$.
    end for
end for
Output: $LS_{\mathrm{EG}} \leftarrow \frac{1}{T} \sum_{t=1}^{T} \log\left(w^{(t)} \cdot \rho^{(t)}\right)$.
```

The following theorem by Reference [10] bounds the difference in wealth gained when using a fixed portfolio vector versus the update rule Def. (1) applied to portfolio vector initialized to be the uniform vector $\left(\frac{1}{n}, \cdots, \frac{1}{n}\right) \in \mathbb{R}^n$.

Theorem 1 ([10])**.** *Let $u \in \mathbb{R}_+^n$ be a portfolio vector, and let $\rho^{(1)}, \cdots, \rho^{(t)}$ be of price relatives with $\max_{i \in [n]} \rho_i^{(t)} = 1$ and $\rho_i^{(t)} \geqslant r_{\min} > 0$ for all $i \in [n], t \in [T]$, where it is assumed that $r_{\min}$ is known. Set $w^{(1)} = \left(\frac{1}{n}, \cdots, \frac{1}{n}\right)$ and $\eta = 2r_{\min}\sqrt{\frac{2\log n}{T}}$. The update in Def. (1) produces portfolio vectors that achieve the following bound:*

$$\sum_{t=1}^{T} \log\left(w^{(t)} \cdot \rho^{(t)}\right) \quad \geqslant \quad \sum_{t=1}^{T} \log\left(u \cdot \rho^{(t)}\right) - \frac{\sqrt{2T\log n}}{2r_{\min}}. \tag{12}$$

Thm. 1 implies the corollary below.

Corollary 1 (Guarantee and run time of Algorithm 1 [10])**.** *Algorithm 1 with $\eta = 2r_{\min}\sqrt{\frac{2\log n}{T}}$ achieves*

$$LS^* - LS_{\mathrm{EG}} \quad \leqslant \quad \frac{1}{r_{\min}}\sqrt{\frac{\log n}{2T}}. \tag{13}$$

with a total run time of $O(Tn)$ and a transaction cost of $O(TnC)$.

Compared to the naive bound $LS^* - LS_{\mathrm{EG}} \leqslant \log\left(\frac{1}{r_{\min}}\right)$, this bound decreases with T and is better when $T \geqslant \frac{\log n}{2r_{\min}^2 \log^2\left(\frac{1}{r_{\min}}\right)}$.

3.2 Sampling-based online portfolio optimization algorithm

We now consider including a sampling procedure, which leads to a reduction in the total transaction cost as we only invest in the sampled asset.

Fact 1 (ℓ_1-sampling [53, 54])**.** *Given a probability vector $p \in [0,1]^n$, there exists a data structure that samples the index $i \in [n]$ with probability p_i which can be constructed in $O(n)$ time. The time required for obtaining one sample is $O(1)$.*

The assumptions of Ref. [53] allow to omit $\log(n)$ factors in the time for construction and sampling, and we adopt the same assumptions. Based on this data structure, we construct Algorithm 2. This algorithm only samples a single asset from the portfolio vector and invests in that asset. For the portfolio update, however, the complete vector of price relatives is used and the complete new portfolio vector is computed. The following theorem gives a upper bound on the regret of the logarithmic wealth obtained from sampling from the exponential gradient update.

Theorem 2. *Let $\delta \in (0, 1/4)$ and LS_{EG} as in Algorithm 1. Algorithm 2 outputs $LS_{samp} := \frac{1}{T}\sum_{t=1}^{T} \log \rho_{i^{(t)}}^{(t)}$ with bias $\left|\mathbb{E}\left[LS_{\mathrm{samp}}\right] - LS_{\mathrm{EG}}\right| \leqslant 2\log\left(\frac{1}{r_{\min}}\right)\sqrt{\frac{\log\frac{1}{\delta}}{2T}}$ with success probability at least $1 - 4\delta$. With $\eta = 2r_{\min}\sqrt{\frac{2\log n}{T}}$, the regret bound is*

$$LS^* - LS_{\mathrm{samp}} \leqslant \frac{2}{r_{\min}}\sqrt{\frac{\log(\frac{n}{\delta})}{2T}}, \tag{14}$$

with success probability at least $1 - 2\delta$. The total run time is $O(Tn)$ and the transaction cost is $O(TC)$.

Algorithm 2 Sampling-based Online Portfolio Optimization Algorithm

```
Input: $n$, $\eta$, $T$, $C$.
Initialize $w^{(1)} = \left(\frac{1}{n}, \cdots, \frac{1}{n}\right) \in \mathbb{R}^n$.
for $t = 1$ to $T$ do
    Prepare sampling data structure for $w^{(t)}$ using Fact 1.
    $i^{(t)} \leftarrow$ Sample from $w^{(t)}$.
    Invest in asset $i^{(t)}$ at cost $C$.
    Wait until end of day.
    Receive price relative vector $\rho^{(t)}$.
    for $i = 1$ to $n$ do
        $w_i^{(t+1)} \leftarrow EG(\eta, w^{(t)}, \rho^{(t)})_i$.
    end for
end for
Output: $LS_{\text{samp}} := \frac{1}{T}\sum_{t=1}^{T} \log \rho_{i^{(t)}}^{(t)}$.
```

Proof. The expected gain is bounded by

$$\begin{aligned} \mathbb{E}_w\left[LS_{\text{samp}}\right] &= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_w\left[\log\left(\rho_{i^{(t)}}^{(t)}\right)\right] && (15)\\ \text{(Jensen's)} &\leqslant \frac{1}{T}\sum_{t=1}^{T} \log\left(\mathbb{E}_w\left[\rho_{i^{(t)}}^{(t)}\right]\right) && (16)\\ &= \frac{1}{T}\sum_{t=1}^{T} \log\left(w^{(t)} \cdot \rho^{(t)}\right) \equiv LS_{\text{EG}}. && (17) \end{aligned}$$

Since $\log r_{\min} \leqslant \log \rho_i^{(t)} \leqslant 0$, then by Eq. (17), Lemma 5, and using Hoeffding's inequality (see Fact. 3) with $q > 0$, we obtain

$$\begin{aligned} \mathbb{P}\left[\left|LS_{\text{samp}} - LS_{\text{EG}}\right| \geqslant q\right] &\leqslant \mathbb{P}\left[\left|LS_{\text{samp}} - \mathbb{E}_w[LS_{\text{samp}}]\right| \geqslant q\right] && (18)\\ &= \mathbb{P}\left[\left|T \cdot LS_{\text{samp}} - T \cdot \mathbb{E}_w\left[LS_{\text{samp}}\right]\right| \geqslant T \cdot q\right] && (19)\\ \text{(Hoeffding's)} &\leqslant 2e^{-\frac{2q^2T}{\log^2 r_{\min}}}. && (20) \end{aligned}$$

Setting $q = \log\left(\frac{1}{r_{\min}}\right)\sqrt{\frac{\log\frac{1}{\delta}}{2T}}$, we obtain

$$\mathbb{P}\left[\left|LS_{\text{samp}} - LS_{\text{EG}}\right| \geqslant \log\left(\frac{1}{r_{\min}}\right)\sqrt{\frac{\log\frac{1}{\delta}}{2T}}\right] \leqslant 2e^{\frac{\log^2\left(\frac{1}{r_{\min}}\right)\log\delta}{\log^2 r_{\min}}} \leqslant 2e^{\log\delta} = 2\delta. \qquad (21)$$

For the success probability, we hence obtain

$$\mathbb{P}\left[\left|LS_{\text{samp}} - LS_{\text{EG}}\right| \leqslant \log\left(\frac{1}{r_{\min}}\right)\sqrt{\frac{\log\frac{1}{\delta}}{2T}}\right] \geqslant 1 - 2\delta. \qquad (22)$$

For the bias, note that $\left|\mathbb{E}\left[LS_{\text{samp}}\right] - LS_{\text{EG}}\right| \leqslant \left|\mathbb{E}\left[LS_{\text{samp}}\right] - LS_{\text{samp}}\right| + \left|LS_{\text{samp}} - LS_{\text{EG}}\right|$, which we can bound by $2q$ with success probability at least $1 - 4\delta$, using Eq. (18). Combining Thm. 1

and Eq. (22), the regret is bounded by

$$\begin{aligned} LS_{\text{samp}} - LS^* &= L_{\text{samp}} - LS_{\text{EG}} + LS_{\text{EG}} - LS^* && (23)\\ &\leqslant \log\left(\frac{1}{r_{\min}}\right)\sqrt{\frac{\log\frac{1}{\delta}}{2T}} + \frac{1}{r_{\min}}\sqrt{\frac{\log n}{2T}} && (24)\\ &\leqslant \frac{2}{r_{\min}}\sqrt{\frac{\log\frac{n}{\delta}}{2T}}, && (25)\end{aligned}$$

which holds with probability at least $1-2\delta$. □

The regret bound worsens slightly in two regards. Firstly, we obtain a constant factor to the regret bound. Secondly, the algorithm is probabilistic and we obtain a $\log\frac{1}{\delta}$ dependence in the regret bound, where 2δ is the failure probability of the algorithm. Usually, this failure probability can be taken as some small constant such as 0.001 to obtain a 99.9% confidence that the algorithm ran correctly. Moreover, economic reasoning suggests that it is important to diversify the portfolio at each step, or, in other words, obtain multiple samples per step. Hence, while the regret bound only slightly worsens there may be other benefits to investing in multiple assets. However, we will not discuss this multi-sampling, as the purpose of introducing a sampling procedure in the algorithm is to show that the transaction cost can be significantly reduced at a small cost in the regret bound.

3.3 Convergence theorem for erroneous updates

Before we move to an approximate classical algorithm and our quantum algorithm, we generalize the convergence result from the original work. The generalizations are in terms of the availability of the inner product and the normalization factor, both of which will be known only approximately in the quantum algorithm. The generalization is embodied in the following definition of an erroneous update rule.

Definition 2 (Erroneous exponentiated gradient update)**.** *Let $\eta > 0$, $w \in \mathbb{R}^n$, and $\rho \in \mathbb{R}^n$. In addition, let $\epsilon_I \in (0, 1/2)$ and $\tilde{I} > 0$ such that $|\tilde{I} - I| \leqslant I\epsilon_I$, where $I := w \cdot \rho$. Moreover, let $\epsilon_Z \in (0, 1/2)$ and $\tilde{Z} > 0$ such that $|\tilde{Z} - Z| \leqslant Z\epsilon_Z$, where $Z = \sum_{j=1}^{n} w_j \exp\left(\eta\frac{\rho_j}{\tilde{I}}\right)$. We define the erroneous weight update $EEG(\eta, w, \rho, \tilde{I}, \tilde{Z}) \in \mathbb{R}^n$ as the mapping which computes for $i \in [n]$*

$$(\eta, w, \rho, \tilde{I}, \tilde{Z}, i) \to \frac{w_i \exp\left(\eta\frac{\rho_i}{\tilde{I}}\right)}{\tilde{Z}}. \tag{26}$$

We use the shorthand notation $EEG(\eta)$ if the other inputs are clear from the context.

The main theorem for this update rule is as follows, for which we modify the proof of Theorem 1 from Ref. [10].

Theorem 3 (Main convergence theorem for erroneous updates)**.** *Let $u \in \mathbb{R}^n_+$ be a portfolio vector, and let $\rho^{(1)}, \cdots, \rho^{(t)}$ be price relatives with $\max_{i\in[n]} \rho_i^{(t)} = 1$ and $\rho_i^{(t)} \geqslant r_{\min} > 0$ for all $i \in [n], t \in [T]$. With $w^{(1)} = \left(\frac{1}{n}, \cdots, \frac{1}{n}\right)$ and $\eta = 2r_{\min}\sqrt{\frac{2\log n}{T}}$, the update in Def. (2) produces portfolio vectors that achieve the bound*

$$\sum_{t=1}^{T} \log\left(w^{(t)} \cdot \rho^{(t)}\right) \geqslant \sum_{t=1}^{T} \log\left(u \cdot \rho^{(t)}\right) - \frac{3}{r_{\min}}\sqrt{\frac{2\log n}{T}}, \tag{27}$$

when $\epsilon_I = \frac{3\eta}{4r_{\min}}$ *and* $\epsilon_Z = 0$*, and*

$$\sum_{t=1}^{T} \log\left(w^{(t)} \cdot \rho^{(t)}\right) \geqslant \sum_{t=1}^{T} \log\left(u \cdot \rho^{(t)}\right) - \frac{9}{r_{\min}}\sqrt{\frac{2\log n}{T}}, \tag{28}$$

when $\epsilon_I = \frac{3\eta}{4r_{\min}}$ *and* $\epsilon_Z = \frac{\eta^2}{r_{\min}}$.

Proof. Let $D\left(u\|v\right) := \sum_{i=1}^{n} u_i \log\left(\frac{u_i}{v_i}\right)$. Fix $w^{(t)}$ and let $w^{(t+1)} = EEG(\eta, w^{(t)}, \rho^{(t)}, \tilde{I}^{(t)}, \tilde{Z}^{(t)})$, with $\tilde{I}^{(t)}, \tilde{Z}^{(t)}$ as in Def. (2). Let $\Delta_t = D(u\|w^{(t+1)}) - D(u\|w^{(t)})$. Then

$$\Delta_t = -\sum_{i=1}^{n} u_i \log\left(\frac{w_i^{(t+1)}}{w_i^{(t)}}\right) \tag{29}$$

$$= -\sum_{i=1}^{n} u_i \left(\eta\left(\frac{\rho_i^{(t)}}{\tilde{I}^{(t)}}\right) - \log\tilde{Z}^{(t)}\right) \tag{30}$$

$$= -\eta\frac{u\cdot\rho^{(t)}}{\tilde{I}^{(t)}} + \log\tilde{Z}^{(t)} \tag{31}$$

$$\leqslant -\eta\left(1 - \frac{4}{3}\epsilon_I\right)\frac{u\cdot\rho^{(t)}}{I^{(t)}} + \log\tilde{Z}^{(t)}. \tag{32}$$

Since $\rho_i^{(t)} \in [0,1]$ and $k^y \leqslant 1 - 1(1-k)y$ for $k > 0$ and $y \in [0,1]$, we have

$$\tilde{Z}^{(t)} \leqslant (1+\epsilon_Z)Z^{(t)} = (1+\epsilon_Z)\sum_{i=1}^{n} w_i^{(t)} e^{\eta\frac{\rho_i^{(t)}}{\tilde{I}^{(t)}}} \leqslant (1+\epsilon_Z)\sum_{i=1}^{n} w_i^{(t)}\left(1 - \left(1 - e^{\frac{\eta}{\tilde{I}}}\right)\rho_i^{(t)}\right) \tag{33}$$

$$= (1+\epsilon_Z)\left(1 - \left(1 - e^{\frac{\eta}{\tilde{I}^{(t)}}}\right)\tilde{I}^{(t)}\right). \tag{34}$$

Using the fact that $\log\left(1 - c\left(1 - e^y\right)\right) \leqslant cy + \frac{y^2}{8}$ for all $c \in [0,1]$ and $y \in \mathbb{R}$, $\log\tilde{Z}_t$ can be bounded by

$$\log\tilde{Z}^{(t)} = \log\left(1+\epsilon_Z\right) + \log\left(1 - \left(1 - e^{\frac{\eta}{\tilde{I}^{(t)}}}\right)\tilde{I}^{(t)}\right) \leqslant \epsilon_Z + \eta + \frac{\eta^2}{8(\tilde{I}^{(t)})^2} \tag{35}$$

$$\leqslant \epsilon_Z + \eta + \frac{\eta^2}{8(I^{(t)})^2}\left(1 + \frac{4}{3}\epsilon_I\right)^2 \leqslant \epsilon_Z + \eta + \frac{2\eta^2}{9(I^{(t)})^2} \leqslant \epsilon_Z + \eta + \frac{2\eta^2}{9r_{\min}^2}, \tag{36}$$

which is true as $\log(1+\epsilon_Z) \leqslant \epsilon_Z$ since $\epsilon_Z \geqslant 0$ and using $\rho_i^{(t)} \geqslant r_{min}$. Combining with Eq. (29), we have

$$\Delta_t \leqslant -\eta\left(1 - \frac{4}{3}\epsilon_I\right)\frac{u\cdot\rho^{(t)}}{I^{(t)}} + \epsilon_Z + \eta + \frac{2\eta^2}{9r_{\min}^2} \tag{37}$$

$$= \eta\left(1 - \left(1 - \frac{4}{3}\epsilon_I\right)\frac{u\cdot\rho^{(t)}}{I^{(t)}}\right) + \epsilon_Z + \frac{2\eta^2}{9r_{\min}^2} \tag{38}$$

$$\leqslant -\eta\log\left(\frac{u\cdot\rho^{(t)}}{I^{(t)}}\right) + \frac{4\eta}{3}\epsilon_I\frac{u\cdot\rho^{(t)}}{I^{(t)}} + \epsilon_Z + \frac{2\eta^2}{9r_{\min}^2}, \tag{39}$$

$$\leqslant -\eta\log\left(\frac{u\cdot\rho^{(t)}}{I^{(t)}}\right) + \epsilon_I\frac{4\eta}{3r_{\min}} + \epsilon_Z + \frac{2\eta^2}{9r_{\min}^2}, \tag{40}$$

where we use the fact that $1 - e^x \leqslant -x$ for all x. Using a telescoping sum over $t \in [T]$, we have

$$\begin{aligned} -D\left(u \| w^{(1)}\right) &\leqslant D\left(u \| w^{(T+1)}\right) - D\left(u \| w^{(1)}\right) & (41) \\ &\leqslant \eta \sum_{t=1}^{T} \left(\left(w^{(t)} \cdot \rho^{(t)} \right) - \log\left(u \cdot \rho^{(t)} \right) \right) + \epsilon_I \frac{4\eta T}{3 r_{\min}} + T\epsilon_Z + \frac{2\eta^2 T}{9 r_{\min}^2}. & (42) \end{aligned}$$

Rearranging the terms, we obtain

$$\sum_{t=1}^{T} \log\left(w^{(t)} \cdot \rho^{(t)} \right) \geqslant \sum_{t=1}^{T} \log\left(u \cdot \rho^{(t)} \right) - \frac{D(u \| w^{(1)})}{\eta} - \epsilon_I \frac{4T}{3 r_{\min}} - \frac{T\epsilon_Z}{\eta} - \frac{2\eta T}{9 r_{\min}^2}. \quad (43)$$

Since we let $w^{(t1)} = \left(\frac{1}{n}, \cdots, \frac{1}{n}\right)$, we have $D(u \| w^{(1)}) \leqslant \log n$. Setting

$$\epsilon_I = \frac{3\eta}{4 r_{\min}}, \qquad \epsilon_Z = 0, \quad (44)$$

gives

$$\sum_{t=1}^{T} \log\left(w^{(t)} \cdot \rho^{(t)} \right) \geqslant \sum_{t=1}^{T} \log\left(u \cdot \rho^{(t)} \right) - \frac{3}{r_{\min}} \sqrt{\frac{2\log n}{T}} \quad (45)$$

Setting

$$\epsilon_I = \frac{3\eta}{4 r_{\min}}, \qquad \epsilon_Z = \frac{\eta^2}{r_{\min}}, \quad (46)$$

gives

$$\sum_{t=1}^{T} \log\left(w^{(t)} \cdot \rho^{(t)} \right) \geqslant \sum_{t=1}^{T} \log\left(u \cdot \rho^{(t)} \right) - \frac{9}{r_{\min}} \sqrt{\frac{2\log n}{T}} \quad (47)$$

Thus, we obtain the desired bounds. □

3.4 Classically-sampled inner product

We show an algorithm where we classically sample the inner product. This algorithm does not offer any reduction in the run time, and the transaction cost is the same as in Algorithm 2. We present the algorithm to provide a gradual transition to the quantum algorithm, as the correctness analysis will be similar for the quantum algorithm. First, restate a lemma on classically-sampled inner products as follows. We would like to highlight that for all $t \in [T]$, the strategies $w^{(t)}$ are the same for Algorithm 1 and Algorithm 2, but, due to this erroneous update, are different for the following Algorithm 3 and Algorithm 4. The following lemma estimates inner products with relative error. As in our portfolio setting, the vector x has a lower bound for its entries.

Lemma 1 (Inner product estimation). *Let $\delta \in (0,1)$. Given query access to $x \in [x_{\min}, 1]^n$ and ℓ_1-sampling access to a probability vector $p \in [0,1]^n$, we can determine, with success probability at least $1-\delta$, the inner product $\alpha := p \cdot x \in [x_{\min}, 1]$ to multiplicative error $\epsilon_I \leqslant x_{\min}$, with $O\left(\frac{1}{\epsilon_I^2 x_{\min}} \log \frac{1}{\delta}\right)$ queries and samples, and $\tilde{O}\left(\frac{1}{\epsilon_I^2 x_{\min}} \log \frac{1}{\delta}\right)$ time complexity.*

Proof. Consider the additive version of this lemma as given in Reference [55, 47, 56, 57], adapted to the ℓ_1 case: Let X be a random variable with outcome x_j with probability p_j. Note that $\mathbb{E}[X] = \sum_j p_j x_j = \alpha$ and $Var(X) \leqslant \sum_j x_j^2 p_j \leqslant \alpha$. Apply the median-of-means method [56] on

$\frac{27}{\epsilon^2}\log\frac{1}{\delta}$ samples of X to be within $\epsilon\sqrt{Var(X)} \leqslant \epsilon\sqrt{\alpha}$ of $p \cdot x$ with probability at least $1 - \frac{\delta}{2}$ using $O\left(\frac{1}{\epsilon^2}\log\frac{1}{\delta}\right)$ queries.

For the multiplicative estimation, run the above algorithm with the precision parameter being set to $\epsilon = \epsilon_I$. We obtain an estimate $\widetilde{\alpha}_1$ of the inner product $\alpha = p \cdot x$ such that $|\alpha - \widetilde{\alpha}_1| \leqslant \epsilon_I\sqrt{\alpha}$, Then, re-run the algorithm with precision $\epsilon = \epsilon_I\sqrt{\widetilde{\alpha}_1}/2$. We will in turn obtain an estimate $\widetilde{\alpha}_2$ such that

$$|\alpha - \widetilde{\alpha}_2| \leqslant \frac{\epsilon_I\sqrt{\widetilde{\alpha}_1}}{2}\sqrt{\alpha} \leqslant \frac{\epsilon_I\sqrt{\alpha+\epsilon_I}}{2}\sqrt{\alpha} \leqslant \epsilon_I\alpha. \tag{48}$$

This costs $O\left(\frac{1}{\epsilon_I^2 x_{\min}}\log\frac{1}{\delta}\right)$ queries to obtain the desired guarantee. □

Algorithm 3 Approximate sampling-based Online Portfolio Optimization Algorithm

```
Input: $n$, $\eta$, $T$, $C$, $\delta$.
Initialize $w^{(1)} = (\frac{1}{n}, \cdots, \frac{1}{n}) \in \mathbb{R}^n$.
for $t = 1$ to $T$ do
    Prepare sampling data structure for $w^{(t)}$ using Fact 1.
    $i^{(t)} \leftarrow$ Sample from $w^{(t)}$.
    Invest in asset $i^{(t)}$ at cost $C$.
    Wait until end of day.
    Receive price relative vector $\rho^{(t)}$.
    $\tilde{I}^{(t)} \leftarrow$ estimate inner product $w^{(t)} \cdot \rho^{(t)}$ via Lemma 1 with relative error $\epsilon_I = \frac{3\eta}{4r_{\min}}$ and success probability $1 - \frac{\delta}{T}$.
    for $i = 1$ to $n$ do
        $w_i^{(t+1)} \leftarrow EEG(\eta, w^{(t)}, \rho^{(t)})_i$, with $\tilde{I}^{(t)}$ and exact norm computation.
    end for
end for
Output: $LS^C_{\text{samp}} := \frac{1}{T}\sum_{t=1}^{T}\log\rho^{(t)}_{i^{(t)}}$.
```

Thm. 3 implies the corollary below.

Corollary 2 (Guarantee and run time of Algorithm 3). *Let $\delta \in (0, 1/3)$. Algorithm 3 with $\eta = 2r_{\min}\sqrt{\frac{2\log n}{T}}$, achieves*

$$LS^* - LS^C_{samp} \leqslant \frac{6}{r_{\min}}\sqrt{\frac{\log\frac{n}{\delta}}{2T}}. \tag{49}$$

with success probability at least $1 - 3\delta$ in $O\left(Tn + \frac{T^2}{r_{\min}}\log\left(\frac{1}{\delta}\right)\right)$ time and incurs a transaction cost of $O(TC)$.

Proof. For the guarantee, we use Thm. 2 and Thm 3, and omit more detailed steps here. Aside from the inner product sampling, the run time and transaction cost is the same as in Thm. 2. A single inner product sampling to accuracy $\epsilon_I = \frac{3}{2}\sqrt{\frac{2\log n}{T}}$ takes time $O\left(\frac{T}{r_{\min}}\log\frac{1}{\delta}\right)$, and is performed T times in the algorithm. A union bound of all steps in the algorithm succeeding and the Hoeffding bound leads to the stated total success probability. □

3.5 The quantum online portfolio optimization algorithm

We now present our quantum online portfolio optimization algorithm and its analysis. We change the input assumption to a natural quantum extension of the classical input. The correctness guarantee essentially follows from Theorem 3. We obtain a quadratic speedup in the run time compared to the classical algorithm. Our quantum online portfolio optimization algorithm makes use of the following procedures: quantum state preparation, norm estimation and inner product estimation. Before we present the main algorithm, we will introduce these quantum subroutines.

In the quantum setting, instead of classical access to the price relatives we assume quantum access to the price relatives. The online nature of the problem is given by the fact that we obtain these oracles at the different times.

Data Input 1 (Online gain oracles)**.** *Let $\rho^{(1)}, \cdots, \rho^{(T)}$ be price relatives with $\max_{i\in[n]} \rho_i^{(t)} = 1$ and $\rho_i^{(t)} \geqslant r_{\min} > 0$ for all $i \in [n], t \in [T]$. Define the unitary $U_{\rho^{(t)}}$ operating on $O(\log n)$ quantum bits such that for all $j \in [n]$, $U_{\rho^{(t)}} |j\rangle \left|\vec{0}\right\rangle = |j\rangle \left|\rho_j^{(t)}\right\rangle$. At time $t \in [T]$ (end of day), assume access to unitaries $U_{\rho^{(1)}}, \cdots, U_{\rho^{(t)}}$.*

The update rule Eq. (1) can be rewritten in terms of a sum over all previous price relatives and inner products, as the following observation shows.

Fact 2. *Let $w^{(1)} = \left(\frac{1}{n}, \cdots, \frac{1}{n}\right)$. Given the update rule Eq. (1), we can express, for $t \in [T], i \in [n]$,*

$$w_i^{(t+1)} = \frac{\exp\left(\eta \sum_{t'=1}^{t} \frac{\rho_i^{(t)}}{w^{(t')}\cdot\rho^{(t')}}\right)}{\sum_{j=1}^{n} \exp\left(\eta \sum_{t'=1}^{t} \frac{\rho_j^{(t')}}{w^{(t')}\cdot\rho^{(t')}}\right)}. \tag{50}$$

Given Data Input 1, the following computations can be performed in superposition of the index $i \in [n]$ for the assets.

Lemma 2. *Let $t \leqslant T$. Let there be given the set of unitaries $U_{\rho^{(t')}}$ for $t' \in [t-1]$ as in Data Input 1, a vector $\tilde{I} \in \mathbb{R}^{t-1}$, and some reals $\eta, a \in \mathbb{R}$. There exists unitary operators performing the following computations:*

$$|i\rangle |\bar{0}\rangle \rightarrow |i\rangle \left| \sum_{t'=1}^{t-1} \frac{\rho_i^{(t')}}{\tilde{I}^{(t')}} \right\rangle, \quad |i\rangle |\bar{0}\rangle \rightarrow |i\rangle \left|q_i^{(t)}\right\rangle, \quad |i\rangle |\bar{0}\rangle \rightarrow |i\rangle \left|\frac{q_i^{(t)}}{a}\right\rangle, \tag{51}$$

where $q_i^{(t)} = \exp\left(\eta \sum_{t'=1}^{t-1} \frac{\rho_i^{(t')}}{\tilde{I}^{(t')}}\right)$ to sufficient numerical precision. These computations take $O(T)$ queries and to the data input and requires $O(T + \log n)$ qubits and quantum gates.

Proof. With a computational register involving $O(T)$ ancilla qubits for the gains and the ratios, perform

$$\begin{aligned}
|j\rangle |\bar{0}\rangle &\rightarrow |j\rangle \left|\rho_j^{(1)}\right\rangle \cdots \left|\rho_j^{(t-1)}\right\rangle |\bar{0}\rangle && (52)\\
&\rightarrow |j\rangle \left|\rho_j^{(1)}\right\rangle \cdots \left|\rho_j^{(t-1)}\right\rangle \left|\frac{\rho_j^{(1)}}{\tilde{I}^{(1)}}\right\rangle \cdots \left|\frac{\rho_j^{(t-1)}}{\tilde{I}^{(t-1)}}\right\rangle |\bar{0}\rangle && (53)\\
&\rightarrow |j\rangle \left|\rho_j^{(1)}\right\rangle \cdots \left|\rho_j^{(t-1)}\right\rangle \left|\frac{\rho_j^{(1)}}{\tilde{I}^{(1)}1}\right\rangle \cdots \left|\frac{\rho_j^{(t-1)}}{\tilde{I}^{(t-1)}}\right\rangle \left|\sum_{t'=1}^{t-1} \frac{\rho_j^{(t')}}{\tilde{I}^{(t')}}\right\rangle, && (54)
\end{aligned}$$

to sufficient accuracy using the oracles and quantum circuits for basic arithmetic operations. Uncomputing the intermediate registers with additional queries gives us the desired result. In addition, computations of $|i\rangle \left|q_i^{(t)}\right\rangle$ and $|i\rangle \left|\frac{q_i^{(t)}}{a}\right\rangle$ can be achieved using the quantum circuits for basic arithmetic operations. □

We restate the quantum state preparation, norm and inner product estimation procedure from Reference [58, 59, 60, 61, 47] for the convenience of the reader.

Lemma 3 (Quantum state preparation and norm estimation)**.** *Given a vector $v \in [0,1]^n$ with $\max_j v_j = 1$ and quantum access to v. Then:*

(i) Let $\epsilon > 0$ and $\delta \in (0,1)$. There exists a quantum algorithm that outputs an estimate $\tilde{z}$ of the ℓ_1-norm $\|v\|_1$ of v, such that $|\|v\|_1 - \tilde{z}| \leqslant \epsilon\|v\|_1$, with probability at least $1-\delta$. The algorithm uses $O\left(\frac{\sqrt{n}}{\epsilon}\log\left(\frac{1}{\delta}\right)\right)$ queries and $\tilde{O}\left(\frac{\sqrt{n}}{\epsilon}\log\left(\frac{1}{\delta}\right)\right)$ gates.

(ii) Let $\zeta \in (0,1/2]$ and $\tilde{z} > 0$ be given such that $|\tilde{z} - \|v\|_1| \leqslant \zeta\|v\|_1$. Let $\delta \in (0,1)$. An approximation $|\tilde{p}\rangle = \sum_{j=1}^{n} \sqrt{\tilde{p}_j}\,|j\rangle$ to the state $|v\rangle = \sum_{j=1}^{n} \sqrt{\frac{v_j}{\|v\|_1}}\,|j\rangle$ can be prepared with probability $1-\delta$ using $O(\sqrt{n}\log\frac{1}{\delta})$ calls to the unitary of (i) and $\tilde{O}\left(\sqrt{n}\log\left(\frac{1}{\delta}\right)\right)$ gates. The approximation in ℓ_1-norm of the probabilities is $\left\|\tilde{p} - \frac{v}{\|v\|_1}\right\|_1 \leqslant 2\zeta$.

Proof. There exists a unitary operator that prepares the state $\frac{1}{\sqrt{n}}\sum_{j=1}^{n}|j\rangle\left(\sqrt{v_j}\,|0\rangle + \sqrt{1-v_j}\,|1\rangle\right)$ with two queries to the vector, a controlled rotation, and $O\big(\log n\big)$ gates [61]. Using this unitary, the proof of part (i) is as in References [58, 59, 47]. For part (ii), note that for $i \in [n]$, we have that $\tilde{p}_i = \frac{v_i}{\tilde{z}}$ and $\|v\|_1 - \tilde{z} \leqslant |\|v\|_1 - \tilde{z}| \leqslant \zeta\|v\|_1$. Hence, $\frac{1}{\tilde{z}} \leqslant \frac{1}{\|v\|_1(1-\zeta)}$. Then, $\left\|\tilde{p} - \frac{v}{\|v\|_1}\right\|_1 = \sum_{i=1}^{n}\left|\frac{v_i\|v\|_1 - v_i\tilde{z}}{\tilde{z}\|v\|_1}\right| \leqslant \frac{\zeta}{1-\zeta}\sum_{i=1}^{n}\frac{v_i}{\|v\|_1} \leqslant 2\zeta$. □

Next, we show a lemma on the estimation of inner products. Using quantum maximum finding, we are able to use the norm computation of Lemma 3 for outputting the inner product.

Lemma 4 (Quantum inner production estimation with relative accuracy [47])**.** *Let $\epsilon, \delta \in (0,1)$ and given quantum access to two non-zero vectors $u, v \in [0,1]^n$. Then, an estimate $\widetilde{IP}$ for the inner product can be obtained such that $\left|\widetilde{IP} - u \cdot \frac{v}{\|v\|_1}\right| \leqslant \epsilon\ u \cdot \frac{v}{\|v\|_1}$ with success probability $1-\delta$. This requires $O\left(\frac{\sqrt{n}}{\epsilon}\log\left(\frac{1}{\delta}\right)\right)$ queries and $\tilde{O}\left(\frac{\sqrt{n}}{\epsilon}\log\left(\frac{1}{\delta}\right)\right)$ quantum gates.*

Proof. First, use quantum maximum finding [62] to find $v_{\max}$ with success probability $1-\frac{\delta}{4}$ using $O(\sqrt{n}\log\frac{1}{\delta})$ queries and $\tilde{O}(\sqrt{n}\log\frac{1}{\delta})$ quantum gates. Next, apply Lemma 3 on the normalized vector $\frac{v}{v_{\max}}$ to obtain an estimate γ_v of of its 1-norm $\left\|\frac{v}{v_{\max}}\right\|_1$ to relative error $\epsilon_v = \epsilon/4$ with success probability $1-\frac{\delta}{4}$. This uses $O(\frac{\sqrt{n}}{\epsilon}\log\frac{1}{\delta})$ queries and $\tilde{O}(\frac{\sqrt{n}}{\epsilon}\log\frac{1}{\delta})$ quantum gates.

Let $s \in [0,1]^n$ be the Hadamard product of u and v, i.e., $s_i = u_i \cdot v_i$ for all $i \in [n]$. Use quantum maximum finding again to find $s_{\max}$ with success probability $1-\frac{\delta}{4}$ using $O(\sqrt{n}\log\frac{1}{\delta})$ queries and $\tilde{O}(\sqrt{n}\log\frac{1}{\delta})$ quantum gates.

If $s_{\max} = 0$, then $\widetilde{IP} = 0$. Otherwise, apply Lemma 3 to the normalized vector $\frac{s}{s_{\max}}$ to obtain an estimate γ_s of its 1-norm $\left\|\frac{s}{s_{\max}}\right\|_1$ to relative error $\epsilon_s = \epsilon/4$ with success probability at least $1-\frac{\delta}{4}$. This uses $O\left(\frac{\sqrt{n}}{\epsilon}\log\frac{1}{\delta}\right)$ queries and $\tilde{O}\left(\frac{\sqrt{n}}{\epsilon}\log\frac{1}{\delta}\right)$ quantum gates.

Now, we have $\left|\frac{\gamma_s s_{\max}}{\gamma_v v_{\max}} - \frac{u\cdot v}{\|v\|_1}\right| \leqslant \frac{1}{\|v\|_1\gamma_v v_{\max}}\left(u\cdot v\,|\|v\|_1 - \gamma_v v_{\max}| + \|v\|_1\,|u\cdot v - \gamma_s s_{\max}|\right) \leqslant \frac{u\cdot v}{\gamma_v v_{\max}}\left(\epsilon_v + \epsilon_s\right) \leqslant \frac{u\cdot v}{\|v\|_1}\frac{\epsilon_v+\epsilon_s}{1-\epsilon_v} \leqslant \epsilon\frac{u\cdot v}{\|v\|_1}$, since $\epsilon_v < 1/2$. Let $\widetilde{IP} = \frac{\gamma_s s_{\max}}{\gamma_v v_{\max}}$, which leads to the desired bound. The total success probability follows from the union bound. □

We now present our quantum online portfolio optimization Algorithm 4 and its analysis. The correctness guarantee uses Theorem 3. Using Lemma 4 and the other quantum subroutines we obtain a quadratic speedup in the run time compared to that algorithm. The following theorem gives our main result for the regret and the run time of Algorithm 4.

Algorithm 4 Quantum Online Portfolio Optimization Algorithm

```
Input: $n$, $\eta$, $T$, $C$, $\delta$
Initialize $U_{\rho^{(0)}} = \mathbb{I}$, $\tilde{I} = 1$ and $q^{(1)} = (1, \cdots, 1)$, $U_{q^{(1)}} = \mathbb{I}$.
for $t = 1$ to $T$ do
    $q_{\max} \leftarrow$ Find the largest element of $q^{(t)}$ using $U_{q^{(t)}}$ and quantum maximum finding [62] with success probability $1 - \frac{\delta}{4T}$.
    $\tilde{Z}^{(t)} \leftarrow$ Estimate the norm of $\frac{q^{(t)}}{q_{\max}}$ using $U_{q^{(t)}}$, $q_{\max}$, Lemma 2, and Lemma 3(i), with relative error $\epsilon_Z = \frac{\eta^2}{r_{\min}}$ and success probability $1 - \frac{\delta}{4T}$.
    $U_{w^{(t)}} \leftarrow$ Prepare quantum circuit for approximating $\left|\tilde{w}^{(t)}\right\rangle$ of the quantum state $\left|w^{(t)}\right\rangle$, where $w^{(t)} = \frac{q^{(t)}}{\|q^{(t)}\|_1}$, using $U_{q^{(t)}}$, $q_{\max}$, $\tilde{Z}^{(t)}$ , Lemma 2, Lemma 3(ii), with success probability $1 - \frac{\delta}{4T}$.
    $j^{(t)} \leftarrow$ Prepare $U_{w^{(t)}} |\bar{0}\rangle$ and measure in the computational basis.
    Invest in $j^{(t)}$ at cost $C$.
    Wait until end of day.
    Receive price relative oracle $U_{\rho^{(t)}}$.
    $\rho^{(t)}_{j^{(t)}} \leftarrow$ Query $U_{\rho^{(t)}}$ with $\left|j^{(t)}\right\rangle |0\rangle$.
    $\tilde{I}^{(t)} \leftarrow$ Estimate $\tilde{w}^{(t)} \cdot \rho^{(t)}$ using $U_{w^{(t)}}$, $U_{\rho^{(t)}}$, and Lemma 4, with relative error $\epsilon_I = \frac{3\eta}{4r_{\min}}$ and success probability $1 - \frac{\delta}{4T}$.
    $U_{q^{t+1}} \leftarrow$ Prepare quantum circuit to compute $q^{(t+1)} = \exp\left(\eta \sum_{t'=1}^{t} \frac{\rho^{(t')}}{\tilde{I}^{(t')}}\right)$ using $\{U_{\rho^{(t')}}\}_{t'=1}^{t}$ and Lemma 2.
end for
Output: $LS^{Q}_{\text{samp}} := \frac{1}{T} \sum_{t=1}^{T} \log \rho^{(t)}_{j^{(t)}}$.
```

Theorem 4 (Quantum online portfolio optimization)**.** *Let $\delta \in (0, 1/3)$. Algorithm 4 with $\eta = 2r_{\min}\sqrt{\frac{2\log n}{T}}$, outputs $LS^Q_{samp} := \frac{1}{T}\sum_{t=1}^{T}\log \rho^{(t)}_{i(t)}$ with the regret bound*

$$LS^* - LS^Q_{samp} \leqslant \frac{20}{r_{\min}}\sqrt{\frac{\log\frac{n}{\delta}}{2T}}, \tag{55}$$

with success probability at least $1-3\delta$. The total run time is $\tilde{O}\left(\frac{T^3\sqrt{n}}{r_{\min}}\log\left(\frac{1}{\delta}\right)\right)$ and the transaction cost is $O(TC)$.

Proof. Condition the following argument on all probabilistic steps of the algorithm succeeding, which occurs with probability $1-\delta$ from the union bound. At each time step $t \in [T]$, the quantum algorithm produces a portfolio vector $\tilde{w}^{(t)}$. Similar to the proof of Thm. 2, the expected gain using samples from these portfolio vectors is bounded by

$$\mathbb{E}_{\tilde{w}}[LS^Q_{\text{samp}}] \leqslant \frac{1}{T}\sum_{t=1}^{T}\log\left(\tilde{w}^{(t)}\cdot\rho^{(t)}\right) =: \widetilde{LS}. \tag{56}$$

Similar to the analysis of Thm. 2, we obtain

$$\mathbb{P}\left[\left|LS^Q_{\text{samp}} - \widetilde{LS}\right| \leqslant \log\left(\frac{1}{r_{\min}}\right)\sqrt{\frac{\log(\frac{1}{\delta})}{2T}}\right] \geqslant 1 - 2\delta. \tag{57}$$

Using Theorem 3 and Eq. (57), the regret is bounded by

$$\begin{aligned} LS^* - LS^Q_{\text{samp}} &= LS^* - \widetilde{LS} + \widetilde{LS} - LS^Q_{\text{samp}} & (58)\\ &\leqslant \frac{9}{r_{\min}}\sqrt{\frac{2\log n}{T}} + \log\left(\frac{1}{r_{\min}}\right)\sqrt{\frac{\log\frac{1}{\delta}}{2T}} & (59)\\ &\leqslant \left(\frac{20}{r_{\min}}\right)\sqrt{\frac{\log\frac{n}{\delta}}{2T}}, & (60)\end{aligned}$$

with success probability at least $1-2\delta$. By the union bound, the total success probability is at least $1-3\delta$ by taking into account the success probability of the algorithm. For the run time, consider that $U_{q(t)}$ costs $O(T+\log n)$ by Lemma 2. Using the values for η, ϵ_I and ϵ_Z, the total run time is

$$\begin{aligned} & O(T(\text{quantum maximum finding} + \text{quantum norm estimation} & (61)\\ + \ & \text{quantum state preparation} + \text{quantum inner product estimation})) & (62)\\ \subseteq \ & \tilde{O}\left(T^2\left(\sqrt{n}\log\left(\frac{1}{\delta}\right) + \frac{\sqrt{n}}{\epsilon_Z}\log\left(\frac{1}{\delta}\right) + \sqrt{n}\log\left(\frac{1}{\delta}\right) + \frac{\sqrt{n}}{\epsilon_I}\log\left(\frac{1}{\delta}\right)\right)\right) & (63)\\ \subseteq \ & \tilde{O}\left(T^2\sqrt{n}\left(1 + \sqrt{\frac{T}{\log n}} + \frac{T}{r_{\min}\log n}\right)\log\left(\frac{1}{\delta}\right)\right) & (64)\\ \subseteq \ & \tilde{O}\left(\frac{T^3\sqrt{n}}{r_{\min}}\log\left(\frac{1}{\delta}\right)\right). & (65)\end{aligned}$$

□

4 Discussion and conclusion

The online setting is more general than the offline setting in that it allows for the input to be given sequentially, where the sequence could be chosen adversarially. The adversarial property implies that the inputs could be selected with the knowledge of the present state of the algorithm, say, with the knowledge of our portfolio vector. The regret bounds hold nevertheless, also in the quantum setting. This online setting is rather natural for certain portfolio optimization situations, where the investment strategy can be inferred by other market participants from transaction data. Online algorithms in the portfolio optimization context have been studied in practice in References [10, 63, 64, 65].

We have devised a quantum online portfolio optimization algorithm that runs in time $\tilde{O}\left(\frac{T^3\sqrt{n}}{r_{\min}}\log\left(\frac{1}{\delta}\right)\right)$ and has transaction cost that is independent on the number of assets. Our quantum algorithm achieves a slightly worse (by a constant factor) regret bound, but is more space efficient as compared to its classical counterpart [10], not considering the space requirement for the input oracles. The classical online portfolio optimization algorithm by Reference [10] uses linear (in terms of the number of assets) time and space to update the portfolio vector in every iteration. In our quantum algorithm, we leverage on the fact that the portfolio vectors can be computed efficiently via simple unitaries that perform arithmetic operations to save on the space/memory of the algorithm.

In our setting, the transaction cost was taken to be a constant for each investment, independent of the size of the investment. This models the fact that for each investment some fixed amount of work has to be performed, e.g., the communication of the trade between counterparties and the transfer of the asset. This type of transaction cost serves to illustrate the benefits of the sampling algorithm over the standard algorithm. For future work, one can consider imposing additional constraints on the portfolio optimization problem. For instance, a common optimization is to minimize transaction cost via including a term $\|w^{(t)} - w^{(t-1)}\|_1$ in the portfolio optimization problem or consider portfolio optimization in the robust setting, where the parameters belong to an uncertainty set. Various flavours of robustness such as constraint, objective and relative robustness in conjunction with different types of uncertainty sets [66] are also worth investigating.

5 Acknowledgements

Research at CQT is funded by the National Research Foundation, the Prime Minister's Office, and the Ministry of Education, Singapore under the Research Centres of Excellence programme's research grant R-710-000-012-135. We also acknowledge funding from the Quantum Engineering Program (QEP 2.0) under grant NRF2021-QEP2-02-P05.

References

[1] Martin Zinkevich. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. *Proceedings, Twentieth International Conference on Machine Learning*, 2(February):928–935, 2003.

[2] Elad Hazan. Introduction to Online Convex Optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[3] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *arXiv preprint cs/0408007*, 2004.

[4] Elad Hazan, Alexander Rakhlin, and Peter Bartlett. Adaptive online gradient descent. *Advances in Neural Information Processing Systems*, 20, 2007.

[5] Huahua Wang and Arindam Banerjee. Randomized block coordinate descent for online and stochastic optimization. *arXiv preprint arXiv:1407.0107*, 2014.

[6] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. *Journal of the ACM*, 63(2), 2016.

[7] Nicol N Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-Newton method for online convex optimization. In *Artificial intelligence and statistics*, pages 436–443. PMLR, 2007.

[8] Sanjeev Arora, Rong Ge, Ravi Kannan, and Ankur Moitra. Computing a nonnegative matrix factorization-provably. *SIAM Journal on Computing*, 45(4):1582–1611, 2016.

[9] David P. Helmbold and Manfred K. Warmuth. Learning permutations with exponential weights. *Journal of Machine Learning Research*, 10(10):1705–1736, 2009.

[10] David P. Helmbold, Robert E. Schapire, Yoram Singer, and Manfred K. Warmuth. On-line portfolio selection using multiplicative updates. *Mathematical Finance*, 8(4):325–347, 1998.

[11] Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 385–394, 2005.

[12] Sébastien Bubeck and Ronen Eldan. Multi-scale exploration of convex functions and bandit convex optimization. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 583–589, Columbia University, New York, New York, USA, 2016. PMLR.

[13] Sébastien Bubeck, Ronen Eldan, and Yin Tat Lee. Kernel-based methods for bandit convex optimization. *J. ACM*, 68(4), 2021.

[14] Tor Lattimore. Improved regret for zeroth-order adversarial bandit convex optimisation. *Mathematical Statistics and Learning*, 2(3):311–334, 2020.

[15] Harry Markowitz. Portfolio Optimization. *The Journal of Finance*, 7(1):77–91, 1952.

[16] Laurent El Ghaoui, F. Oustry, and M. Oks. Worst-Case Value-at-Risk and Robust Asset Allocation: A Semidefinite Programming Approach. Technical report, EECS Department, University of California, Berkeley, 2000.

[17] Prisadarng Skolpadungket, Keshav Dahal, and Napat Harnpornchai. Portfolio optimization using multi-objective genetic algorithms. In *2007 IEEE Congress on Evolutionary Computation*, pages 516–523. IEEE, 2007.

[18] Antonin Ponsich, Antonio Lopez Jaimes, and Carlos A Coello Coello. A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Transactions on Evolutionary Computation*, 17(3):321–344, 2012.

[19] Yu-Min Yen and Tso-Jung Yen. Solving norm constrained portfolio optimization via coordinate-wise descent algorithms. *Computational Statistics & Data Analysis*, 76:737–759, 2014.

[20] Can Berk Kalaycı, Ökkeş Ertenlice, Hasan Akyer, and Hakan Aygören. A review on the current applications of genetic algorithms in mean-variance portfolio optimization. *Pamukkale University Journal of Engineering Sciences*, 23:470–476, 2017.

[21] Ameer Tamoor Khan, Xinwei Cao, Shuai Li, Bin Hu, and Vasilios N Katsikis. Quantum beetle antennae search: a novel technique for the constrained portfolio optimization problem. *Science China Information Sciences*, 64(5):1–14, 2021.

[22] Edwin J. Elton, Martin J. Gruber, Stephen J. Brown, and William N. Goetzmann. *Modern Portfolio Theory and Investment Analysis*. John Wiley & Sons, 2014.

[23] Alan J. King. Asymmetric risk measures and tracking models for portfolio optimization under uncertainty. *Annals of Operations Research*, 45(1):165–177, 1993.

[24] Hiroshi Konno and Hiroaki Yamazaki. Mean-Absolute Deviation Portfolio Optimization Model and Its Applications to Tokyo Stock Market. *Management Science*, 37(5):519–531, 1991.

[25] Terence C. Mills. Stylized facts on the temporal and distributional properties of daily FT-SE returns. *Applied Financial Economics*, 7(6):599–604, 1997.

[26] Peter W Turnbull. A Review of Portfolio Planning Models for Industrial Marketing and Purchasing Management. *European Journal of Marketing*, 24:7–22, 1990.

[27] J.P Morgan. RiskMetricsTM—Technical Document. *... Trust Company of New York: New ...*, pages 2–296, 1996.

[28] R. Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, 26(7):1443–1471, 2002.

[29] Jyrki Kivinen and Manfred K. Warmuth. Additive versus exponentiated gradient updates for linear prediction. *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 209–218, 1995.

[30] Bin Li and Steven C H Hoi. Online portfolio selection : A survey y. *ACM Computing Surveys (CSUR)*, 46(3):1–36, 2014.

[31] Román Orús, Samuel Mugel, and Enrique Lizaso. Quantum computing for finance: Overview and prospects. *Reviews in Physics*, 4:1–13, 2019.

[32] Adam Bouland, Wim van Dam, Hamed Joorati, Iordanis Kerenidis, and Anupam Prakash. Prospects and challenges of quantum finance. *arXiv preprint arXiv:2011.06492*, 2020.

[33] Daniel J. Egger, Claudio Gambella, Jakub Marecek, Scott McFaddin, Martin Mevissen, Rudy Raymond, Andrea Simonetto, Stefan Woerner, and Elena Yndurain. Quantum Computing for Finance: State-of-the-Art and Future Prospects. *IEEE Transactions on Quantum Engineering*, 1:1–24, 2021.

[34] Dylan Herman, Cody Googin, Xiaoyuan Liu, Alexey Galda, Ilya Safro, Yue Sun, Marco Pistoia, and Yuri Alexeev. A Survey of Quantum Computing for Finance. *arXiv preprint arXiv:2201.02773*, 2022.

[35] Gili Rosenberg, Poya Haghnegahdar, Phil Goddard, Peter Carr, Kesheng Wu, and Marcos López De Prado. Solving the optimal trading trajectory problem using a quantum annealer. *IEEE Journal of Selected Topics in Signal Processing*, 10(6):1053–1060, 2016.

[36] Patrick Rebentrost and Seth Lloyd. Quantum computational finance: quantum algorithm for portfolio optimization. *arXiv preprint arXiv:1811.03975*, 2018.

[37] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15), 2009.

[38] Andrew M Childs, Robin Kothari, and Rolando D Somma. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM Journal on Computing*, 46(6):1920–1950, 2017.

[39] Iordanis Kerenidis, Anupam Prakash, and Dániel Szilágyi. Quantum algorithms for portfolio optimization. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, pages 147–155, 2019.

[40] Mark Hodson, Brendan Ruck, Hugh Ong, David Garvin, and Stefan Dulman. Portfolio rebalancing experiments using the Quantum Alternating Operator Ansatz. *arXiv preprint arXiv:1911.05296*, 2019.

[41] N. Slate, E. Matwiejew, S. Marsh, and J. B. Wang. Quantum walk-based portfolio optimisation. *Quantum*, 5:1–17, 2021.

[42] Romina Yalovetzky, Pierre Minssen, Dylan Herman, and Marco Pistoia. Nisq-hhl: Portfolio optimization for near-term quantum hardware. *arXiv preprint arXiv:2110.15958*, 2021.

[43] Srinivasan Arunachalam and Reevu Maity. Quantum boosting. In *International Conference on Machine Learning*, pages 377–387. PMLR, 2020.

[44] Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[45] Adam Izdebski and Ronald de Wolf. Improved quantum boosting. *arXiv preprint arXiv:2009.08360*, 2020.

[46] Adam Klivans and Raghu Meka. Learning graphical models using multiplicative weights. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 343–354. IEEE, 2017.

[47] Patrick Rebentrost, Yassine Hamoudi, Maharshi Ray, Xin Wang, Siyi Yang, and Miklos Santha. Quantum algorithms for hedging and the learning of Ising models. *Physical Review A*, 103(1):012418, 2020.

[48] Daochen Wang, Xuchen You, Tongyang Li, and Andrew M Childs. Quantum exploration algorithms for multi-armed bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10102–10110, 2021.

[49] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[50] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms.* Cambridge University Press, 2020.

[51] Jianhao He, Feidiao Yang, Jialin Zhang, and Lvzhou Li. Quantum algorithm for online convex optimization. *Quantum Science and Technology*, 7(2), 2022.

[52] Ohad Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *The Journal of Machine Learning Research*, 18(1):1703–1713, 2017.

[53] M.D. Vose. A Linear Algorithms for Generating Random Numbers with A Given Distribution. *IEEE Transactions on Software Engineering*, 17(972), 1991.

[54] A J Walker. New fast method for generating discrete random numbers with arbitrary frequency distributions. *Electronics Letters*, 10(8):127–128, 4 1974.

[55] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 217–228, 2019.

[56] Nai-Hui Chia, András Gilyén, Tongyang Li, Han-Hsuan Lin, Ewin Tang, and Chunhao Wang. Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning. In *Proceedings of the 52nd Annual ACM SIGACT symposium on theory of computing*, pages 387–400, 2020.

[57] Nai-Hui Chia, Han-Hsuan Lin, and Chunhao Wang. Quantum-inspired sublinear classical algorithms for solving low-rank linear systems. *arXiv preprint arXiv:1811.04852*, 2018.

[58] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.

[59] Tongyang Li, Shouvanik Chakrabarti, and Xiaodi Wu. Sublinear quantum algorithms for training linear and kernel-based classifiers. *36th International Conference on Machine Learning, ICML 2019*, pages 6784–6804, 2019.

[60] Joran van Apeldoorn and András Gilyén. Quantum algorithms for zero-sum games. *arXiv preprint arXiv:1904.03180*, 2019.

[61] Yassine Hamoudi, Patrick Rebentrost, Ansis Rosmanis, and Miklos Santha. Quantum and classical algorithms for approximate submodular function minimization. *Quantum Information and Computation*, 19(15-16):1325–1349, 2019.

[62] Christoph Dürr and Peter Høyer. A Quantum Algorithm for Finding the Minimum. *arXiv preprint quant-ph/9607014*, 1996.

[63] Xin Wang, Tao Sun, and Zhi Liu. Kernel-based aggregating learning system for online portfolio optimization. *Mathematical Problems in Engineering*, 2020, 2020.

[64] Bin Li, Jialei Wang, Dingjiang Huang, and Steven CH Hoi. Transaction cost optimization for online portfolio selection. *Quantitative Finance*, 18(8):1411–1424, 2018.

[65] Majid Khedmati and Pejman Azin. An online portfolio selection algorithm using clustering approaches and considering transaction costs. *Expert Systems with Applications*, 159:113546, 2020.

[66] Gerard Cornuejols and Reha Tütüncü. *Optimization methods in finance*, volume 5. Cambridge University Press, 2006.

A Useful inequality and lemma

Fact 3 (Hoeffding's inequality)**.** *Let $(\Omega, \Sigma, \mathbb{P})$ be a probability space, and consider random variables $Y_1, \cdots, Y_n$, where $Y_i : \Omega \to [c_i, d_i]$ for $i \in [n]$. Let $Z = \sum_{i=1}^{n} Y_i$. Then, for all $a > 0$,*

$$\mathbb{P}\left[|Z - \mathbb{E}[Z]| \geqslant a\right] \leqslant 2 \exp\left(-\frac{2a^2}{\sum_{i=1}^{n}(d_i - c_i)^2}\right). \tag{66}$$

Lemma 5. *Let $(\Omega, \Sigma, \mathbb{P})$ be a probability space, $A : \Omega \to \mathbb{R}$ be a random variable, $B, C, d \in \mathbb{R}$ where $C \leqslant B$ and $d > 0$. The following statement is true when $A(\omega), B, C$ are all positive or all negative:*

$$\mathbb{P}\left[|A - B| \geqslant d\right] \leqslant \mathbb{P}\left[|A - C| \geqslant d\right] \tag{67}$$

Proof. We prove the complement of Eq. (67):

$$\mathbb{P}\left[|A - B| \leqslant d\right] \geqslant \mathbb{P}[|A - C| \leqslant d] \tag{68}$$

Consider the set

$$\begin{aligned}
\{\omega \in \Omega : |A(\omega) - B| \leqslant d\} &= \{\omega \in \Omega : -d \leqslant A(\omega) - B \leqslant d\} && (69)\\
&= \{\omega \in \Omega : A(\omega) - d \leqslant B \text{ and } A(\omega) + d \geqslant B\} && (70)\\
&\supseteq \{\omega \in \Omega : A(\omega) - d \leqslant C \text{ and } A(\omega) - C + d \geqslant B - C\} && (71)\\
&\supseteq \{\omega \in \Omega : A(\omega) - d \leqslant C \text{ and } A(\omega) - C + d \geqslant 0\} && (72)\\
&= \{\omega \in \Omega : A(\omega) - d \leqslant C \text{ and } A(\omega) + d \geqslant C\} && (73)\\
&= \{\omega \in \Omega : |A(\omega) - C| \leqslant d\} && (74)
\end{aligned}$$

Therefore, Eq. (68) and its complement hold. □