Computer Science 4963/6963 (Internet Economics): Final Project

Due: May 7, 2012

1 Overview

You will be working to extend your trading strategy from HW3 in order to take into account the fact that other traders may also be participating in the market. The information structure is the same, the major difference being that other trading strategies can now also trade and change prices. The order of trading in each round is random.

The deliverables for this project are (1) a concise project report from each team, and (2) code for your bot. Specifics of what we expect are described below.

2 Underlying Model

The underlying model for the true price is the same as in HW3.

There are up to R rounds $1 \ldots R$, each representing a distinct time period. There is an underlying true probability that the event will occur, which is unknown to everyone, but can change over time. At round i, let p_i be the true probability of the event occurring. During this round i, your trader will be told the outcome of one Bernoulli trial (biased coin toss) with success probability p_i. The trader can then buy and sell the security (there are no inventory or cash restrictions – your trader can take arbitrary positive and negative positions).

2.1 Time evolution of p_i

The true probability is a jump process. p_0 is chosen uniformly at random between 0 and 1. At the beginning of a round, p_i is calculated as follows: with probability $1/R$, $p_i \sim N(p_{i-1}, \sigma_{\text{jump}})$, and $p_i = p_{i-1}$ otherwise. That is, at any time, with probability $1/R$ the true probability jumps, and when it does jump, the new true value is drawn at random from a normal distribution centered on the present true value. If $p_i \leq 0$ or $p_i \geq 1$, the security liquidates prematurely at 0 or 100 respectively. Otherwise, the value of the security is $100p_R$ after round R.

2.2 Parameters

$R = 100$ and $\sigma_{\text{jump}} = 0.2$.

3 Competing Algorithms

We have provided code that implements various other types of trading bots. Specifically, there is code for a simple *fundamentals trader*, which receives the same information your bot does and trades in the direction of its belief, which is based on the information it has available to it. There is also code for two different *technical traders*. These are algorithms that do not use information about the fundamental value but instead try to profit from market movements using certain heuristics. In effect, they add noise to the market prices. Details on how to incorporate the different types of traders are in the appropriate readme file – you can control the proportion of fundamentals vs technical traders in the environment, and one of the goals of the project is to analyze performance as you vary this proportion.

The presence of these other traders allows you to potentially glean more information from the market price than would be available to just your own bot. On the other hand, these traders are themselves competing to make money off the market maker, just as you are. Finally, the technical traders add noise to the signal in the market price, but perhaps provide more opportunity for profit!

4 Software Framework

We have provided an extended version of the Python software framework from HW3. Please look at it carefully and read the `README_PART_TWO` file. In addition to the tasks previously handled, the framework now includes additional code for integrating other trading bots.

5 Team Composition

You may work in teams of two or three. If you wish, you can change the team from HW3. Again, please keep in mind that it will be easiest if one of your team members is conversant in Python. Each team needs to hand in only one report (described below).

6 What To Do

First, make sure you understand the assignment, and ask questions if you don't, or even if you just want to clarify! Second, start reading and understanding the code, and also start thinking about what kind of trading strategy you want to use.

7 What To Turn In

I will look specifically for answers to the following questions in your writeup. It is best to address them explicitly. You may add other information that you feel is relevant as well.

1. How does your trader estimate p_i at time i? What algorithm do you use to try and account for the probability of jumps?

2. How do you use the market price in order to improve either your estimate or your trading strategy based on the estimate?

3. At which periods does your trader choose to trade? How did you decide this?

4. How does your trader decide what quantities to trade?

5. Report the mean profit and standard deviation[1] of profits your trader achieves over at least 1000 simulation runs across a variety of different proportions of fundamentals vs. technicals traders. Write at least half a page analyzing these results in detail and explaining why you think they come out the way they do.

6. Graphically depict the evolution of prices in one simulation run which you consider typical for your trader.

Please also email the code for your bot to both `lavoia at cs dot rpi dot edu` and `sanmay at cs dot rpi dot edu` **by 8 AM on Monday May 7**. We will run a trading competition between all the bots in class on May 7.

You will be graded primarily on the quality of your writeup, but also in terms of performance.

[1] Controlling the standard deviation is essentially a way of controlling risk. You may be willing to sacrifice some expected profit in order to have less "downside risk" for your strategy. This is particularly important in the real world, where you don't get to run your strategy thousands of times if you run out of money! It may also become a consideration in our class contest where we won't run as many simulations as you might be able to at home.