Signal Processing 91 (2011) 890–905

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

A tree-weighting approach to sequential decision problems with multiplicative loss

Suleyman S. Kozat [a,*], Andrew C. Singer [b,1], Andrew J. Bean [b,1]

[a] Department of ECE, Koc University, Istanbul, Turkey
[b] Department of ECE, University of Illinois at Urbana-Champaign, Urbana, IL, USA

ARTICLE INFO

Article history:
Received 3 February 2010
Received in revised form
7 September 2010
Accepted 8 September 2010
Available online 1 October 2010

Keywords:
Universal
Convex-combination
Portfolio
Investment
Context-tree
Piecewise models

ABSTRACT

In this paper, we consider sequential decision problems in which the decision at each time is taken as a convex-combination of observations and whose performance metric is multiplicatively compounded over time. Such sequential decision problems arise in gambling, investing and in a host of signal processing applications from statistical language modeling to mixed-modality multimedia signal processing. Using a competitive algorithm framework, we construct sequential strategies that asymptotically achieve the performance of the best piecewise-convex strategy that could have been chosen by observing the entire sequence of outcomes in advance. Using the notion of context-trees, a mixture approach is able to asymptotically achieve the performance of the best choice of both the partitioning of the space of past observations and convex strategies within each region, for every sequence of outcomes. This performance is achieved with linear complexity in the depth of the context-tree, per decision. For the application of sequential investment, we also investigate transaction costs incurred for each decision. An explicit algorithmic description and examples demonstrating the performance of the algorithms are given. Our methods can be used to sequentially combine probability distributions produced by different statistical language models used in speech recognition or natural language processing and by different modalities in multimedia signal processing.

1. Introduction

In this paper, we consider sequential decision problems whose metric of performance is multiplicatively compounded over time and in which the decisions made at each time amount to picking a strategy taken as a convex-combination of the vector-valued outcomes. In this general framework that encompasses a number of applications, the observations, which are observed sequentially, are represented as vectors in the positive orthant, i.e., $\{\boldsymbol{x}[t]\}_{t\geq 1}$, $\boldsymbol{x}[t]\in\mathbb{R}_{+}^{m}$, where some entries of $\boldsymbol{x}[t]$ can be zero implying that the observations are simply vectors of nonnegative numbers. We represent our decision at time t as $\boldsymbol{b}[t]$, where $\boldsymbol{b}[t]\in\mathbb{R}_{+}^{m}$ and $\sum_{j=1}^{m} b_j[t]=1$ for all t. Based on the decision $\boldsymbol{b}[t]$ and then observing $\boldsymbol{x}[t]$, we incur the benefit or reward $\boldsymbol{b}[t]^T\boldsymbol{x}[t]$ at time t, yielding the accumulated gain over all past observations as $\prod_{t=1}^{n}\boldsymbol{b}^T[t]\boldsymbol{x}[t]$ for all n. In this paper, the goal is to maximize the accumulated gain over any possible and unknown $\{\boldsymbol{x}[t]\}_{t\geq 1}$ by sequentially choosing appropriate $\{\boldsymbol{b}[t]\}_{t\geq 1}$ for any n. The decisions $\{\boldsymbol{b}[t]\}_{t\geq 1}$ are "sequential" such that $\boldsymbol{b}[t]$ only depends on the past sequence of observations, i.e., $\boldsymbol{x}[1],\ldots,\boldsymbol{x}[t-1]$, but not on information from the future. Extensions of this basic framework for more general decision vectors, such as $\boldsymbol{b}[t]\in[-1,\infty)^m$ [1,2], and or observation vectors, such as $\boldsymbol{x}[t]\in[-1,\infty)^m$ [3], are also possible. We study the basic

* Corresponding author. Tel.: +90 212 3381540.
E-mail addresses: skozat@ku.edu.tr (S.S. Kozat), acsinger@illinois.edu (A.C. Singer), ajbean@illinois.edu (A.J. Bean).
[1] Tel.: +1 217 2449263.

0165-1684/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2010.09.007

structure of the problem here and our results can be readily extended to these more general cases.

Sequential decision problems whose metric of performance is multiplicatively compounded over time arise in a host of signal processing applications from statistical language modeling [4–6] and Gaussian mixture models [5] to mixed-modality multimedia signal processing [7,8]. We refer the reader to [9], and the references therein, for a more general discussion of such sequential decision problems. For concreteness of terminology, in this paper we will pay particular attention to the application of sequential investing in a market of m stocks, and note that the mathematical techniques used here apply more generally to the contexts discussed in [9]. In the investing context, the observation vectors $\{\boldsymbol{x}[t]\}_{t\geq 1}$ represent the market gains such that the jth entry $x_j[t]$ of the vector $\boldsymbol{x}[t]$ represents the gain achieved by the jth stock on the tth day of investment, as might be measured by the ratio of opening price of the jth stock on the tth day to the opening price of the jth stock on the $(t-1)$th trading day. The sequence of decisions $\{\boldsymbol{b}[t]\}_{t\geq 1}$ will correspond to a sequence of investments made by allocating a fraction of the current holdings, or wealth, of the player to each of m stocks or other financial instruments. The selection of such a "portfolio" amounts to choosing a particular weighting among stocks of the wealth available on a given investment period. An investment strategy at day t is represented by the "portfolio vector" $\boldsymbol{b}[t]$, $\boldsymbol{b}[t]\in\mathbb{R}_+^m$ and $\sum_{j=1}^m b_j[t]=1$ for all t. Each entry $b_j[t]$ corresponds to the fraction of the total wealth available to be invested in the jth stock on the tth day. The achieved wealth after n investment periods is given by the product of the gains achieved in each successive day of trading, i.e., $\prod_{t=1}^n \boldsymbol{b}^T[t]\boldsymbol{x}[t]$. In another context, $\boldsymbol{x}[t]$ could correspond to probabilities derived from various signal source models for a sequence of observations and their convex-combination would amount to a mixture distribution in a Bayesian mixture representation [5,6].

The sequential portfolio assignment problem studied here has been considered broadly in the signal processing [10], machine learning [3,2] and information theory [11,1] research literature. The objective is to select a sequence of investment strategies, or portfolios, for a market with a finite number of stocks, to maximize wealth over time over any possible deterministic observation vectors $\{\boldsymbol{x}[t]\}_{t\geq 1}$ without stochastic assumptions on $\{\boldsymbol{x}[t]\}_{t\geq 1}$. To define a meaningful performance measure, we define a competitive algorithm framework in which the goal is to perform well with respect to a candidate class of investment strategies, i.e., a competition class, over any possible $\{\boldsymbol{x}[t]\}_{t\geq 1}$. As an example, a problem studied extensively in this context is to find a sequential algorithm that asymptotically achieves the wealth of the best constant rebalanced portfolio tuned to the individual sequence of observation vectors. When the portfolio vector remains constant for all t, i.e., $\boldsymbol{b}[t]=\boldsymbol{b}$, corresponding to an apportionment of assets at each point in time that is a fixed constant convex-combination, $\boldsymbol{b}$, the strategy is called a "constant rebalanced portfolio" (CRP). In this context, we try to achieve $\max_{\boldsymbol{b}}\prod_{t=1}^n \boldsymbol{b}^T\boldsymbol{x}[t]$, where $\boldsymbol{b}$ is fixed and $\{\boldsymbol{x}[t]\}_{t\geq 1}$ is an arbitrary individual sequence, using a sequential algorithm. The goal is to find a sequential algorithm $\boldsymbol{b}[t]$ such that when it is applied to any deterministic and unknown $\{\boldsymbol{x}[t]\}_{t\geq 1}$ (which is revealed sequentially), it will achieve the gain $\prod_{t=1}^n \boldsymbol{b}[t]^T\boldsymbol{x}[t]$ that is close to $\max_{\boldsymbol{b}}\prod_{t=1}^n \boldsymbol{b}^T\boldsymbol{x}[t]$ for all n (or asymptotically equal to $\max_{\boldsymbol{b}}\prod_{t=1}^n \boldsymbol{b}^T\boldsymbol{x}[t]$ as n goes to infinity), without knowing n or $\{\boldsymbol{x}[t]\}_{t\geq 1}$ beforehand. If we can find such an algorithm, this algorithm is said to be competitive with respect to the class of all CRPs, since it asymptotically achieves the performance of even the best CRP in the class that is tuned to the underlying $\{\boldsymbol{x}[t]\}_{t\geq 1}$, for all n. In the context of probabilistic model combinations, such a CRP would amount to a constant weighting among constituent probabilistic models, i.e., an *a priori* weighting. Cover [11] presented an algorithm that asymptotically achieves the wealth of the best CRP for any sequence of observation vectors, that is, his algorithm can sequentially achieve nearly the same performance of an investment strategy that could only have been chosen in hindsight, after observing the entire sequence of stock market values in advance, but which was restricted to only select a CRP.

In the first part of this paper, we investigate sequential portfolios that compete against the best piecewise-constant rebalanced portfolios (PCRPs), which are a direct extension of CRPs. Here, instead of trying to achieve the performance of the best CRP that is tuned to the underlying sequence of observations, we try to achieve the performance of the best PCRP. In our framework, the space in which the sequence of observations lies is partitioned into a union of disjoint regions, over each of which, a CRP is fitted independently. This is a natural nonlinear extension to linear modeling where piecewise models are used to approximate more general nonlinear functions such as in [12]. As an example, suppose at trading period t, we divide the space $\boldsymbol{x}[t-1]\in\mathbb{R}_+^m$ as in Fig. 1 into K disjoint regions V_k, where $\cup_{k=1}^K V_k=\mathbb{R}_+^m$ (e.g., K=4 for Fig. 1). Here, if $\boldsymbol{x}[t-1]\in V_1$, then stock 1 outperformed stock 2 at trading day $t-1$ (however, both stocks lost value, i.e., $x_1[t-1]<1, x_2[t-1]<1$). If $\boldsymbol{x}[t-1]\in V_1\cup V_2$ where the gain of stock 1 was greater than that of stock 2 at trading day $t-1$, then investing in stock 1 more than stock 2 in the next trading period t may be a good idea. This strategy may work if there is useful information in the relative performance of various assets in the market. Hence, to follow this idea, we define the PCRP competition class by assigning a CRP to each region and

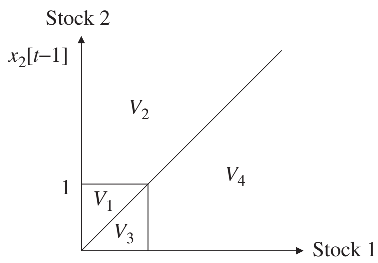

Fig. 1. A partition of the $\mathbb{R}_+^m$. Each axis corresponds to the observation (price-relative) of a given stock in the market.

such a PCRP invests during each trading day using the portfolio that depends on the relative performance of each stock on the previous day. For PCRPs, the portfolio used in each region is a fixed CRP, $\boldsymbol{b}_k$, such that if $\boldsymbol{x}[t-1] \in V_k$ then we invest with $\boldsymbol{b}_k$ at trading period t. The CRPs can be selected arbitrarily for each region. We point out that as the number of regions grows, the piecewise constant model can better approximate any fixed nonlinear portfolio $\boldsymbol{b} = \boldsymbol{f}(\boldsymbol{x}[t-1])$ for some arbitrary locally convex smoothly varying nonlinear function $\boldsymbol{f}(\cdot)$ [13]. However, we emphasize that neither the optimal partitioning of $\mathbb{R}_+^m$ nor the best CRPsfor each region are known in advance, and both depend on $\{\boldsymbol{x}[t]\}_{t\geq 1}$ and n.

Note that if the piecewise regions or the partition of $\mathbb{R}_+^m$ is fixed, the assignment of portfolio vectors to each region are known. As in the first part of this paper and in [11], independently applying the algorithm of [11] for each piecewise region will asymptotically achieve the performance of the optimal CRPs in each region. In the second part of this paper, we extend these results when the partitioning of the past observations space, i.e., partition of the $\mathbb{R}_+^m$, is also a design parameter that can be selected from a large class of possible partitions. In this sense, if we consider the partition information as the side-information, the side-information generating mechanism is also a design parameter [11]. The class of possible partitions will be compactly represented using a "context-tree" [14], which will be used to define a doubly exponential number of partitions. We have neither *a priori* knowledge of the selected partition nor the best model parameters, i.e., the best PCRP given that partition. Here, we demonstrate an algorithm that asymptotically achieves the performance of the best sequential portfolio (corresponding to a particular partition) from the doubly exponentially large class of such partitioned portfolios. To accomplish this, we use the notion of context-trees as shown in Fig. 2 which is explained in Section 2.2. By using context-trees, we are able to compete against the best partition among a doubly exponential number of possible partitions that can be embedded in a context-tree with computational complexity only linear in the depth of the context-tree.

Competition against the known side-information dependent CRP is investigated in [11] and then several different sequential algorithms have been introduced that also attempt to achieve the performance of the best CRP, albeit either with different bounds or different performance on historical data [3,15]. This basic problem has been extended to portfolios with side-information [11,3,16], transaction costs [17], margin and short sales [1,2], smoothly varying target classes [16], competition against the best switching constant rebalanced portfolios [18,10] and internal regret [19]. We emphasize that we only use Cover's algorithm as an example in our derivations and the methods we use are generic such that they can use other algorithms such as those in [3,19,15,20,21] in our theorems. Note that the alternative algorithms are provided since although the algorithm of [11] has "asymptotically" tight performance bounds, its exact implementation requires $O(n^{m-1})$ computational complexity per investment period. In this sense, the alternative algorithms such as the ones in [3,22,15] are introduced to provide both computational efficiency and logarithmic regret at the same time (if possible). These alternative algorithms are also experimentally shown to outperform the algorithm of [11] in certain scenarios, especially when the market has large number of stocks [3,21]. As an example, the exponentiated gradient based algorithm of [3] has linear complexity per investment period

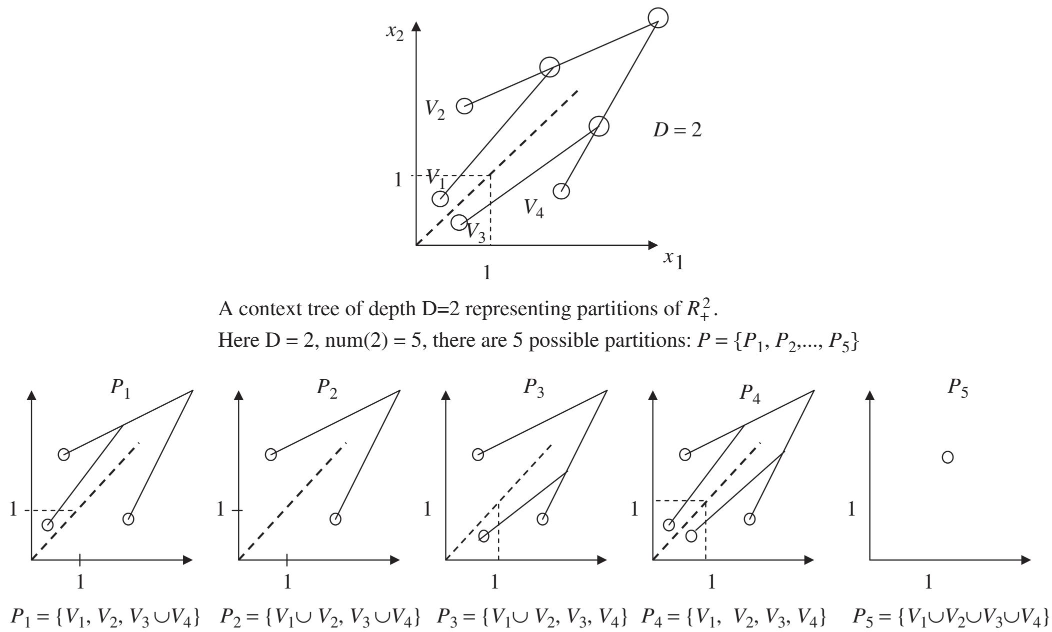

Fig. 2. A depth-2 context-tree. Each leaf of the context-tree corresponds to a region of $\mathbb{R}_+^m$. This context-tree represents five different partitions of $\mathbb{R}_+^m$, $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_5\}$.

with $O(\sqrt{2\log m/n})$ normalized regret and the follow-the-leader based algorithm of [22] has $O(m^3)$ complexity per investment period with $O(4m\log(n)/n)$ normalized regret (only when certain parameters are optimized in hindsight, unlike Cover's algorithm). To demonstrate the versatility and ease of our scheme for incorporating new algorithms into the studied framework, we use algorithms from [3,22] in addition to the algorithm from [11] in the Simulations section. Moreover, unlike [11], our model includes the presence of transactions costs and can be straightforwardly extended to investing on margin and short sales. While competition against CRPs was extended to more general target classes in [16,3], we point out that in all these cases considering side-information, the side-information generating mechanism or the side-information itself is known or fixed. Hence, in these results, the competition with respect to the side-information sequence is achieved by merely repeating the basic algorithm for each side-information value. However, in this paper, the side-information generating mechanism can also be selected by the competition class. Only in hindsight, one can determine which partition of the $\mathbb{R}_+^m$, i.e., the side-information, will yield the optimal growth. Without such a priori knowledge, our algorithm asymptotically achieves the performance of any such partition, i.e., the best side-information generating mechanism from this class.

Context-trees and the context-tree weighting algorithm have been used extensively in lossless source coding and related fields essentially to assign Bayesian mixture probabilities to binary sequences [23,14]. In these frameworks, context-trees are mainly used to efficiently calculate a weighted average of probabilities produced by an exponential number of models represented on the context-tree. However, in this paper, the purpose of using context-trees is not to directly calculate a weighted average of wealths produced by an exponential number of investment models, which was the main tool in [14,24,25] to achieve the performance of the best model. Here, we specifically design "an algorithm" that when applied to the sequence of price relatives, yields a performance that is as large as this weighted average. Hence, we use the context-tree concept to construct this algorithm, however, not to perform any weighted averaging. The key difference and the main problem that is solved, unlike [14], is to construct this algorithm using the tools of context-trees for convex-combinations under log loss.

Furthermore, although the application of such models and the context-tree weighting algorithm to universal prediction appeared in [13] for the piecewise linear prediction of bounded arbitrary sequences under the square error loss, there are important differences. While the problem of universal portfolio selection considered here can be viewed as a sequential decision problem with a restricted form of the log loss, the results in [13] for square error are incompatible with the portfolio context. We note that the log loss function considered in here is not bounded and the regret defined in [13] is with respect to a loss function which is exp-concave and bounded. These conditions must hold for the scheme in [13] to hold. Hence, the algorithmic steps as well as the proofs of the performance for the algorithms are different. Furthermore, intrinsic to portfolio selection, here, we also consider the case when there are transaction costs present and provide an algorithm using context-tree weighting that performs as well as the best context dependent algorithm under transaction costs.

We begin our discussion of piecewise constant rebalanced portfolios with the case when the partition is fixed and known in Section 2.1. We then extend these results using context-trees in Section 2.2 to include comparison classes with arbitrary partitions from a doubly exponential class of possible partitions. In each section, we provide theorems that upper-bound the regret with respect to the best competing algorithm in the class. The theorems are constructive, in that they yield algorithms satisfying the corresponding bounds. An explicit implementation of the context-tree PCRP algorithm is also given. Extension to investment under transaction costs is given in Section 2.3. The paper is then concluded with simulations of the algorithms on historical data.

2. Piecewise constant rebalanced portfolios

2.1. Fixed partition

In this section, we investigate the framework when the partition of the space of past observation vectors is given, i.e., say $\cup_{k=1}^K V_k = \mathbb{R}_+^m$ is known. Since the partition is fixed, the side-information generating mechanism, i.e., assigning CRPs to each region, is known. In this case, we seek to find a sequential portfolio such that when applied to any $\{\boldsymbol{x}[t]\}_{t\geq 1}$, asymptotically achieves, for all n,

$$\sup_{\boldsymbol{b}_1\in\mathcal{B},\ldots,\boldsymbol{b}_K\in\mathcal{B}} \prod_{t=1}^{n} \boldsymbol{b}_{s[t-1]}^T \boldsymbol{x}[t], \tag{1}$$

where $s[t-1]=k$ when $\boldsymbol{x}[t-1]\in V_k$ and $\mathcal{B}$ is the simplex. That is, we wish to obtain a sequential portfolio that achieves a wealth over every sequence of observations $\{\boldsymbol{x}[t]\}_{t\geq 1}$ as large as the best fixed PCRPs for that sequence with a partition of the observation space given by $\cup_{k=1}^K V_k = \mathbb{R}_+^m$. The algorithm introduced here will be sequential such that it will only depend on $\boldsymbol{x}[1],\ldots,\boldsymbol{x}[t-1]$. In this case, we have the following result:

Theorem 1. *We can construct a sequential algorithm $\tilde{\boldsymbol{b}}[t]$ such that when applied to any arbitrary and unknown sequence of observation vectors $\{\boldsymbol{x}[t]\}_{t\geq 1}$ (which are revealed sequentially) such that $\boldsymbol{x}[t]\in\mathbb{R}_+^m$ for all t and for which some components of $\boldsymbol{x}[t]$ can be zero, for all n, satisfies*

$$\ln\prod_{t=1}^{n}\tilde{\boldsymbol{b}}^T[t]\boldsymbol{x}[t] \geq \ln\prod_{t=1}^{n}\boldsymbol{b}_{s[t-1]}^T\boldsymbol{x}[t] - \frac{K(m-1)}{2}\ln(n/K+1) - K\ln(2), \tag{2}$$

where $s[t-1]$ is the indicator variable, i.e., $s[t-1]=k$ when $\boldsymbol{x}[t-1]\in V_k$ and $\boldsymbol{b}_k\in\mathcal{B}$, $k=1,\ldots,K$, are arbitrary CRPs assigned to regions.

Note that the bound in (2) holds for all $\boldsymbol{b}_k\in\mathcal{B}$, $k=1,\ldots,K$, i.e., even for the optimal CRPs for each region for any n.

The Proof of Theorem 1 and construction of the sequential algorithm is given in Appendix A.

2.2. Representing the piecewise regions via context-trees

In this section, the sequential algorithm introduced in here competes against not only with the best CRPs for a given partition, but also against the best partition of the space of past observations defined on a context-tree. To accomplish this, we define a depth-D context-tree with 2^D leaves, as shown in Fig. 2, where, for this tree, D=2. For a depth-D context-tree, the 2^D finest region bins correspond to leaves of the tree. As an example, on this tree, each of the leaves are assigned to regions: $\{1 \geq x_2 \geq x_1 \geq 0\}$, $\{x_2 \geq x_1 > 1\}$, $\{1 \geq x_1 > x_2 \geq 0\}$ and $\{x_1 > x_2 > 1\}$. Of course, more general partitioning schemes could be represented by such a context-tree, i.e., the leaves can be assigned arbitrarily. For a tree of depth-D, there exist $2^{D+1}-1$ nodes such that 2^D of these nodes are leaf nodes and remaining nodes are internal nodes. On this tree, each node ρ, $\rho = 1,\ldots,2^{D+1}-1$, represents a portion (or a volume) of the positive quadrant $\mathbb{R}_+^m$, V_ρ. The leaves are assigned to the finest partition bins. The region corresponding to an inner node, e.g., say ρ, V_ρ, which is not a leaf node, is constructed by the union of regions represented by the nodes of its children. For a binary tree, each inner node has two children, left child node ρ_L with V_{ρ_L} and right child node ρ_R with V_{ρ_R} such that $V_\rho = V_{\rho_L} \cup V_{\rho_R}$. One can define a doubly exponential number, $\mathrm{num}(D) \approx (1.5)^{2^D}$ of complete subtrees on this context-tree. As an example, in Fig. 2 given a depth-2 context-tree, we present num(2)=5 different complete subtrees. In the context of this paper, a subtree is complete, if the union of the regions assigned to its leaves (which are the leaves or the inner nodes of the context-tree) gives the whole positive orthant. Hence, each subtree corresponds to a complete and disjoint "partition" of the positive orthant $\mathbb{R}_+^m$. Specifically, for each subtree i=1,…,num(D), we define the partition $\mathcal{P}_i = \{V_{i,1},\ldots,V_{i,K_i}\}$ with $\cup_{k=1}^{K_i} V_{i,k} = \mathbb{R}_+^m$, where each $V_{i,k}$ is assigned to a leaf of the corresponding subtree, which is naturally a node (or a leaf) of the context-tree. For example in Fig. 2, for $\mathcal{P}_1$, we have $V_{1,1} = V_1$, $V_{1,2} = V_2$ and $V_{1,3} = V_3 \cup V_4$.

For each such partition $\mathcal{P}_i$ defined on this context-tree, a sequential algorithm, $\hat{\boldsymbol{b}}_{\mathcal{P}_i}[t]$, can be assigned such as the sequential algorithm from Section 2.1 which achieves the performance of the best PCRP for that partition. To achieve the performance of the sequential algorithm with the best partition defined on this context-tree, we will first demonstrate a sequential portfolio, $\tilde{\boldsymbol{b}}_u[t]$, such that when applied to any $\{\boldsymbol{x}[t]\}_{t\geq 1}$, for all n, asymptotically achieves

$$\sup_{\mathcal{P}_i} \prod_{t=1}^{n} \hat{\boldsymbol{b}}_{\mathcal{P}_i}^T[t]\boldsymbol{x}[t],$$

i = 1,…,num(D). Hence, the sequential portfolio $\tilde{\boldsymbol{b}}_u[t]$ uniformly achieves the performance of the sequential portfolio with the best partitioning among the class of a doubly exponential number of portfolios represented on the context-tree. Furthermore, if the sequential algorithms $\hat{\boldsymbol{b}}_{\mathcal{P}_i}[t]$ assigned to each partition are selected such that they achieve the performance of the best PCRP as in Section 2.1 for that partition, then $\tilde{\boldsymbol{b}}_u[t]$ will achieve the performance of the best PCRP with the corresponding best partition. In this sense, the algorithms introduced here are "twice-universal" in the nomenclature of universal source coding [26], i.e., universal with respect to the partitions and the parameters, and are based on sequential probability assignment [14,27,23].

Using this context-tree, we can construct a sequential algorithm with complexity only linear in the depth of the context-tree per investment period satisfying:

Theorem 2. *Given a context-tree with corresponding nodes ρ, $\rho = \{1,\ldots,2^{D+1}-1\}$ and arbitrary sequential portfolios assigned to each node $\hat{\boldsymbol{b}}_\rho[t]$, suppose $\hat{\boldsymbol{b}}_{\mathcal{P}_i}[t]$ represents the sequential portfolio obtained by the combination of the sequential portfolios corresponding to its piecewise regions $\mathcal{P}_i = \{V_{i,1},\ldots,V_{i,k}\}$, i.e., $\hat{\boldsymbol{b}}_{\mathcal{P}_i}[t] = \hat{\boldsymbol{b}}_\rho[t]$ if $\boldsymbol{x}[t-1] \in V_{i,k}$ and $V_{i,k} = V_\rho$. Then, we can construct a sequential portfolio $\tilde{\boldsymbol{b}}_u[t]$ with complexity linear in the depth of the context-tree per investment period such that when applied to any arbitrary and unknown sequence of observation vectors $\{\boldsymbol{x}[t]\}_{t\geq 1}$ (such that $\boldsymbol{x}[t] \in \mathbb{R}_+^m$ for all t and for which some components of $\boldsymbol{x}[t]$ can be zero and $\boldsymbol{x}[t]$ are revealed sequentially) satisfies, for all n,*

$$\ln \prod_{t=1}^{n} \tilde{\boldsymbol{b}}_u^T[t]\boldsymbol{x}[t] \geq \sup_{\mathcal{P}_i}\left(\ln \prod_{t=1}^{n} \hat{\boldsymbol{b}}_{\mathcal{P}_i}^T[t]\boldsymbol{x}[t] - (2K_i-1)\ln(2)\right),$$

where i=1,…,num(D).

The proof of Theorem 2 and construction of the sequential algorithm of Theorem 2 is given in Appendix B. We note that although there are a total of $2^{D+1}-1$ algorithms present at the nodes, each update of the context-tree has a computational complexity only linear in D. The construction of the universal portfolio, $\tilde{\boldsymbol{b}}_u[t]$, is given at the end of the proof of Theorem 2, where a complete algorithmic description as well as a pseudo-code is also given. This theorem implies that, without *a prior* knowledge of any complexity constraint on the algorithm, such as prior knowledge of the depth of the context-tree against which it is competing, the sequential portfolio can compete well with each and every subpartition within the depth-D full tree used in its construction.

We emphasize that there are no restrictions on the sequential portfolios running independently within each region, i.e., $\hat{\boldsymbol{b}}_\rho[t]$. Hence, in order to achieve the performance of the best PCRP among the class of all PCRPs represented on this context-tree, one can use specific portfolios on each node that achieve the performance of the best CRP for that regions. As an example, if we choose $\hat{\boldsymbol{b}}_\rho[t]$ as Cover's algorithm [11] running independently in each region, then we have the following result for this specific algorithm:

Theorem 3. *We can construct a sequential portfolio $\tilde{b}_u[t]$ with complexity linear in the depth of the context-tree per investment period such that when applied to arbitrary and unknown $\{\boldsymbol{x}[t]\}_{t\geq 1}$ (where $\boldsymbol{x}[t] \in \mathbb{R}_+^m$ for all t and for which some components of $\boldsymbol{x}[t]$ can be zero and $\boldsymbol{x}[t]$ is revealed sequentially) for all n satisfies*

$$\ln \prod_{t=1}^{n} \tilde{\boldsymbol{b}}_u^T[t]\boldsymbol{x}[t] \geq \sup_{\mathcal{P}_i}\left(\sup_{\boldsymbol{b}_{i,k}\in\mathcal{B}} \ln \prod_{t=1}^{n} \boldsymbol{b}_{i,s_i[t-1]}^T\boldsymbol{x}[t] - (2K_i-1)\ln(2) - \frac{K_i(m-1)}{2}\ln\left(\frac{n}{K_i}+1\right) - K_i\ln(2)\right), \quad (3)$$

where $i=1,\ldots,\mathrm{num}(D)$, $k=1,\ldots,K_i$, $s_i[t-1]$ *is the state indicator variable for partition* $\mathcal{P}_i$, *i.e.*, $s_i[t-1]=k$ *if* $\boldsymbol{x}[t-1] \in V_{i,k}$ *and* $\boldsymbol{b}_{i,k} \in \mathcal{B}$ *are arbitrary CRPs for each node.*

The outline of the proof of Theorem 3 is given in Appendix C. Note that the algorithm of Theorem 3 competes against a competition class that has a continuum of experts, i.e., a doubly exponential number of experts represented by the context-tree and a continuum of CRPs in each region. We point out that in the construction of the algorithm of Theorem 3, we use Cover's algorithm in each node. However, if we use alternative algorithms instead of Cover's algorithm, such as the ones in [3,22], then the computational complexity of the final algorithm would be reduced, however, the performance bound in (3) would degrade.

2.3. Context-trees with transaction cost (or commission)

We now consider the extension of previous algorithms and theorems to include the presence of commissions. Here, an investor pays a fixed percentage commission for his transactions, particularly $0 \le c_{\text{sell}} \le 1$ for selling and $0 \le c_{\text{buy}} \le 1$ for buying, where we assumed that these rates are the same for all stocks [28]. As an example, to buy A amount of stock 1 and to sell B amount of stock 2, an investor should pay a total amount of $Ac_{\text{buy}}+Bc_{\text{sell}}$ in transaction costs. Clearly, keeping a CRP requires potentially significant trading. If one starts with a capital of 1 dollar and invests with a constant rebalanced portfolio $\boldsymbol{b}=[b_1,\ldots,b_m]^T$, then at the end of the first period, one has $b_i x_i$ dollars in each stock $i=1,\ldots,m$, where x_i is the relative price change of the ith stock. Now, the new portfolio vector is given by $[b_1 x_1/\sum_i (b_i x_i),\ldots,b_m x_m/\sum_i (b_i x_i)]^T$ (which can be significantly different than $\boldsymbol{b}$) and must be adjusted to $\boldsymbol{b}$ before the next trading. An extensive study of how this trading could be done to minimize the wealth loss due to commission is covered in [17]. Our results are unaffected by how this trading is done.

We first investigate the framework considered in Theorem 2, where given a sequence of observation vectors $\boldsymbol{x}^n$, we have sequential portfolios, $\hat{\boldsymbol{b}}_\rho[t]$, for each node ρ on the context-tree. Here, $\hat{\boldsymbol{b}}_{\mathcal{P}_i}[t]$ is the sequential portfolio obtained by the combination of the sequential portfolios corresponding to its regions $\mathcal{P}_i=\{V_{i,1},\ldots,V_{i,K_i}\}$, i.e., $\hat{\boldsymbol{b}}_{\mathcal{P}_i}[t]=\hat{\boldsymbol{b}}_\rho[t]$ if $\boldsymbol{x}[t-1] \in V_{i,k}$ and $V_{i,k}=V_\rho$. If from one trading day to the next, the state does not change, i.e., $\boldsymbol{x}[t-1] \in V_{i,k}$ and $\boldsymbol{x}[t] \in V_{i,k}$, then the algorithm would pay a transaction cost of rebalancing the portfolio to $\hat{\boldsymbol{b}}_\rho[t]$ at time t if node ρ corresponds to $V_{i,k}$. However, if the state changes at time t, i.e., $\boldsymbol{x}[t-1] \in V_{i,k}$, $\boldsymbol{x}[t] \in V_{i,l}$ and $k \ne l$, then the algorithm pays a transaction cost to rebalance the portfolio to $\hat{\boldsymbol{b}}_\kappa[t]$ if node κ corresponds to $V_{i,l}$. We define the wealth achieved by this algorithm as $W^c(\boldsymbol{x}^n|\hat{\boldsymbol{b}}_{\mathcal{P}_i},\mathcal{P}_i)$, including costs c_{sell} and c_{buy}, where $c=c_{\text{sell}}+c_{\text{buy}}$. The wealth $W^c(\boldsymbol{x}^n|\hat{\boldsymbol{b}}_{\mathcal{P}_i},\mathcal{P}_i)$ can be significantly less than $W(\boldsymbol{x}^n|\hat{\boldsymbol{b}}_{\mathcal{P}_i},\mathcal{P}_i)$ if c is large. We demonstrate that:

Theorem 4. *Given a context-tree of depth-D with corresponding nodes* ρ, $\rho=\{1,\ldots,2^{D+1}-1\}$ *and sequential portfolios for each node* $\hat{\boldsymbol{b}}_\rho[t]$, *we can construct sequential portfolios* $\tilde{\boldsymbol{b}}^c_u[t]$ *with complexity linear in the depth of the context-tree per investment period such that for any* $c=c_{\text{sell}}+c_{\text{buy}}$, *and for all n, when applied to any arbitrary and unknown* $\{\boldsymbol{x}[t]\}_{t\ge 1}$ (*where* $\boldsymbol{x}[t] \in \mathbb{R}^m_+$ *for all t and some components of* $\boldsymbol{x}[t]$ *can be zero and* $\boldsymbol{x}[t]$ *is revealed sequentially*) *satisfies*

$$\ln \tilde{W}^c_u(\boldsymbol{x}^n) \ge \sup_{\mathcal{P}_i} (\ln W^c(\boldsymbol{x}^n|\hat{\boldsymbol{b}}_{\mathcal{P}_i},\mathcal{P}_i)-(2K_i-1)\ln(2)),$$

where $W^c_u(\boldsymbol{x}^n)$ *is the wealth achieved by the universal algorithm with commissions.*

The outline of the proof of Theorem 4 is given in Appendix D. We next consider the presence of commission with CRPs in each region, $\boldsymbol{B}_i=[\boldsymbol{b}_{i,1},\ldots,\boldsymbol{b}_{i,K_i}]$ where each $\boldsymbol{b}_{i,k} \in \mathcal{B}$. We define the wealth achieved by this algorithm as $W^c(\boldsymbol{x}^n|\boldsymbol{B}_i,\mathcal{P}_i)$, including costs c_{sell} and c_{buy}, where $c=c_{\text{sell}}+c_{\text{buy}}$ and the CRPs are given by $\boldsymbol{B}_i=[\boldsymbol{b}_{i,1},\ldots,\boldsymbol{b}_{i,K_i}]$. For this framework, have the following result:

Theorem 5. *We can construct sequential portfolios* $\tilde{\boldsymbol{b}}^c_u[t]$ *with complexity linear in the depth of the context-tree per investment period such that for any* $c=c_{\text{sell}}+c_{\text{buy}}$, *and for all n, when applied to* $\{\boldsymbol{x}[t]\}_{t\ge 1}$ (*where* $\boldsymbol{x}[t] \in \mathbb{R}^m_+$ *for all t and some components of* $\boldsymbol{x}[t]$ *can be zero and* $\boldsymbol{x}[t]$ *is revealed sequentially*) *satisfies*

$$\ln \tilde{W}^c_u(\boldsymbol{x}^n) \ge \sup_{\mathcal{P}_i} \Bigg(\ln W^c(\boldsymbol{x}^n|\boldsymbol{B}_i,\mathcal{P}_i)-(2K_i-1)\ln(2) -K_i(m-1)\ln\left(\frac{n(1+c)}{K_i}+1\right)-K_i\ln(2)-N_i\ln\left(\frac{1}{1-c}\right)\Bigg),$$

where $W^c_u(\boldsymbol{x}^n)$ *is the wealth achieved by the universal algorithm with commission and* N_i *is the total number of state changes taken by* $\mathcal{P}_i$ *on* $\boldsymbol{x}^n$.

The outline of the proof of Theorem 5 is given in Appendix D.

2.4. Algorithmic description

In this section, we present the algorithmic description of the context-tree algorithm introduced in Theorem 2. We will also point out the variables that need to be changed to get the universal algorithms introduced in Theorems 3–5. A complete description of the universal context-tree algorithm is given in Fig. 3.

A depth-D context-tree has $2^{D+1}-1$ nodes and for each node $\rho=1,\ldots,2^{D+1}-1$ we assign a corresponding sequential portfolio $\boldsymbol{B}_\rho[n-1]=\tilde{\boldsymbol{b}}_\rho[n-1]$. This portfolio vector can be selected as Cover's sequential portfolio as in (15), however, we define it as a generic vector since any sequential portfolio from [11,3,15] can be used instead. Moreover, selecting these sequential vectors as the portfolios introduced in [17] yields the universal context-tree algorithm under transaction costs introduced in Theorems 4 and 5.

For each node ρ, we also assign two variables: the "assigned" total wealth $T_\rho[n-1] \triangleq \tilde{W}_\rho(\boldsymbol{x}^{n-1})$ and the achieved wealth by the corresponding sequential portfolio of that node $A_\rho[n-1] \triangleq \prod_{t=1}^{n_\rho-1} \tilde{\boldsymbol{b}}^T_\rho[t]\boldsymbol{d}_\rho[t]$. For a full tree of depth D, we need to store a total of $(m+2)(2^{D+1}-1)$ variables. At each time $n-1$, only $D+1$ of these portfolios or $m(D+1)$ variables will be used or updated.

Algorithm:

for $\rho = 1 : 2^{D+1} - 1$
  $\boldsymbol{B}_{\rho}[0] = \mathbf{1}/m$
  $\boldsymbol{A}_{\rho}[0] = 1$
for $n = 1, \ldots, N$,
  % find affected nodes in $O(D)$
  $\boldsymbol{a} = [\,]$;
  for $\rho = 1, \ldots, 2^{D+1} - 1$,
    if $\boldsymbol{x}[n-1] \in V_{\rho}$,
      $\boldsymbol{a} = [\boldsymbol{a}; \rho]$;
  % find weight for each node:
  $\beta_0[n-1] = \frac{1}{2}$;
  for $\rho = \boldsymbol{a}[2], \ldots, \boldsymbol{a}[D+1]$,
    $\beta_k[n-1] = \frac{1}{2} T_{\boldsymbol{a}[k],\mathrm{sib}}[n-1]\beta_{k-1}[n-1]$;
    $\alpha_k[n-1] = \frac{\beta_k[n-1]A_{\boldsymbol{a}[k]}[n-1]}{T_{\boldsymbol{a}[1]}[n-1]}$;
  % portfolio in $O(D)$ operations:
  $\tilde{\boldsymbol{b}}_u[n] = \sum_{k=0}^{K} \alpha_k[n-1]\boldsymbol{B}_{\boldsymbol{a}[k]}[n-1]$;
  % update node probabilities in $O(D)$ operations
  for $k = D+1, \ldots, 1$,
    $A_{\boldsymbol{a}[k]}[n] = A_{\boldsymbol{a}[k]}[n-1]\boldsymbol{B}^T_{\boldsymbol{a}[k]}[n-1]\boldsymbol{x}[n]$;
    if $k = D+1$ (leaf node), $T_{\boldsymbol{a}[k]}[n] = A_{\boldsymbol{a}[k]}[n]$;
    elseif $k \neq D+1$, $T_{\boldsymbol{a}[k]}[n] = \frac{1}{2} T_{\boldsymbol{a}[k]_L}[n-1]T_{\boldsymbol{a}[k]_R}[n-1] + \frac{1}{2}A_{\boldsymbol{a}[k]}[n]$;
    $\boldsymbol{B}_{\boldsymbol{a}[k]}[n] = \tilde{\boldsymbol{b}}_{\boldsymbol{a}[k]}[n]$;

Fig. 3. An algorithmic description of the context-tree portfolio algorithm.

Initially for all nodes $\rho = 1, \ldots, 2^{D+1}-1$, all the assigned and achieved wealth for each node should be 1, i.e., $T_{\rho}[0] = 1, A_{\rho}[0] = 1$ and all initial portfolios should be $\boldsymbol{B}_{\rho}[0] = [1/m, \ldots, 1/m]^T$ since, when there is no data, we start with an equal investment in each stock. At each time $n-1$, we first determine the affected nodes, i.e., the nodes ρ_k such that $\boldsymbol{x}[n-1] \in V_{\rho_k}$. Although in Fig. 3, we loop through all the nodes in the third step of the algorithmic description to find the affected nodes, this calculation is really $O(D)$ since we need to check only the nodes starting from an affected leaf until the root node of the tree. These affected nodes are stored in vector $\boldsymbol{a}$. On the figure, $\boldsymbol{a}[k]$ represents the kth component of the vector $\boldsymbol{a}$. For these nodes, we calculate $\beta_k[n-1]$ which are in turn to be used to calculate $\alpha_k[n-1]$ and final portfolio, after $O(D)$ operations. Here, each $\beta_k[n-1]$ is recursively generated by the product of the assigned wealth of the corresponding sibling nodes and $\beta_{k-1}[n-1]$. For the update, only the variables and the portfolios of the selected nodes ($D+1$ of them) are updated using the new observation vector $\boldsymbol{x}[n]$. Hence, we efficiently combine num(D) sequential portfolios only using $D+1$ node portfolios and $O(D+1)$ operations for combination of the portfolios, per investment period.

3. Simulations

In this section, we illustrate the performance of our algorithms on historical data sets collected from the New York Stock exchange over a 22-year period until 1985[1] [29]. In the initial set of experiments, we demonstrate performance of our algorithms and illustrate effects of the internal parameters within the algorithms on the final performance using the stock pair Kinark–Iroquois, which are chosen because of their volatility. For the initial set of experiments, there are no transaction costs involved. We then present results over larger and more comprehensive data sets with and without transaction costs. Here, we first use a context-tree of depth D=2 with sequential portfolios given in (15) from [11] at each node. We implement two different context-tree algorithms corresponding to two different partitions of the space of past observations, i.e., the positive orthant $\mathbb{R}^m_+$. The first partition corresponds to the partition given in Fig. 1, referred here as “diagonal partitioning”, and the second partition corresponds to the partitioning $\mathbb{R}^m_+$ into rectangle regions, i.e., each leaf corresponds to one of the rectangular regions: ($x_1 \leq 1$ and $x_2 \leq 1$), ($x_1 > 1$ and $x_2 \leq 1$), ($x_1 \leq 1$ and $x_2 > 1$) and ($x_1 > 1$ and $x_2 > 1$), referred here as “rectangular partitioning”. For rectangular partitioning, to get a binary tree of depth more than D=2, we continue to perform splittings further from each leaf, e.g., for the leaf with $x_1 \leq 1$, we split further whether $x_1 \leq \frac{1}{2}$ or $\frac{1}{2} < x_1$. Naturally, this splitting corresponds to splitting a high dimensional cube into hyper-rectangular regions. Although, other methods exist to partition the space of past price relative vectors, we observe that these two partitions have produced good results and are straightforward to implement. In Fig. 4, we plot the wealth achieved by the context-tree algorithm with diagonal partitioning (CTW-CRP diagonal partitioning), context-tree algorithm with rectangular partitioning (CTW-CRP rectangular passioning), Cover's universal portfolio [11] (Cover's portfolio) and the best CRP tuned to the sequence of observations (best CRP). We next compare the performance of the context-tree algorithm to the

[1] We thank Dr. Erik Ordentlich for providing us with the historical data.

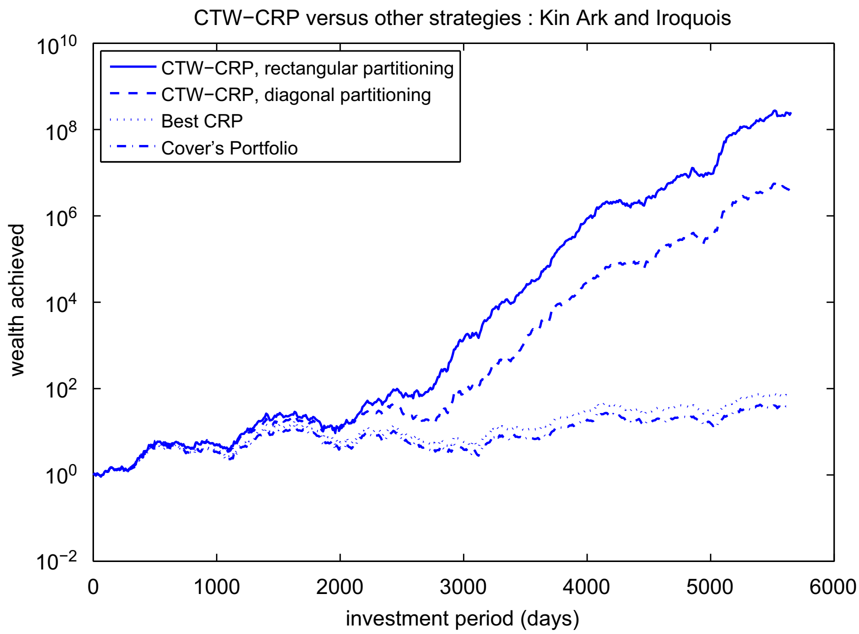

Fig. 4. Wealth achieved on the Iroquois–Kinark stock pair by: the context-tree universal portfolios using portfolio from (15) in each node, (CTW-CRP), with $D=2$ for two different partitioning (diagonal partitioning and rectangular partitioning), Cover's universal portfolio [11] and the best CRP.

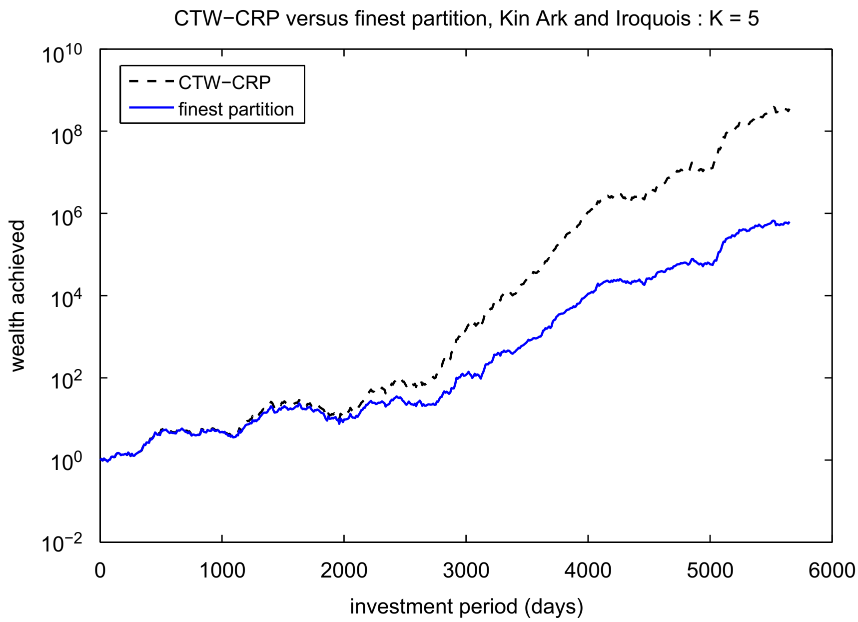

Fig. 5. Wealth achieved on the Iroquois–Kinark stock pair by: the context-tree universal portfolios using portfolio from (15) in each node (CTW-CRP) with $D=5$ and rectangular partitioning, the sequential portfolio corresponding to the finest partition.

performance of the sequential portfolio corresponding to the finest partition on the context-tree, i.e., the sequential portfolio corresponding to the partition with 2^D leaves, in Fig. 5. Here, we use a depth $D=5$ context-tree with a rectangular partition of $\mathbb{R}_+^m$. Since the context-tree algorithm adaptively combines portfolios of several different partitions, it is able to favor either the coarser models with small numbers of parameters (that may have better performance when the data length is small) or the finest model with larger numbers of parameters (that may have better modeling power) depending on the respective performance. We also plot the wealth achieved by the context-tree algorithms on the Kinark–Iroquois pair for different context-tree depths including $D=1,2,3,4,5$ in Fig. 6

with rectangular partitioning to demonstrate the effect of the tree depth on the final wealth. We observe that for this stock pair the depth of the context-tree does not effect performance significantly.

As pointed out in Theorem 3 and in the proof of Theorem 2, the implementation of the context-tree algorithm is generic such that a wide variety of sequential portfolios can be used in each node such as those from [3,22]. Accordingly, in the next set of experiments, we use the sequential portfolios from [3,22] instead of the sequential portfolio from [11] in each node. These implementations have significantly less computational complexity, however, they also have inferior guaranteed performance bounds. For the algorithm from [22], certain

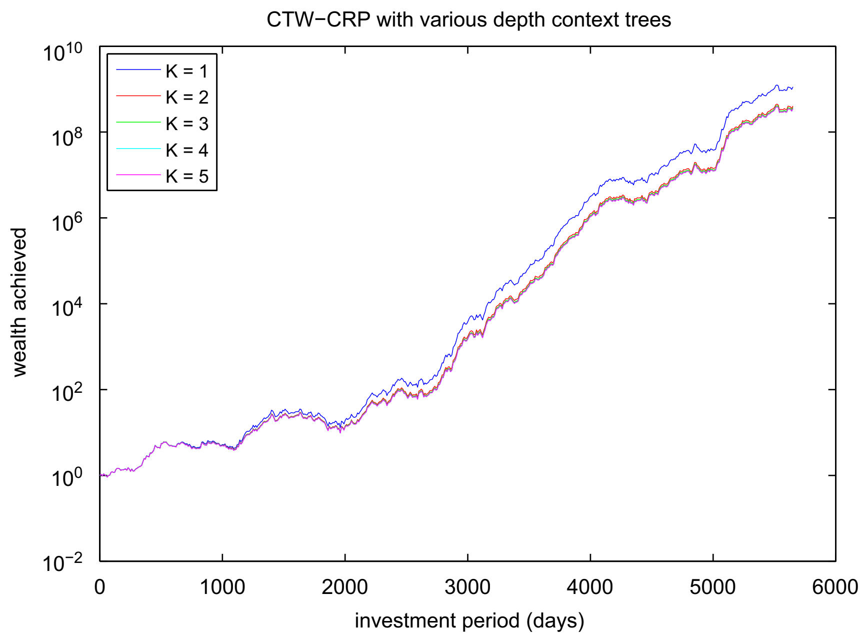

Fig. 6. Wealth achieved on the Iroquois–Kinark stock pair by different depth context-tree universal portfolios using portfolio from (15) in each node (CTW-CRP) and rectangular partitioning, including D=1,2,3,4,5.

a

	m = 2	m = 3
CTW-CRP	9.4505e+005	2.7090e+005
CTW-CRP-Finest	231.2840	35.0917
CTW-CRP-Best	6.5936e+006	1.8456e+006
CTW-ONS	2.2194e+008	5.4999e+008
CTW-ONS-Finest	1.1853e+008	1.5134e+007
CTW-ONS-Best	3.8808e+009	1.3016e+011
CTW-MU	2.1877e+008	5.0219e+008
CTW-MU-Finest	4.8769e+007	3.2106e+006
CTW-MU-Best	3.8808e+009	3.3074e+010
COVERS	18.5884	21.2522
MU1	21.4041	24.3892
MU2	12.6614	12.1242
ONS1	27.5655	32.5797
ONS2	12.3978	11.7638

b

	m = 2	m = 3
CTW-CRP	22.6628	24.2139
CTW-CRP-Finest	20.1467	21.8079
CTW-CRP-Best	40.3668	38.0346
CTW-ONS	100.6965	161.2965
CTW-ONS-Finest	55.9575	90.8693
CTW-ONS-Best	1.1023e+004	7.4008e+004
CTW-MU	87.4047	132.6730
CTW-MU-Finest	38.4194	43.1435
CTW-MU-Best	1.1815e+003	2.9898e+003
COVERS	17.4538	19.3128
MU1	19.0605	20.9273
MU2	14.0180	14.2314
ONS1	22.2631	27.8127
ONS2	13.6954	13.7735

Fig. 7. Performance of several different sequential portfolios on historical stock pairs: the context-tree algorithm using sequential portfolios from (15) in each node "CTW-CRP", from [22] in each node "CTW-ONS", from [3] "CTW-MU". The other sequential portfolios are described in the text with the selected algorithmic parameters. (a) Stock sets include Kinark and Iroquois stocks. (b) Stock sets do not include Kinark and Iroquois stocks.

algorithmic parameters which may depend on the investment horizon or underlying data should be set to achieve a logarithmic regret. However, these difficulties can be easily surpassed by applying this algorithm over exponentially increasing segment lengths and setting the parameters for each segment independently as in [3]. In Fig. 8, we present final wealths achieved after 22 years by several different investment strategies that are applied to stock pairs, i.e., $m=2$, that are randomly chosen among 35 stocks [3]. The presented results are the average wealths over 500 independent trials, i.e., stock pairs. For the results presented in Fig. 7a, the stock pairs include Kinark or Iroquois stocks. The results presented in Fig. 7b do not include all stocks including Kinark and Iroquois, which are shown to be highly "predictable" [30]. The investment strategies include, context-tree algorithms: using (15) in each node "CTW-CRP", using algorithm from [22] in each node "CTW-ONS", using algorithm from [3] in each node "CTW-MU". All context-trees have depth 8 and they use rectangular partitioning. For each context-tree algorithm, we further present the wealth achieved by the finest partition and the best partition on the tree. For CTW-ONS, the algorithmic parameters are selected in accordance with the Theorem 1 of [22], where $\eta=0$, $\beta=0.0004$ and $\delta=1$. For CTW-MU, the learning rate is selected as $\eta=10$. We further present wealth achieved by Cover's algorithm "COVERS", the investment algorithm from [3] "MU1" and from [22] "ONS1" with the learning rates given in the simulation sections of [3], i.e., $\eta=0.05$ and [22], i.e., $\eta=0$, $\beta=1$, $\delta=\frac{1}{8}$, respectively. To make a fair comparison with the CTW-ONS and the CTW-MU algorithms, we also present results for the algorithm from [22] with learning rates $\eta=0$, $\beta=0.0004$, $\delta=1$ "ONS2" and for the algorithm from [3] with $\eta=10$ "MU2". We observe that all context-tree algorithms achieve wealths that are larger than the wealths achieved by the algorithm corresponding to the finest partition on the tree, which could be the straightforward choice as the PCRP. We observe that although the algorithms are unable to achieve the performance of best partition on the context-tree, the results are close and the difference between the wealth of the context-tree algorithm and the best partition cannot

	m = 2	m = 3
CTW-CRP	22.4169	24.0879
CTW-CRP-Finest	19.7353	21.6492
CTW-CRP-Best	38.8162	37.3954
CTW-ONS	94.9536	149.1974
CTW-ONS-Finest	49.9478	100.5908
CTW-ONS-Best	1.3267e+004	7.1238e+004
CTW-MU	82.2727	125.4128
CTW-MU-Finest	29.3923	37.3555
CTW-MU-Best	987.9530	2.2498e+003
COVERS	17.4538	19.3128
MU1	19.0605	20.9273
MU2	14.0180	14.2314
ONS1	22.2631	27.8127
ONS2	13.6954	13.7735

Fig. 8. Wealth achieved over $m=2$ and 3, with history two, i.e., context-tree is used to partition $\{\boldsymbol{x}[t-2],\boldsymbol{x}[t-1]\}$. Results are averaged over 500 independent trials.

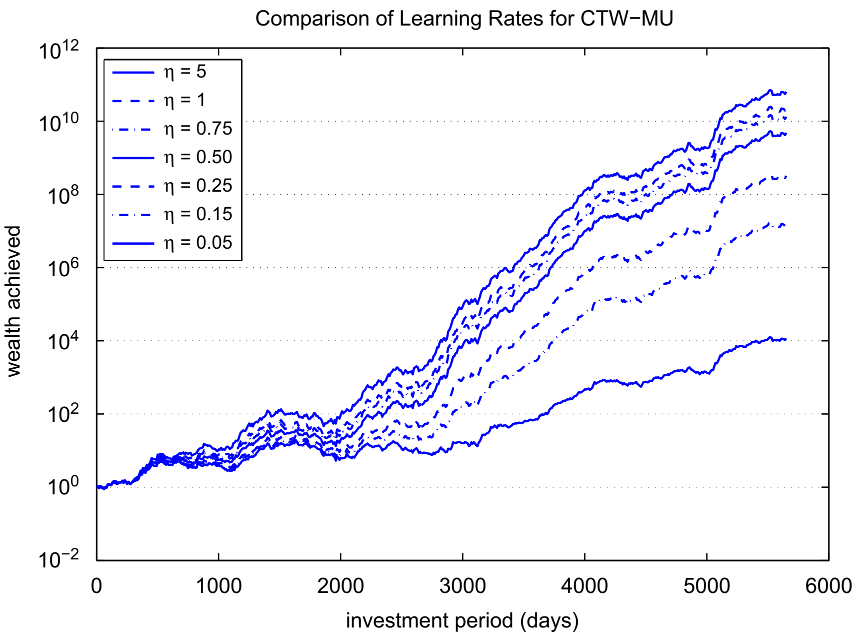

Fig. 9. CTW-MU algorithm with several different learning rates and $D=2$. The curves in the plot are in the same order with the captions on the plot.

be larger than the regret given in Theorem 2. In these simulations, we observe that the CTW-ONS algorithm achieves the largest wealth gain and all the context-tree algorithms achieve wealths that are larger than the wealths achieved by the other simulated algorithms. We observe that when the stock pairs are "predictable" [30], e.g., we point out the huge wealth difference when the stock sets include the Kinark–Iroquois pair or not, the context-tree algorithms achieve tremendous wealth gains. This demonstrates the significant modeling power of the PCRPs. As the next set of experiments, in Fig. 9, we present the results for the same algorithms when they are applied to stock sets of three, i.e., m=3, where three stocks are again selected randomly among 35 stocks independently in each trial. The presented results are averaged over 500 trials. We observe similar wealth gains as in the previous experiments. We note that most of the algorithms simulated achieve larger wealth gains when we increase the stock size.

To illustrate the effect of tree depth on the final wealth, in Fig. 10, we display wealth gain for the same algorithms when they are applied to random stock pairs with history of two, i.e., we partition the space $\{\boldsymbol{x}[t-2],\boldsymbol{x}[t-1]\}$, excluding Kinark and Iroquois stocks. We observe similar wealth gains when we increase the history to two, suggesting that the CTW algorithms are able to exploit enough dependency from the most recent past. To observe the effect of parameters selected for the introduced algorithms, as an example, we simulate the CTW-MU algorithm with different learning rates η in Fig. 11 when it is applied to Kinark–Iroquois pair. We observe that although the selection of the learning rate affects the performance, the relative effect is not significant for small changes in the parameter values.

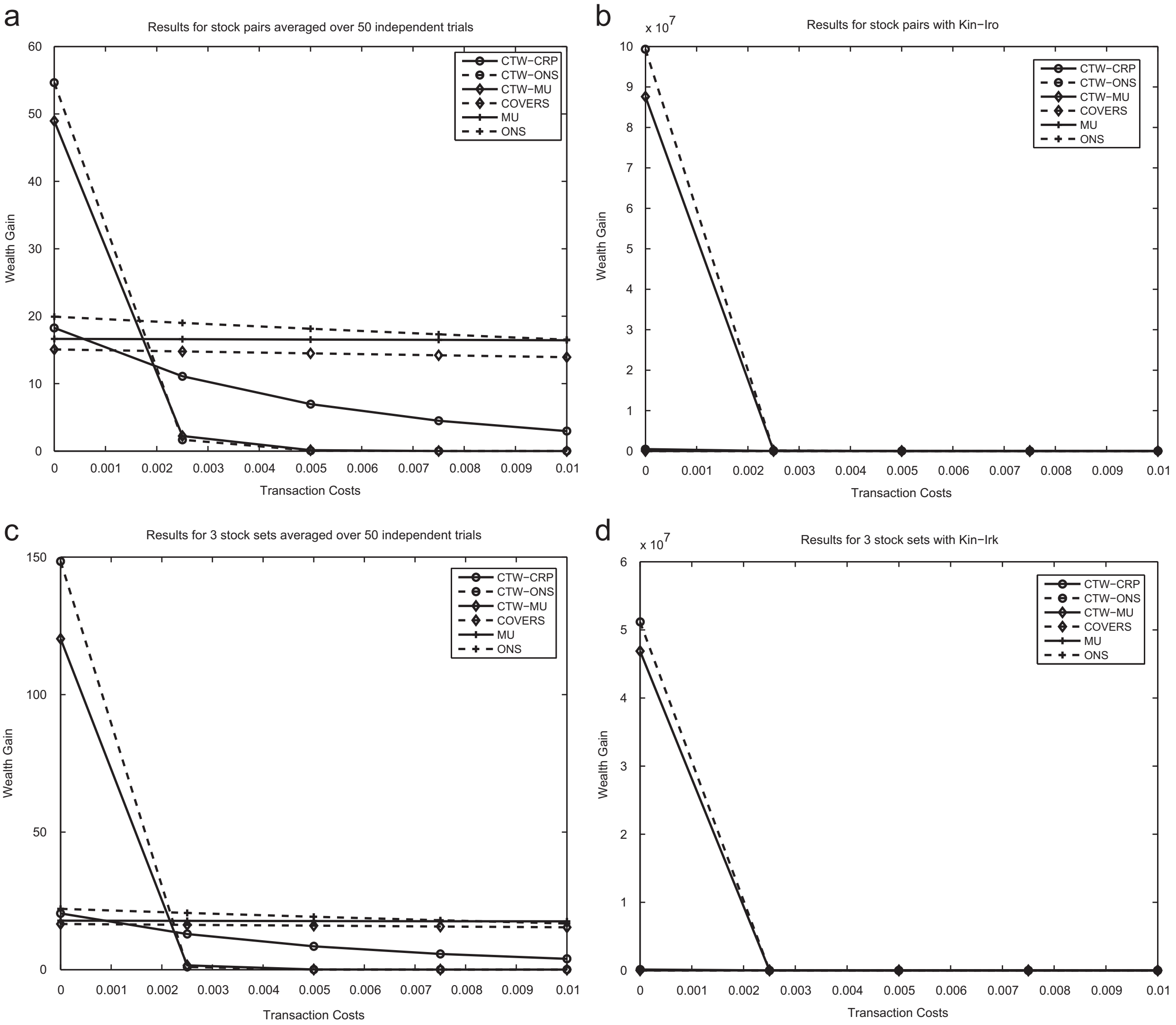

Fig. 10. Wealth achieved under different transaction costs ranging from c=0 to 0.01, where the results are averaged over 50 independent trials. (a) Two stocks pairs excluding Kinark and Iroquois stocks. (b) Two stock pairs including all stocks. (c) Sets of three stocks, excluding Kinark and Iroquois stocks. (d) Sets of three stocks including all stocks.

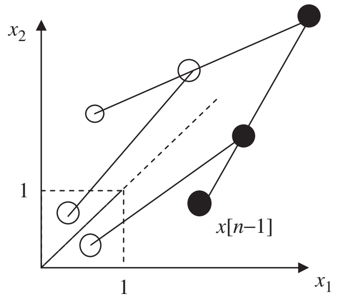

Fig. 11. A depth-2 full binary tree is shown with D+1 nodes to be updated.

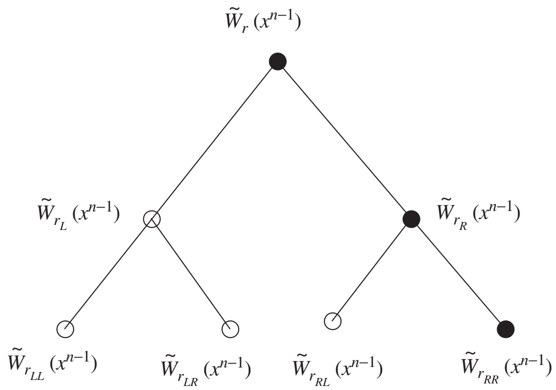

Fig. 12. Assigned wealth to each node at time $n-1$. Assigned wealth of each filled circle corresponding to the affected nodes are updated after observing $\boldsymbol{x}[n]$. For calculation of $\beta_k[n-1]$, we will use the sibling nodes of these affected nodes.

In the final set of experiments, we apply our algorithms to stock sets under several different transaction costs. For presentation purposes, we only display context-tree algorithms that give the largest wealth gains in Fig. 12. For transaction cost model, we adhere to the transaction strategy introduced in [17] and apply transaction cost only in buying stocks in accordance with [17]. The results are displayed for several different transactions costs ranging from c=0 up to c=0.01. In Fig. 12, we present the final wealths achieved after 22 years for stock pairs averaged over 50 independent trials. Here, the stock sets do not include Kinark and Iroquois stocks as in Fig. 10a and c, however, include these stocks as in Fig. 10b and d. We note that the context-tree based algorithms introduced in this paper can demand substantial amount of trading per investment period, which is also the case for most well-known investment algorithms [17]. However, we observe that for modest transaction costs, the tremendous gains achieved by the CTW based algorithms compensate much of the wealth reduction due to transaction costs. We note that for large volumes of stock operations and institutional investors, the transaction costs can be as low as 6 cents per stock unit. This amounts to a modest transaction cost ratio and may motivate such complex, but high yield algorithms. Furthermore, to avoid such excessive rebalancing, several methods [31,32] exist that rebalance to the suggested portfolio only if the difference between the present and the past portfolios exceeds a certain threshold. This threshold is then selected based on the volatility and amount of the transaction cost.

4. Conclusions

In this paper, we consider the problem of investing using PCRPs from a competitive algorithm perspective. Using context-trees and methods based on sequential probability assignment, we have shown a portfolio selection algorithm the logarithm of whose achieved wealth is within $O(\ln(n))$ of that of the best PCRP, which can only be selected using all of the data in hindsight. We use a method similar to context-tree weighting to compete well against a doubly exponential class of possible partitions of the space of observation vectors, for which we pay at most a "structural regret" proportional to the size of the best context-tree. For each partition, we use a universal portfolio to compete against the continuum of all possible CRPs. The results are then extended to the case when there are also transaction costs. The resulting algorithms are efficient, with time complexity only linear in the depth of the context-tree per combination and perform well for a variety of historical data. In our simulations, the context-tree investment strategy using the algorithm from [22] achieved the largest wealth.

Appendix A. The proof of Theorem 1 and construction of the sequential algorithm

When the partition is known and fixed, i.e., $\cup_{k=1}^{K} V_k = \mathbb{R}_+^m$, we have K independent CRP selection problems. If we define K time vectors (or index sequences) of length n_k, $t_k^{n_k} = \{t : s[t-1]=k, t=1,\ldots,n\}$, with k=1,…,K, and sequences $\boldsymbol{x}_k^{n_k} = \{\boldsymbol{x}[t_k[l]]\}_{l=1}^{n_k}$, then the universal sequential portfolio can be constructed using the universal portfolio of Cover's [11] independently in each region, i.e.,

$$\tilde{\boldsymbol{b}}[n] = \tilde{\boldsymbol{b}}_{s[n-1]}[n-1] \tag{4}$$

with Cover's portfolio [11]

$$\tilde{\boldsymbol{b}}_k[n] \triangleq \frac{\int_{\mathcal{B}} \boldsymbol{b} \prod_{l=1}^{n_k} \boldsymbol{b}^T \boldsymbol{x}[t_k[l]] \boldsymbol{\mu}(\boldsymbol{b})}{\int_{\mathcal{B}} \prod_{l=1}^{n_k} \boldsymbol{b}^T \boldsymbol{x}[t_k[l]] \boldsymbol{\mu}(\boldsymbol{b})}, \tag{5}$$

where n_k is the number of points of $\boldsymbol{x}^n$ that belong to V_k. This portfolio satisfies for any k and n_k [11]:

$$\ln \prod_{l=1}^{n_k} \tilde{\boldsymbol{b}}_k^T[l-1]\boldsymbol{x}_k[l] \geq \ln \prod_{l=1}^{n_k} \boldsymbol{b}_k^T \boldsymbol{x}_k[l] - \frac{(m-1)}{2}\ln(n_k+1) - \ln(2) \tag{6}$$

for any $\boldsymbol{b}_k$. Application of this performance bound into K separate regions and taking the maximum value for the right-hand side with respect to n_k values yield the corresponding result in Theorem 1, i.e., $\sum_{k=1}^{K} \ln(n_k+1) \leq \sum_{k=1}^{K} \ln(n/K+1)$ given $\sum_{k=1}^{K} n_k = n$ since $\ln(\cdot)$ is a concave function. This concludes the proof of Theorem 1. □

Appendix B. Proof of Theorem 2

Given a partition $\mathcal{P}_i=\cup_{k=1}^{K_i}V_{i,k}$ of the positive orthant $\mathbb{R}_+^m, \mathcal{P}_i \in \mathcal{P}$ (the competing class), we consider a family of portfolios, each with its own set of CRP vectors $\boldsymbol{B}_i=[\boldsymbol{b}_{i,1},\ldots,\boldsymbol{b}_{i,K_i}]$. Here, each $\boldsymbol{b}_{i,k}$ represents a CRP vector for the kth region of partition $\mathcal{P}_i$, i.e., when $\boldsymbol{x}[t-1]\in V_{i,k}$, we use $\boldsymbol{b}_{i,k}$. For each pairing of $\mathcal{P}_i$ and $\boldsymbol{B}_i$, we consider the wealth achieved by the corresponding algorithm

$$W(\boldsymbol{x}^n|\boldsymbol{B}_i,\mathcal{P}_i)\triangleq\prod_{t=1}^{n}\boldsymbol{b}_{i,s_i[t-1]}^T\boldsymbol{x}[t],$$

where $s_i[t-1]$ is the state indicator variable for partition $\mathcal{P}_i$, i.e., $s_i[t-1]=k$ if $\boldsymbol{x}[t-1]\in V_{i,k}$.

Given $\mathcal{P}_i$, the algorithm in the family with the best CRPs in each region achieves the largest wealth $W(\boldsymbol{x}^n|\boldsymbol{B}_i^*,\mathcal{P}_i)\triangleq\sup_{\boldsymbol{B}_i}W(\boldsymbol{x}^n|\boldsymbol{B}_i,\mathcal{P}_i)$. Maximizing $W(\boldsymbol{x}^n|\boldsymbol{B}_i^*,\mathcal{P}_i)$ over all $\mathcal{P}_i$ (on the tree) yields $W(\boldsymbol{x}^n|\boldsymbol{B}_i^*,\mathcal{P}_i^*)\triangleq\sup_{\mathcal{P}_i}W(\boldsymbol{x}^n|\boldsymbol{B}_i^*,\mathcal{P}_i)$. Here, $W(\boldsymbol{x}^n|\boldsymbol{B}_i^*,\mathcal{P}_i^*)$ corresponds to the best PCRPs in the class on the full tree of depth D. Note that without constraints computational complexity of the members of the competing class, since this performance is computed based on observations of the entire sequence in advance, $W(\boldsymbol{x}^n|\boldsymbol{B}_i^*,\mathcal{P}_i^*)$ corresponds to the algorithm operating on the finest partition of the space dictated by the leaves of the full binary tree, with the best CRP used within each region governed by each leaf in the tree. However, there is no guarantee that the performance of our sequential algorithm will perform the best if we choose this finest-grain model, owing to the dramatic increase in the number of parameters to be learned sequentially by the algorithm. It turns out that the finest grain model will generally not have the best performance when the algorithms within each node in the tree are required to sequentially compete with the best batch algorithm within each partition. As such, our goal is to perform well with respect to all possible partitions – which enables the algorithm to opportunistically make use of coarser models when the data is not sufficiently rich to support those of finest-grain, and to grow the contributions of finer grain models as the observed data can support it. As will be shown, the context-tree weighting approach enables the algorithm to achieve the performance of the best of any partition-based algorithm. Within each partition, the algorithm can sequentially achieve the performance of the best batch process. This property of "twice-universality," first over the class of partitions of the space of observations, and then again over the set of parameters within each partition, enables the algorithm to sequentially achieve the best possible performance out of the doubly exponential number, num(D), of partitions and the infinite set of parameters given the partition.

Given a context-tree with corresponding nodes ρ, $\rho=\{1,\ldots,2^{D+1}-1\}$ and arbitrary sequential portfolios assigned to each node $\hat{\boldsymbol{b}}_\rho[t]$, suppose $\hat{\boldsymbol{b}}_{\mathcal{P}_i}[t]$ represents the sequential portfolio obtained by the combination of the sequential portfolios corresponding to its piecewise regions $\mathcal{P}_i=\{V_{i,1},\ldots,V_{i,k}\}$, i.e., $\hat{\boldsymbol{b}}_{\mathcal{P}_i}[t]=\hat{\boldsymbol{b}}_\rho[t]$ if $\boldsymbol{x}[t-1]\in V_{i,k}$ and $V_{i,k}=V_\rho$. For any n, each sequential portfolio $\hat{\boldsymbol{b}}_{\mathcal{P}_i}[t]$ achieves the wealth

$$\tilde{W}(\boldsymbol{x}^n|\mathcal{P}_i)\triangleq\prod_{t=1}^{n}\hat{\boldsymbol{b}}_{\mathcal{P}_i}^T[t]\boldsymbol{x}[t]. \tag{7}$$

We observe that there exist num(D) such sequential algorithms corresponding to num(D) partitions. Suppose, we invest a portion of our initial wealth on each of these num(D) sequential algorithms and collect the final wealth at the end. We require that these initial investments should be recursively computable on the context-tree to compactly calculate num(D) sequential algorithms. One such assignment is given in [14] such that for each sequential portfolio corresponding to each partition $\mathcal{P}_i$, we invest $2^{-C(\mathcal{P}_i)}$ portion of our initial wealth. Here, $C(\mathcal{P}_i)=K_i+n_{\mathcal{P}_i}-1$ is defined as the "cost" of partition $\mathcal{P}_i$ and corresponds to number of bits required to represent the partition $\mathcal{P}_i$ on the tree [14], where $n_{\mathcal{P}_i}$ is the number of leaves of $\mathcal{P}_i$ that are not the leaves of the original tree. Note that $C(\mathcal{P}_i)\le 2K_i-1$, since the leaves of $\mathcal{P}_i$ that are not the leaves of the original tree are just the subset of the total leaves of $\mathcal{P}_i$, i.e., $n_{\mathcal{P}_i}\le K_i$. It can be shown that $\sum_{\mathcal{P}_i}2^{-C(\mathcal{P}_i)}=1$ [14] and this initial investment is recursively computable on the context-tree. We note that the values $C(\mathcal{P}_i)$ are only needed for proof purposes and are not present in the algorithmic description. Then, the final combined wealth at the end is given by

$$\tilde{W}_u(\boldsymbol{x}^n)\triangleq\sum_{\mathcal{P}_i}2^{-C(\mathcal{P}_i)}\tilde{W}(\boldsymbol{x}^n|\mathcal{P}_i). \tag{8}$$

By this initial wealth assignment, we will next demonstrate that num(D) such sequential algorithms can be compactly represented and calculated on the corresponding context-tree.

Clearly, for any $\mathcal{P}_i$, $\tilde{W}_u(\boldsymbol{x}^n)\ge 2^{-C(\mathcal{P}_i)}\tilde{W}(\boldsymbol{x}^n|\mathcal{P}_i)$, since $2^{-C(\mathcal{P}_i)}\ge 0$ and $\tilde{W}(\boldsymbol{x}^n|\mathcal{P}_i)\ge 0$, for all i. This yields

$$\ln(\tilde{W}_u(\boldsymbol{x}^n))\ge\ln(\tilde{W}(\boldsymbol{x}^n|\mathcal{P}_i))-C(\mathcal{P}_i)\ln(2), \tag{9}$$

$$\ge\ln(\tilde{W}(\boldsymbol{x}^n|\mathcal{P}_i))-(2K_i-1)\ln(2) \tag{10}$$

for any $\mathcal{P}_i$, where the second line follows since $C(\mathcal{P}_i)\le 2K_i-1$. We now have a total wealth $\tilde{W}_u(\boldsymbol{x}^n)$ whose logarithm is as large as the logarithm of the wealth achieved by the best partition. However, we have not yet identified a sequential portfolio that achieves $\tilde{W}_u(\boldsymbol{x}^n)$.

We now present a portfolio assignment algorithm achieving $\tilde{W}_u(\boldsymbol{x}^n)$. By definition

$$\tilde{W}_u(\boldsymbol{x}^n)=\prod_{t=1}^{n}\frac{\tilde{W}_u(\boldsymbol{x}^t)}{\tilde{W}_u(\boldsymbol{x}^{t-1})}.$$

We observe that

$$\frac{\tilde{W}_u(\boldsymbol{x}^n)}{\tilde{W}_u(\boldsymbol{x}^{n-1})}=\frac{\sum_{\mathcal{P}_i}2^{-C(\mathcal{P}_i)}\tilde{W}(\boldsymbol{x}^n|\mathcal{P}_i)}{\sum_{\mathcal{P}_j}2^{-C(\mathcal{P}_j)}\tilde{W}(\boldsymbol{x}^{n-1}|\mathcal{P}_j)}$$

and this product can be written as

$$\frac{\tilde{W}_u(\boldsymbol{x}^n)}{\tilde{W}_u(\boldsymbol{x}^{n-1})}=\frac{\sum_{\mathcal{P}_i}2^{-C(\mathcal{P}_i)}\prod_{t=1}^{n}\hat{\boldsymbol{b}}_{\mathcal{P}_i}^T[t]\boldsymbol{x}[t]}{\tilde{W}_u(\boldsymbol{x}^{n-1})}$$
$$=\frac{\sum_{\mathcal{P}_i}2^{-C(\mathcal{P}_i)}(\prod_{t=1}^{n-1}\hat{\boldsymbol{b}}_{\mathcal{P}_i}[t]\boldsymbol{x}[t])\hat{\boldsymbol{b}}_{l_i[n]}^T[n]\boldsymbol{x}[n]}{\tilde{W}_u(\boldsymbol{x}^{n-1})}$$

$$= \left(\frac{\sum_{\mathcal{P}_i} 2^{-C(\mathcal{P}_i)} (\prod_{t=1}^{n-1} \hat{\boldsymbol{b}}_{\mathcal{P}_i}[t]\boldsymbol{x}[t])\hat{\boldsymbol{b}}_{l_i[n]}[n]}{\tilde{W}_u(\boldsymbol{x}^{n-1})} \right)^T \boldsymbol{x}[n],$$

where $l_i[n]$ is the labeling operator for the partition i such that $l_i[n]$ provides the node ρ corresponding to $V_{i,k}$ if $\boldsymbol{x}[n] \in V_{i,k}$.

Hence the desired sequential universal portfolio assignment algorithm is given by

$$\tilde{\boldsymbol{b}}_u[n] = \left(\frac{\sum_{\mathcal{P}_i} 2^{-C(\mathcal{P}_i)} \left(\prod_{t=1}^{n-1} \hat{\boldsymbol{b}}_{\mathcal{P}_i}[t]\boldsymbol{x}[t] \right) \hat{\boldsymbol{b}}_{l_i[n]}[n]}{\tilde{W}_u(\boldsymbol{x}^{n-1})} \right). \tag{11}$$

Nevertheless, in this form, the sequential algorithm needs $\mathrm{num}(D) \approx (1.5)^{2^K}$ different sequential algorithms to run on parallel on the sequence of observations. We will now demonstrate that this naive sequential portfolio assignment algorithm can be calculated efficiently by using a context-tree diagram. For this, we will first show that $\tilde{W}_u(\boldsymbol{x}^n)$ can be efficiently calculated by using a context-tree. Next, the ratio $\tilde{W}_u(\boldsymbol{x}^n)/\tilde{W}_u(\boldsymbol{x}^{n-1})$ and the portfolio $\tilde{\boldsymbol{b}}_u[n]$ will be constructed on this context-tree.

To this end, we next *assign* a "wealth" to each node ρ as follows. For a leaf node, the wealth assigned to that node is given by the wealth achieved by its corresponding sequential portfolio, i.e.,

$$\tilde{W}_\rho(\boldsymbol{x}^n) = \prod_{t=1}^{n} \hat{\boldsymbol{b}}_\rho^T[t]\boldsymbol{x}[t]s_\rho[t], \tag{12}$$

which is a function of the performance of the leaf portfolio on the sequence of observations, where $s_\rho[t]$ is the indicator variable for node ρ such that if $\boldsymbol{x}[t] \in V_\rho$, $s_\rho[t]=1$, else $s_\rho[t]=0$. The assigned wealth of an inner node (i.e., not a leaf node) is defined as

$$\tilde{W}_\rho(\boldsymbol{x}^n) = \frac{1}{2}\tilde{W}_{\rho_R}(\boldsymbol{x}^n)\tilde{W}_{\rho_L}(\boldsymbol{x}^n) + \frac{1}{2}\prod_{t=1}^{n} \hat{\boldsymbol{b}}_\rho^T[t]\boldsymbol{x}[t]s_\rho[t], \tag{13}$$

where ρ_R and ρ_L are the left child and the right child of the node ρ, respectively. Here, the assigned wealth of an internal node is a weighted combination of the wealth assigned to each of the children nodes and the wealth achieved by the sequential portfolio of V_ρ. Using the recursion in (13), it can be shown, as in Lemma 2 of [14], that the wealth assigned to the root node $\tilde{W}_r(\boldsymbol{x}^n)$ is equal to the sum of weighted wealth of partitions $\mathcal{P}_i$ in (8), i.e., $\tilde{W}_u(\boldsymbol{x}^n) = \tilde{W}_r(\boldsymbol{x}^n)$.

We now have an efficient way of calculating the universal weighted wealth $\tilde{W}_u(\boldsymbol{x}^n)$. From the bottom of the tree, i.e., from the leaves, we start calculating the assigned wealth recursively until we reach to the root node. For each new sample $\boldsymbol{x}[n]$, we need to repeat this procedure to update the assigned wealth of each node. However, we will next present a sequential update from $\tilde{W}_u(\boldsymbol{x}^{n-1})$ to $\tilde{W}_u(\boldsymbol{x}^n)$ on the context-tree with complexity linear in the depth of the context-tree, i.e., $O(D)$, instead of updating all $2^{D+1}-1$ nodes. This update will be used in (11) to get the final recursion for the universal portfolio and is similar to the sequential update used in [13]. Hence, we will present only the main points in here.

Given $\boldsymbol{x}^{n-1}$ and $\tilde{W}_u(\boldsymbol{x}^{n-1})$, assigned wealth of each node $W_\rho(\boldsymbol{x}^{n-1})$ should be adjusted after observing $\boldsymbol{x}[n]$ to form $\tilde{W}_u(\boldsymbol{x}^n)$. However, owing to the tree structure, only the assigned wealth of nodes that include $\boldsymbol{x}[n-1]$ need to be updated to form $\tilde{W}_u(\boldsymbol{x}^n)$. We have $D+1$ nodes that contain $\boldsymbol{x}[n-1]$: the leaf node that contains $\boldsymbol{x}[n-1]$ and all the nodes that contain the leaf that contains $\boldsymbol{x}[n-1]$. Hence, at each time n, only $D+1$ node wealth in $\tilde{W}_u(\boldsymbol{x}^{n-1})$ must be adjusted to form $\tilde{W}_u(\boldsymbol{x}^n)$. This enables us to update $\tilde{W}_u(\boldsymbol{x}^{n-1})$, a mixture of all $\mathrm{num}(D)$ sequential portfolios with only $D+1$ updates, instead of updating all $\mathrm{num}(D) \approx (1.5)^{2^K}$ nodes to reach $\tilde{W}_u(\boldsymbol{x}^n)$.

Suppose $\boldsymbol{x}[n-1]$ belongs to the lowest leaf of the tree in Fig. 11, i.e., $\{x_1[n-1] \geq x_2[n-1] > 1]\}$. All the nodes along the path of nodes indicated by filled circles in Fig. 11 include $\boldsymbol{x}[n-1]$ and only these need to be updated after observing $\boldsymbol{x}[n]$. For any $\boldsymbol{x}[n-1]$ there exits such a path of $D+1$ nodes. Here, we represent the root node as $\rho = r$; left and right children of the root node as r_L and r_R; and recursively, the left child of the left child of the root node as r_{LL} and the right child of the left child of the parent node as r_{LR}. By this notation it can be shown that $\tilde{W}_u(\boldsymbol{x}^{n-1})$ can be compactly represented as sum of $D+1$ terms [13], collecting all terms that will not be affected by $\boldsymbol{x}[n]$, i.e.,

$$\tilde{W}_u(\boldsymbol{x}^{n-1}) = \sum_{k=0}^{D} \beta_k[n-1] \prod_{t=1}^{n-1} \hat{\boldsymbol{b}}_{\rho_k}^T[t]\boldsymbol{x}_{\rho_k}[t]s_{\rho_k}[t], \tag{14}$$

where ρ_k are the nodes that will be affected by $\boldsymbol{x}[n]$, i.e., for this example, $\rho_0 = r$, $\rho_1 = r_R$, $\rho_2 = r_{RR}$, and $\beta_k[n-1]$ are terms generated by the nodes that are not affected by $\boldsymbol{x}[n]$. We will enumerate the affected nodes using the notation ρ_k, $k=0,\ldots,D$ and for each affected node ρ_k, $\beta_k[n-1]$ contains products of node wealth $\tilde{W}_\rho(\boldsymbol{x}^n)$ that share the same parent nodes with ρ_k but will be unchanged by $\boldsymbol{x}[n]$ (i.e., the sibling node of an affected node that does not include $\boldsymbol{x}[n-1]$). As an example, consider the same tree of depth $D=2$ in Fig. 12 with the assigned node wealth. Then, it is shown in [13] that for each time $n-1$, $\beta_k[n-1]$ can be calculated recursively. We start from the root node $\rho_0 = r$, $\beta_0[n-1] = \frac{1}{2}$. We next recursively define: $\beta_k[n-1] = \frac{1}{2}\beta_{k-1}[n-1]\tilde{W}_{\rho_{k,\mathrm{sib}}}(\boldsymbol{x}^{n-1})$, where $\rho_{k,\mathrm{sib}}$ is the sibling node of the affected node ρ_k, e.g., for $\rho_1 = r_R$, sibling node is $\rho_{1,\mathrm{sib}} = r_L$, for $\rho_2 = r_{RR}$, sibling node is $\rho_{2,\mathrm{sib}} = r_{RL}$. Hence, at each time $n-1$, $\beta_k[n-1]$ can be calculated recursively with only D updates. Clearly in the calculation of $\beta_k[n-1]$, we use the nodes that will be unchanged by $\boldsymbol{x}[n]$. Thus, to obtain $\tilde{W}_u(\boldsymbol{x}^n)$, we need to update only the product terms in (14). Since for all the product terms in (14) the last sample is $\boldsymbol{x}[n]$ by definition

$$\tilde{W}_u(\boldsymbol{x}^n) = \sum_{k=0}^{D} \beta_k[n-1] \left(\prod_{t=1}^{n-1} \hat{\boldsymbol{b}}_{\rho_k}[t]\boldsymbol{x}_{\rho_k}[t]s_{\rho_k}[t] \right) \hat{\boldsymbol{b}}_{\rho_k}^T[n]\boldsymbol{x}[n],$$

hence the sequential update for $\tilde{W}_u(\boldsymbol{x}^n)$ from $\tilde{W}_u(\boldsymbol{x}^{n-1})$. A complete algorithmic description of this tree update with required storage and number of operations will be given in Section 2.4.

Thus, $\tilde{W}_u(\boldsymbol{x}^n)/\tilde{W}_u(\boldsymbol{x}^{n-1})$ can be written as

$$\frac{\tilde{W}_u(\boldsymbol{x}^n)}{\tilde{W}_u(\boldsymbol{x}^{n-1})} = \sum_{k=0}^{D} \alpha_k[n-1]\hat{\boldsymbol{b}}_{\rho_k}^T[n]\boldsymbol{x}[n],$$

where weights $\alpha_k[n-1]$ are defined as

$$\alpha_k[n-1] \triangleq \frac{\beta_k[n-1]\prod_{t=1}^{n_{\rho_k}-1}\hat{\boldsymbol{b}}_{\rho_k}^T[t]\boldsymbol{x}_{\rho_k}[t]}{\tilde{W}_u(\boldsymbol{x}^{n-1})}.$$

Hence the final universal portfolio is given by

$$\tilde{\boldsymbol{b}}_u[n] = \sum_{k=0}^{D}\alpha_k[n-1]\hat{\boldsymbol{b}}_{\rho_k}[n],$$

where ρ_k are the nodes such that $\boldsymbol{x}[n-1]\in V_{\rho_k}$, i.e., affected nodes. This completes the proof of Theorem 2. □

Appendix C. Outline of proof of Theorem 3

To get the universal algorithm of Theorem 3, we select particular sequential portfolios $\hat{\boldsymbol{b}}_\rho[t]$ for each node. For each node ρ, we assign a sequential portfolio from [11] that is trained just on the sequence of observations belonging to that node, i.e.,

$$\hat{\boldsymbol{b}}_\rho[n] = \frac{\int_{\mathcal{B}}\boldsymbol{b}\prod_{k=1}^{n_\rho}\boldsymbol{b}^T\boldsymbol{d}_\rho[k]\boldsymbol{\mu}(\boldsymbol{b})}{\int_{\mathcal{B}}\prod_{k=1}^{n_\rho}\boldsymbol{b}^T\boldsymbol{d}_\rho[k]\boldsymbol{\mu}(\boldsymbol{b})}, \tag{15}$$

where $t_\rho^{n_\rho}=\{t : x[t-1]\in V_\rho\}$, $\boldsymbol{d}_\rho^{n_\rho}=\{\boldsymbol{x}[t_\rho^{n_\rho}[k]]\}_{k=1}^{n_\rho}$ and n_ρ is the length of the sequence that belongs to node ρ until time n. From the proof of Theorem 1, we observe that each $\hat{\boldsymbol{b}}_\rho[t]$ asymptotically achieves the performance of best CRP for that region yielding the performance bound

$$\ln\prod_{l=1}^{n_k}\tilde{\boldsymbol{b}}_k^T[l-1]\boldsymbol{x}_k[l] \ge \ln\prod_{l=1}^{n_k}\boldsymbol{b}_k^T\boldsymbol{x}_k[l] - \frac{(m-1)}{2}\ln(n_k+1) - \ln(2).$$

Application of this performance bound into K_i separate regions for each $\mathcal{P}_i$ and combining the result with the bound of Theorem 2 yields Theorem 3. □

Appendix D. Outline of proofs of Theorems 4 and 5

Proofs of Theorems 4 and 5 parallel the proof of Theorem 1. Given the achieved wealth $W^c(\boldsymbol{x}^n|\hat{\boldsymbol{b}}_{\mathcal{P}_i},\mathcal{P}_i)$ for any $\mathcal{P}_i$, we again define a weighted mixture of the wealth as

$$\tilde{W}^c_{\text{temp}}(\boldsymbol{x}^n) \triangleq \sum_{\mathcal{P}_i}2^{-C(\mathcal{P}_i)}\tilde{W}^c(\boldsymbol{x}^n|\hat{\boldsymbol{b}}_{\mathcal{P}_i},\mathcal{P}_i). \tag{16}$$

Here, $\tilde{W}^c_{\text{temp}}(\boldsymbol{x}^n)$ would corresponds to investing on each partition $\mathcal{P}_i$, $2^{-C(\mathcal{P}_i)}$ of the initial money and collecting the wealth at time n. However, instead of dividing the money to all $\mathcal{P}_i$, we can again implement this investment strategy using a context-tree (as in Theorem 1). Since using the context-tree we just efficiently implement the ordinary weighting in (16), our implementation using the context-tree can achieve even larger final wealth due to occasional savings in commission costs, e.g., if $\hat{\boldsymbol{b}}_{\mathcal{P}_i}[n]$ is selling a stock and $\hat{\boldsymbol{b}}_{\mathcal{P}_j}[n]$ is buying the same stock, they can just exchange the stocks instead of selling and buying to avoid the commission. Hence,

$$\tilde{W}^c_u(\boldsymbol{x}^n) \ge \tilde{W}^c_{\text{temp}}(\boldsymbol{x}^n) = \sum_{\mathcal{P}_i}2^{-C(\mathcal{P}_i)}\tilde{W}^c(\boldsymbol{x}^n|\hat{\boldsymbol{b}}_{\mathcal{P}_i},\mathcal{P}_i)$$

for each $\mathcal{P}_i$, where $\tilde{W}^c_u(\boldsymbol{x}^n)$ is the achieved wealth by the context-tree. For implementation on the context-tree, we define the wealth of a leaf node as $\tilde{W}^c_\rho(\boldsymbol{x}^n)$ as the wealth achieved by the node portfolio over the sequence $\boldsymbol{d}_\rho^{n_\rho}$ of observations belonging to this node with commission c. We next define the "assigned" wealth of an inner node as

$$\tilde{W}^c_\rho(\boldsymbol{x}^n) = \tfrac{1}{2}\tilde{W}^c_{\rho_L}(\boldsymbol{x}^n)\tilde{W}^c_{\rho_R}(\boldsymbol{x}^n) + \tfrac{1}{2}\tilde{W}^c_\rho(\boldsymbol{x}^n|\rho), \tag{17}$$

where $\tilde{W}^c_\rho(\boldsymbol{x}^n|\rho)$ is the wealth achieved by this inner node portfolio over the sequence $\boldsymbol{d}_\rho^{n_\rho}$ of observations belonging to this inner node with commission c. Applying the recursion in (17) we obtain that the assigned wealth of the root node is $\tilde{W}^c_r(\boldsymbol{x}^n) = \tilde{W}^c_u(\boldsymbol{x}^n)$. Following similar steps to those of Theorem 1, the final portfolio is given by

$$\tilde{\boldsymbol{b}}_u[n] = \sum_{k=0}^{K}\alpha_k^c[n-1]\hat{\boldsymbol{b}}_{\rho_k}[n]s_{\rho_k}[n],$$

where ρ_k are the nodes such that $\boldsymbol{x}[n-1]\in V_{\rho_k}$, i.e., affected nodes, $\hat{\boldsymbol{b}}_{\rho_k}[n]$ is the corresponding sequential portfolio for that node and the weights are defined as

$$\alpha_k[n-1] \triangleq \frac{\beta_k[n-1]\tilde{W}^c_{\rho_k}(\boldsymbol{x}^n|\rho_k)}{\tilde{W}^c_u(\boldsymbol{x}^{n-1})}.$$

This completes the outline of proof of Theorem 4. □

For Theorem 5, as the first step, given any $\mathcal{P}_i$, we will try to find a sequential algorithm that achieves $W^c(\boldsymbol{x}^n|\mathcal{B}_i,\mathcal{P}_i)$ for any $\mathcal{B}_i=[\boldsymbol{b}_{i,1},\ldots,\boldsymbol{b}_{i,K_i}]$, i.e., the wealth achieved using CRPs $\boldsymbol{b}_{i,k}$ for each region independently with transaction cost c. We will then generalize this result using a double mixture approach to get the final universal algorithm following the steps of Theorem 1 or Theorem 4.

We first construct K_i time vectors (or index sequences) of length $n_{i,k}$ each, $t_{i,k}^{n_{i,k}}=\{t : s_i[t-1]=k, t=1,\ldots,n\}$, with k=1,…,K, and sequences $\boldsymbol{x}_{i,k}^{n_{i,k}}=\{\boldsymbol{x}[t_{i,k}[k]]\}_{k=1}^{n_{i,k}}$. We then define the wealth achieved by a CRP on this sequence of observations with transaction cost c as $W^c(\boldsymbol{x}_{i,k}^{n_{i,k}}|\boldsymbol{b})$. By this, we replace Cover's algorithm in Theorem 1 with algorithm from [17]

$$\tilde{\boldsymbol{b}}_j^c[t] \triangleq \frac{\int_{\mathcal{B}}\boldsymbol{b}W^c(\boldsymbol{x}_{i,k}^{n_{i,k}}|\boldsymbol{b})\boldsymbol{\mu}(\boldsymbol{b})}{\int_{\mathcal{B}}W^c(\boldsymbol{x}_{i,k}^{n_{i,k}-1}|\boldsymbol{b})\boldsymbol{\mu}(\boldsymbol{b})}, \tag{18}$$

where $\boldsymbol{\mu}(\boldsymbol{b})$ is now an mth order uniform distribution. We define the wealth achieved by this algorithm over the period $\boldsymbol{x}_{i,k}^{n_{i,k}}$ as $W^c(\boldsymbol{x}_{i,k}^{n_{i,k}}|\tilde{\boldsymbol{b}})$ which has the following result:

$$\ln(W^c(\boldsymbol{x}_{i,k}^{n_{i,k}}|\tilde{\boldsymbol{b}})) \ge \ln(W^c(\boldsymbol{x}_{i,k}^{n_{i,k}}|\boldsymbol{b}) - (m-1)\ln(1+(1+c)n_{i,k})) \tag{19}$$

for any $\boldsymbol{b}$. Hence, the algorithm given in (18) asymptotically achieves the performance of the best CRP for that region. Given any $\mathcal{P}_i$, we define $\tilde{W}^c(\boldsymbol{x}^n|\mathcal{P}_i)$ as the wealth achieved by using the portfolio defined in (18) for each region. We derive a bound on $\tilde{W}^c(\boldsymbol{x}^n|\mathcal{P}_i)$ by applying (19) for each region, however, we also need to account for the cost of switching portfolios at region change. If we switch from region k to l in $\mathcal{P}_i$, then we need to adjust the portfolio vector $\tilde{\boldsymbol{b}}_k^c[t-1]$ to $\tilde{\boldsymbol{b}}_l^c[t-1]$, where $\tilde{\boldsymbol{b}}_k^c[t-1]$ and $\tilde{\boldsymbol{b}}_l^c[t-1]$ are the corresponding sequential portfolios for that regions, respectively. Hence, we need to account for this transaction cost. At each switch, in the worst case, we can only lose the fraction c of total wealth. Hence, further reducing the wealth by $\ln((1-c))$ (i.e., scaling down the wealth by $(1-c)$ in the worst case per transition) results in

$$\ln(\tilde{W}^c(\boldsymbol{x}^n|\mathcal{P}_i)) \geq \ln\{W^c(\boldsymbol{x}^n|\boldsymbol{B}_i,\mathcal{P}_i)\} - (m-1)\sum_{k=1}^{K_i}\ln(1+(1+c)n_{i,k}) - \ln\left(\frac{1}{(1-c)^{N_i}}\right). \quad (20)$$

After this point, the proof of Theorem 5 follows that of Theorem 1 (or Theorem 4) where we construct weighted universal portfolios on context-trees. This completes the outline of proof of Theorem 5. □

References

[1] T. Cover, E. Ordentlich, Universal portfolios with short sales and margin, in: Proceedings of ISIT, 1998, p. 174.
[2] V. Vovk, C. Watkins, Universal portfolio selection, in: COLT, 1998, pp. 12–23.
[3] D.P. Helmbold, R.E. Schapire, Y. Singer, M.K. Warmuth, Online portfolio selection using multiplicative updates, Mathematical Finance 8 (4) (1998) 325–347.
[4] E. Charniak, Statistical Language Learning, MIT Press, Cambridge, MA, 1993.
[5] L. Rabiner, B.-H. Juang, Fundamentals of Speech Recognition, Prentice Hall, 1993.
[6] L. Rabiner, R.W. Schafer, Digital Processing of Speech Signals, Prentice Hall, 1978.
[7] H. Glotin, D. Vergyr, C. Neti, G. Potamianos, J. Luettin, Weighting schemes for audio–visual fusion in speech recognition, in: ICASSP, 2001, pp. 173–176.
[8] A. Morris, A. Hagen, H. Glotin, H. Bourlard, Multistream adaptive evidence combination to noise robust ASR, in: ICASSP, 2001, pp. 173–176.
[9] S.S. Kozat, A.C. Singer, Switching strategies for sequential decision problems with multiplicative loss with application to portfolios, IEEE Transactions on Signal Processing 57 (2009) 2192–2208.
[10] S.S. Kozat, A.C. Singer, Universal constant rebalanced portfolios with switching, in: ICASSP, 2007, pp. 1129–1132.
[11] T. Cover, E. Ordentlich, Universal portfolios with side-information, IEEE Transactions on Information Theory 42 (2) (1996) 348–363.
[12] S.S. Kozat, K. Visweswariah, R. Gopinath, Feature adaptation based on Gaussian posteriors, in: ICASSP, 2006.
[13] S.S. Kozat, A.C. Singer, G.C. Zeitler, Universal piecewise linear prediction via context trees, IEEE Transactions on Signal Processing 7 (2007) 3730–3745.
[14] F.M.J. Willems, Y.M. Shtarkov, T.J. Tjalkens, The context-tree weighting method: basic properties, IEEE Transactions on Information Theory 41 (1995) 653–664.
[15] A. Agarwal, E. Hazan, Efficient algorithms for online game playing and universal portfolio management, in: Electronic Colloquium on Computational Complexity, 2006.
[16] J.E. Cross, A.R. Barron, Efficient universal portfolios for past dependent target classes, Mathematical Finance 13 (2) (2003) 245–276.
[17] A. Blum, A. Kalai, Universal portfolios with and without transaction costs, Machine Learning 30 (1) (1998) 23–30.
[18] Y. Singer, Switching portfolios, in: Proceedings of Conference on Uncertainty in AI, 1998, pp. 1498–1519.
[19] G. Stoltz, G. Lugosi, Internal regret in online portfolio selection, Machine Learning 59 (2005) 125–159.
[20] A. Kalai, S. Vempala, Efficient algorithms for universal portfolios, in: IEEE Symposium on Foundations of Computer Science, 2000, pp. 486–491.
[21] A. Borodin, R. El-Yaniv, V. Govan, Can we learn to beat the best stock, Journal of Artificial Intelligence Research 21 (2004) 579–594.
[22] A. Agarwal, E. Hazan, S. Kale, R.E. Schapire, Algorithms for portfolio management based on the Newton method, in: Proceedings of 23rd International Conference on Machine Learning, 2006, pp. 9–15.
[23] F.M.J. Willems, Coding for a binary independent piecewise-identically-distributed source, IEEE Transactions on Information Theory 42 (1996) 2210–2217.
[24] F. Willems, The context-tree weighting method: extensions, in: IEEE International Symposium on Information Theory, 1994.
[25] F.M.J. Willems, Y.M. Shtarkov, T.J. Tjalkens, Context weighting for general finite-context sources, IEEE Transactions on Information Theory 47 (1996) 3027–3041.
[26] B.Y. Ryabko, Twice-universal coding, Problems of Information Transmission 20 (3) (1984) 173–177.
[27] G.I. Shamir, N. Merhav, Low-complexity sequential lossless coding for piecewise-stationary memoryless sources, IEEE Transactions on Information Theory 45 (5) (1999) 1498–1519.
[28] M.H.A. Davis, A.R. Norman, Portfolio selection with transaction costs, Mathematics of Operations Research 15 (4) (1990) 676–713.
[29] E. Ordentlich, T. Cover, The cost of achieving the best portfolio in hindsight, Mathematics of Operations Research 23 (4) (1998) 960–982.
[30] L. Gyorfi, G. Lugosi, F. Udina, Nonparametric kernel-based sequential investment strategies, Mathematical Finance 16 (2) (2006) 337–357.
[31] J.M. Mulvey, K.D. Simsek, Rebalancing strategies for long-term investors, in: E.J. Kontoghiorghes, B. Rustem, S. Siokos (Eds.), Computational Methods in Decision-Making, Economics and Finance, 2002, pp. 15–33.
[32] M.W. Brandt, P. Santa-Clara, R.I. Valkanov, Parametric portfolio policies: exploiting characteristics in the cross section of equity returns, in: EFA 2005 Moscow Meetings Paper.