University of Toronto
Department of Computer Science
CSC 270 - Fall 1994

Assignment 3: Beating the Stock Market

Due: in tutorial, November 28

In 1993, a group of computer scientists and mathematicians proved that it was possible to specify an optimal money-making strategy for the buying and selling of some currency or stock, provided that a profitable off-line strategy existed. In less technical terms, this meant that if someone who knew the future could be guaranteed to make money on the stock market, then so could you. The optimal money-making strategy can be thought of as the transactions you should make in a two-player game, in which, each day, your opponent chooses the exchange rate and you choose the amount of stock to buy or sell.

First, we assume an exchange rate sequence $E = e_1, e_2, \cdots$, where e_i specifies the number of units of some stock that can be purchased for one dollar on day i. Each day, you may purchase or sell a certain amount of stock, subject to the constraints that you cannot spend more money than you have available, and you cannot sell more stock than you currently own. It is convenient to represent the buying or selling of stock on any day i by a single number $s_i \in [0, 1]$. We conceptualize every transaction as if you first sell all of your available stock, then spend a fraction, s_i of your available dollars to purchase stock. By this notation, $s_i = 0$ means selling all of your stock, and $s_i = 1$ means using all of your available dollars to buy stock.

The trading strategy requires knowledge of the number of days, n, during which the trading is to take place, and an estimate of the optimal off-line return, Π, achievable during this time. The off-line return denotes the amount of money that someone who knows the future would end up with at the end of the trading period, assuming they begin trading with \$1. With these parameters given, the optimal strategy is to find, each day, the value of s_i which maximizes the minimum return realizable over the remaining $n-1$ days. The on-line return is then described by the recursive formula:

$$\begin{aligned} R_n(\Pi, e_1) &= \max_{\substack{\text{fraction of dol-}\\ \text{lars } s_1 \text{ to trade}}} \ \min_{\substack{\text{2nd day}\\ \text{rate } e_2}} \begin{bmatrix}\text{total worth in dol-}\\ \text{lars after 2nd day}\\ \text{rate is revealed}\end{bmatrix} \cdot R_{n-1}\left(\text{updated } \Pi, e_2\right) \\ &= \max_{0 \le s_1 \le 1} \ \min_{e_2 \ge \frac{e_1}{\Pi}} \left[\left(\frac{e_1}{e_2} - 1\right) s_1 + 1\right] \cdot R_{n-1}\left(\min\left\{\Pi, \frac{\Pi e_2}{e_1}\right\}, e_2\right) \qquad (1) \end{aligned}$$

That is, you must choose the value of s_1 which maximizes $R_n(\Pi, e_1)$ knowing that for any choice of s_1 your opponent will choose the value of e_2 which minimizes the expression.

Note that as the number of remaining days in the trading period decreases, so does the potential profit available. Therefore, for each day i of trading, the estimate of Π must be updated by dividing by $\frac{e_{i-1}}{e_i}$ whenever $e_i < e_{i-1}$.[1]

[1] Unfortunately, it is beyond the scope of this assignment specification to explain the motivation for this update rule.

For the penultimate day of trading, the formula is a special case:

$$R_2(\Pi, e_{n-1}) = \begin{cases} \text{impossible}; & \text{if } \Pi > e_{n-1}/\text{minimum exchange rate} \\ \Pi, \text{with } s_{n-1} = 0; & \text{if } \Pi = 1 \\ \Pi, \text{with } s_{n-1} = 1; & \text{otherwise.} \end{cases} \tag{2}$$

On the last day, of trading, you must exchange all of your stock to dollars, so $s_n = 0$.

What to do

First, use dynamic programming to generate a three-dimensional array of records, $S[i][\Pi][e]$. Each record contains the two fields:

s_i	corresponding to the optimal exchange amount
$R_{n-i+1}(\Pi, e)$	corresponding to the expected return,

for day i, estimated off-line return, Π, and current exchange rate e. Assume that the trading period will be no greater than 10 days, and that, beginning with a total of \$1 to invest, the off-line return will never exceed \$1.50 within this time. By definition, $\Pi \geq 1$. Assume also, that stock prices fluctuate between a minimum of \$1.30 and a maximum of \$1.40 per share.[2] You should divide the range of Π and e into 20 intervals each, and also explore the range of possible exchange amounts, s_i which maximize your return. Be aware that depending on the number of intervals you choose for s_i, it could take a very long time to generate the array. In your report, comment on the complexity of the algorithm. What would happen if we doubled the number of intervals to 40?

Next, describe the set of transactions that the trading strategy recommends, and the resulting return, for the following sequences of exchange rates, with the corresponding initial estimates of off-line return:

- $E = \{1.40, 1.30, 1.40, 1.30, 1.40, 1.30, 1.40, 1.30, 1.40, 1.30\}; \Pi = 1.4485$
- $E = \{1.30, 1.35, 1.40, 1.40, 1.40, 1.40, 1.40, 1.40, 1.35, 1.30\}; \Pi = 1.0769$
- $E = \{1.39, 1.35, 1.31, 1.35, 1.31, 1.35, 1.39, 1.35, 1.31, 1.35\}; \Pi = 1.1579;$

Don't forget to update the estimate of Π before determining the appropriate transaction for each day. Also, do not be surprised if your program recommends very little trading during the first few days of the trading period. Can you explain why this occurs?

In your report, discuss how well you would fare as an investor if you followed the advice of your program.

[2] These values were chosen to reflect American-Canadian currency exchange rates.

Bonus Marks

1. The approach taken by your program for generating the three-dimensional trading strategy array suffers from a potentially serious problem, which may not be apparent to you from the results you obtained. Based on what you have already learned in CSC 270, discuss what could be done to ameliorate this problem.

2. Name the authors of "The Statistical Adversary Allows Optimal Money-Making Trading Strategies," the paper which describes the strategy used in this assignment.