9.3 Online Trading

Ming Lu

2022-04-02

The one-way trading game consists of a trader who needs to exchange some initial wealth given in some currency to some other currency.2 To fix ideas let’s assume the trader wants to exchange dollars to euros. The trader can partition his initial wealth and trade the parts through a sequence of time instants, each part at a different exchange rate offered at the time. Once all the dollars have been exchange the total return of the trader is the amount of euros he accumulated. The problem is then how to do this sequential exchange to maximize total return. Some variants of this problem include: knowledge of a minimum m and a maximum M prices within which exchange rates fluctuates, or the number n of time instants for making the trade. Note that the one-way trading problem resembles the price search problem, where the online player has the apparent advantage of partition his wealth and select different prices to buy sequentially. Nevertheless, it can be shown that the total return is the same as the price obtained by an optimal randomized search algorithm. This is the content of the following theorem of El-Yaniv et al. (2001)

Theorem 9.3 For any deterministic one-way trading algorithm , there exists a randomized algorithm that makes a single trade, and such that for any exchange rate sequence . Conversely, for any randomized single trade algorithm , there exists a one-way trading deterministic algorithm , such that for any sequence of exchange rates

Proof Let the budget for investment be of . Suppose a deterministic algorithm trades a fraction of the budget at the -th period, . Each and ; hence, can be interpreted as the probability of trading the entire budget at the -th period. The average return of this randomized algorithm equals the return of the deterministic algorithm. Conversely, any randomized single trade algorithm can be considered as a mixed strategy (i.e., a probability distribution over

deterministic algorithms), and by linearity of expectation, and that the sum of all traded amounts is 1 , this mixed strategy is equivalent to a deterministic algorithm that trades the initial investment budget through a sequence of fractions.

Can we make profits in the stock markets with online trading algorithms? Theorem assure us that on average it is possible to obtain the same profit applying either a trading strategy that at the time of trade uses all our budget, or a strategy consisting in trading one-way using fragments of our budget sequentially until it is all used up. However, a particular online trading strategy might be more profitable than others, at a certain periods of time, depending on the conditions of the market and the nature of the assets (e.g. their volatility, industrial sectors, economic and social factors, etc.), and so it is of interest to evaluate the performance on real data of different online trading strategies typified by either way of using the initial budget for investment, i.e., by using it all in each transaction, or by fractions. Online trading strategies that use all the budget in each transaction can be defined from the online price search problems of the Sect. ; other online strategy of this type is the difference maximization in Kao and Tate (1999). On the other hand, online strategies that trade with fractions of the allocated budget for investments are to some extend versions of the thread-based policy described in El-Yaniv (1998); others in this family are the max-min search in Loren et al. (2009), the dollar-cost averaging and the value averaging, these last two coming from the folklore in financial markets (but see Marshall for a statistical comparison and El-Yaniv (op. cit.) for performance bounds under certain restricted scenario). We propose as a research problem the question posed at the beginning of this paragraph.

# 9.4 Online Portfolio Selection

In Section we approached the problem of portfolio selection from an offline perspective. The downside with that view is that the best performing offline strategies, as the best constant rebalancing portfolio strategy , are not realistically attainable since it requires full knowledge of the future. Thus, we frame here the portfolio selection problem as a online problem, which is in fact a more natural model since decisions about portfolio management are made in practice only with current information. Nonetheless the offline strategies we have already studied for the portfolio selection problem will be important benchmarks for testing the quality of online solutions through competitive analysis.

Recall that the space of portfolio vectors is , and that denotes the wealth factor achieved by the sequence of portfolio vectors, , with respect to the sequence of market vectors . An online portfolio selection (or rebalancing) strategy is defined as follows.

where is the portfolio used at trading time given only past securities price relatives vectors (i.e. market vectors) . An online portfolio selection algorithm is an algorithm that produces an online portfolio selection strategy.

The performance of an online portfolio selection algorithm can be measured with the same worst-case measures used for offline algorithms, namely: - by comparing its compounded return to that of a benchmark offline algorithm, e.g., the best performing stock or the best constant rebalanced strategy ; - using the exponential growth rate of wealth , .

This is considered in a competitive analysis of performance, where the competitive ratio of is given by

where is an optimal offline portfolio selection algorithm, which is usually taken as . Recall that the is a very high benchmark, since as an offline strategy it outperforms: the best performing stock (and hence, buy and hold); the DJIA index (or any arithmetic market index); and the Value Line (see Example 8.4). Thus, designing an online portfolio selection strategy with a wealth factor similar to the wealth factor achieved by the algorithm, for any sequence of market vectors , would be a much desirable strategy. That an online portfolio selection strategy performing as well as is possible is what we shall see in the Sect. 9.4.1.

# 9.4.1 The Universal Online Portfolio

Cover (1991) proposed an online portfolio selection strategy that performs asymptotically as well as the best constant rebalanced portfolio . Cover’s Universal Online Portfolio (UOP) algorithm assigns a uniform distribution of the initial wealth to the initial portfolio, at the beginning of the first trading period, and for subsequent trading periods selects the performance weighted average of all portfolios over the partial sequence of price relative information known up to the current period. Formally it is defined as the following strategy:

where

and the integration is over the set of -dimensional portfolios . The initial wealth , for it corresponds to .

To have a real sense of the computations involved in this strategy, rewrite the formulas as follows:

where . Note also that , and by the definition of this wealth factor, the strategy is performing a constant rebalance up to time , since the distribution of weights does not changes. Thus, using the notation of Example we could write . And now, let us look at a small specific case.

Example 9.1 Consider a market with three stocks; that is, and the set of possible portfolios is a 2-dimensional space, where each portfolio , with and . The simplex that these set of points describe in is shown in the figure. Then the universal online portfolio strategy consists of

where

and is a polynomial of degree at most in the variables and .

The wealth factor resulting from applying the universal online portfolio strategy UOP is given by

This can be seen to be the average of the wealths achieved by the individual strategies in the simplex .

Proof

From previous remark, , and hence, an immediate consequence of Eq. (9.7) is

under uniform distribution of all portfolios. Other interesting properties of the wealth achieved by the universal online portfolio strategy can be derived from (9.7).

Theorem 9.4 (Properties of UOP) (1) is invariant under permutations of the sequences of stock market vectors . (2) The Universal Portfolio exceeds the Value Line index (VLIC):

Proof (1) In Eq. (9.7) the integrand

and not matter how you permute the stock vectors , the product yields the same result. (2) We sketch the main idea of the proof, and leave as an exercise for the reader to complete the proof (or read through (Cover 1991, Prop. 2.5)). Rewrite in terms of exponential growth rate

Next, use Jensen’s inequality to push everything inside the exponential, do the appropriate cancelations and further manipulations (e.g. recover and apply Jensen’s inequality a second time) to get the desired result.

The first property assess for the robustness of the universal portfolio: it implies that violent periods of changes in the market prices have no worse consequences for the wealth of the universal portfolio, that if the days comprising those unsteady times are dispersed among the regularly behaved price times. The second property shows that the universal portfolio performs as well as the best constant rebalanced portfolio (offline) strategy, , at least with respect to the VLIC index.

Furthermore, a much stronger property of the wealth factor function of the universal online portfolio strategy states that this is similar (or asymptotically close) to the wealth factor of ; that is

for arbitrary sequence of stock vectors . Note that the difference is between the exponential growth rates of the two algorithms: Eq. (9.8) is equivalent to . Hence, this equation says that

the strategy UOP should perform as well as the best constant rebalanced portfolio to the first order in the exponent. The limit in (9.8) is a consequence of the following result. Theorem 9.5 For all markets with stocks, for all periods and all securities price relatives sequence ,

Proof The idea is that portfolios “near” to each other perform similarly, and that a large fraction of portfolios are near the optimal one. Suppose, in hindsight, that is an optimal CRP for the market of stocks. A portfolio near to is given by , for some . Then, a single period’s gain of is at least times the gain of , and so, over periods

A considerable large volume of portfolios is sufficiently near , because the set of near portfolios is a shrunken simplex , translated from the origin to . The volume of simplex is , and

From these equalities we get . Now, using equations (9.7) and (9.10), together with previous observations, we get

Take . Then

All of the previously stated properties of the universal portfolio strategy makes it as profitable as the best constant rebalanced portfolio , and yet realistic due to its online character, so what’s the catch (if any)? Well, indeed, there is a catch, and it is its computational complexity. Observe that the computation of the universal portfolio involves solving an integral over an -dimensional simplex. This has an exponential cost relative to , and even the numerical approach to compute these integrals proposed by Cover in its original work (Cover $1991, \S 8$ ), turns intractable beyond nine assets. Cover proposes to sample portfolios at each trading period and approach the integral by averaging on the wealth of CRP on these samples; more precisely, his suggestion is to fix some level of precision , and take a finite sum over all possible portfolios of the form , where each is a nonnegative integer and . Helmbold et al. (1998) tried this approximation method with real data to select portfolios consisting of nine stocks, and the computing time exceeded hours to calculate the universal portfolio updates over a 22-year trading period. Blum and Kalai (1999) suggested to do randomized sampling to approximate the universal portfolio selection algorithm, and possibly alleviate the computational costs at the sacrifice of precision. We shall review this and other efficient approximation algorithms for UOP in the Sect. 9.4.2.

# 9.4.2 Efficient Universal Online Portfolio Strategies

Blum and Kalai (1999) uniform randomized approximation. Because the universal portfolio strategy is a weighted average of all CRPs, Blum and Kalai suggested to do the following natural uniform randomized approximation. (1) Choose portfolios uniformly at random (i.e., many weight vectors ). (2) Invest a fraction of the money in each of the CRPs and let it sit within these CRPs (i.e. do not transfer between CRPs) all through the periods of trading.

Note that this is like Cover’s approximation that we mentioned before, but instead of considering all possible portfolios, just considers portfolios randomly chosen. How good is this strategy? If the best constant rebalanced portfolio achieves a wealth in the order of (UOP, , for some real number , then by Chebyshev’s inequality one can guarantee that using random portfolios, the approximation achieves a wealth at least , with probability . For a given market one can determine in hindsight the optimal CRP, and then estimate (Helmbold et al. (1997) give a method to do this). However, note that by Theorem the value of could be as big as . Nonetheless, experiments on stock market data by Cover (1991) and Helmbold et al. (1998), all give a ratio of , for various combinations of two stocks, hence making this approximation particularly promising.

Kalai and Vempala (2003) non-uniform randomized approximation. The key idea of Kalai and Vempala is to sample portfolios on a discretization of the simplex , and construct each sample as a random walk. In order to do this is divided into grids of size , and the wealth function is defined on integral multiples of . Then

all portfolios begin at the center of the grid, the point , and continues through a random walk, each step taken according to their wealth performance, as determined by . The details of Kalai and Vempala algorithm, which they named R-Universal, follows below. Its parameters are: , the spacing of the grid; , minimum coordinate; , the number of samples; , the number of steps in the random walk.

Kalai and Vempala showed that, with this non-uniform sampling, the R-Universal algorithm has a performance guarantees of at least , with probability , and a computational cost bounded by a polynomial in , , and .

Other approximation algorithms. Helmbold et al. (1998) give an experts-based approximation algorithm for the UOP, the investment strategy, which uses a weight assignment based on a multiplicative update rule, as those employed in online regression; that is, to find a wealth distribution vector that maximizes , where is a parameter representing the learning rate, and is a distance measure that serves as a penalty term. This investment strategy has a computational cost linear in , but performance guarantees inferior to the one given by Theorem 9.5. Agarwal et al. (2006) give an algorithm based on the Newton method for offline optimization. It has a computational cost of order , but better performance guarantees than the strategy.

What lies beyond? In this presentation of the problem of portfolio selection we have at all times assumed that there are zero transaction costs, nor commissions for trading. The analysis of the universal portfolio strategy in the presence of transaction costs is treated by Blum and Kalai (1999). Another enhancement to the model of universal portfolio selection is to consider side information. This was proposed and analyzed first by Cover and Ordentlich (1996). For further details on these extensions and other related subjects see El-Yaniv (1998) and Cesa-Bianchi and Lugosi (2006), and references therein.

9.5.5: Consider a portfolio with one risk-free asset and one stock. The stock is highly volatile having a price relative factor of on odd days, and of on even days. Thus,

is the sequence of price relative vectors, where the first component corresponds to the risk-free asset and the second component to the stock. (i) Show that a constant rebalanced strategy with weights , , on the above sequence of market vectors gives a wealth, for even,

(ii) Using Eq. (9.11) show that the best constant rebalanced portfolio is obtained with , and compute the value of , for even. (Hint: recall that is the value that maximizes (9.11)). (iii) What is the value of , for even? Try at least an approximate solution.