TJHSST Senior Research Project Implementing Genetic Algorithms in a Financial Application 2009-2010

Nihaar Sinha

June 15, 2010

Abstract

This project investigates the use of genetic algorithms in a financial application of portfolio optimization. Genetic algorithms have long been used in optimization problems as well as the financial sector. Banks and hedge funds pay millions of dollars to programmers who can develop the most accurate optimization algorithms. What this project does is try to replicate that development on a very small scale- using only three different companies and 8 total shares in the portfolio. The genetic algorithm considers a number of factors in coming up with the optimized solution. These factors include an evaluation of the price to earnings ratio of the stock, the yield on the stock, and a special weight determining how diversified the portfolio is. While the metrics used are to an extent crude, they serve the purpose of demonstrating how a concept such as genetic algorithms can be used in the field of finance.

Keywords: genetic algorithms, algorithmic composition, data importing

1 Introduction

1.1 Rationale

FThe recent global recession caused by the infamous subprime meltdown, caused in large part by faulty use of suspect mathematical algorithms, has brought the use of mathematics in finance to the limelight. Many people are now wondering just how mathematics is used by investment banks and hedge funds, why it failed in the case of subprime, and whether all algorithms used in the financial field are bad. This project attempts to demonstrate how a simple programming concept, genetic algorithms, can be used in a financial application such as portfolio optimization. This will hopefully help give people a slight taste of quantitative finance.

1.2 Purpose

This project helps explicate the fundamentals of quantitative finance. It shows the average person just how concepts in mathematics and computer science, such as genetic algorithms, can be used in a financial application. The application examined in this project is portfolio optimization, a popular existing application of mathematics in finance. Hopefully people will understand not just how mathematics is used in finance, but more importantly where it can go wrong and knowing the limits of mathematical applications in predicting the market. With this understanding, policymakers will make the right decisions as our government works on Wall Street reform and further regulations.

1.3 Importance of Topic

As our government begins the arduous task of instituting financial reforms to prevent a future global recession, it is important to educate on some of the fundamental causes of the economic meltdown, so that these important areas do not fall victim to sweeping overreaction seeded in a lack of understanding. Mathematics and finance go hand-in-hand, and with the falter of some of the more sophisticated market predicting algorithms in the sub-prime crisis, it is important to go back to the basics and emphasize what makes these algorithms work.

2 Background

To appreciate this project, one must first have an understanding of the computer science concept of genetic algorithms algorithm. Genetic algorithms involve the use of the concept of evolution, in which the computer actually executes the process of natural selection and finds a solution amongst several possibilities. This saves the user time and resources, and is a great way to analyze huge sets of data. The use of genetic algorithms is present in several different fields, with finance being one of the most prominent.

1. The genetic algorithm starts by initializing a set of randomly created possible solutions. These possible solutions are called chromosomes. The algorithm then proceeds to sort the possible combinations by rank. The algorithm detects which combinations are ?better? than others based on a certain set of parameters coded into the algorithm. This ability to rank certain combinations as ?better? than others is the essence of genetic algorithms. After the algorithm has sorted all of the combinations in order, a cut-off point is determined and all the combinations below that cut-off point are eliminated. What?s left is a shortened pool of strong chromosomes. These become parent chromosomes, and they are matched up to create child chromosomes. For example, if chromosome 1 was ?AB?, and chromosome 2 was ?CD?, then the children would be ?AD? and ?CB?. The idea is that the strong chromosomes will ?pass on? that strong ranking to their ?children?. Once the crossover is finished, the children chromosomes are ranked just like their parents were. Once again, a cut-off point is determined and all the combinations below that point are eliminated. This iterative process continues until there is one solution left. Sometimes the cycle can lead towards a solution that maybe isn?t ideal. In this case, a mutation is often used. A mutation is a random insertion of a combination that would not be an offspring of two parent chromosomes; it is an intentional ?mistake? used to try and redirect the natural selection process towards the ideal result.

2. A basic understanding of some finance is also required to fully grasp this project. The terms used in this paper, as well as the metrics for ranking the combinations in the genetic algorithm, are the earnings per share, the price to earnings ratio, the yield, and the dividends.

3. The earnings per share, or EPS, of a stock is the amount of profit the corporation is getting on each share of its stock. The price per earnings ratio, or P/E, is calculated by simple dividing the stock price by the EPS. The P/E represents how much the market is willing to pay for that stock. In this way, the P/E is an indicator of how ?fair? a stock price is. A P/E of 16 is considered to be a very ?fair? price. A significantly higher P/E, such as 33, would indicate that people are willing to spend way too much to get the stock and the price is artificially inflated, and vice versa. As such, one of the metrics used in this project?s genetic algorithm is how close the P/E is to 16.

4. The dividend on a stock is the amount of profit one makes from that stock. To give an example, a dividend of 2.50 means that a shareholder would get 2.50 dollars of profit from one share of that stock. The yield is calculated by dividing the dividend by the stock price. It represents the accounting profit a shareholder would be getting from a stock. For example, if the dividend is 2.50 and then stock price is 250 dollars, then the shareholder is getting a yield of 10 percent, that is to say, the shareholder is making back 10 percent of that investment. Therefore, the higher the yield the better the investment. For that reason, another metric used in the genetic algorithm is how high the yield is.

5. Finally, one must be familiar with the concept of diversification. The concept states that putting all one?s eggs in one?s basket, or in this case investing in just one company, is inherently risky and therefore a worse investment than one where shares are spread out over multiple companies and industries. For this reason, the third metric in the genetic algorithm is how diversified the potential portfolio is.

3 Procedure/Methodlogy

First the user is prompted for three stock symbol through three JDialog boxes. Once entered, the program imports the source data for the pages of those specific stocks from Yahoo! Finance. The source code is parsed into Strings, and the program then searches the strings for relevant information (such as the earnings per share) due to previous identification. Once the relevant data is parsed, the program comes up with point scores for the different companies selected, and this data is stored. The genetic algorithm

then initializes several random portfolios of 8 shares consisting of the three picked stocks. These are ranked by the three metrics, the bottom half are eliminated, and the remaining combinations are crossed over. The procedure then repeats until there is one solution left. After a few iterations of this process, a scheduled mutation appears and adds a twist to the pool of combinations.

4 Expected Results

The expected result of this program is a portfolio consisting of a certain combination of 8 shares from the three companies chosen by the user. This portfolio should in theory be the most profitable of all the possible portfolios of 8 shares consisting of those three companies based on the metrics used by the genetic algorithm (how close the P/E is to 16, the yield, and the level of diversification in the portfolio). This should give the user an idea as to how portfolio optimization works.

5 Current Results

At this point I am able to read in stock information from Yahoo! Finance, with the specific stock inputted by the user. I have been able to read in the stock price and other information for the most recent update to the Yahoo! Finance page. The program then stores the last trade as a double, and it prints out the information as well. This is because I use the user input to find the information from the source code of the Yahoo! Finance web page that I import. In the source code, the symbol is lower case. For development purposes, the program currently prints out the entire source code of the Yahoo! Finance page as a string. The program also outputs the source code to a program-generated text file. It then calculates all the metrics and point values for the three user-chosen stocks to use in the genetic algorithm. The program additionally loads a GUI that puts together all the information for the user.

6 Future Research

There is much potential for future research in this area. The most obvious next step would be to introduce more complex and sophisticated metrics in the ranking of the genetic algorithms. The current iteration of this project uses three main benchmarks: the closeness of the price per earnings ratio to 16, the yield, and the level of diversification in the portfolio. Yet there is clearly more to putting together a good portfolio than just these three qualifications. Adding more would improve the accuracy of the algorithm. Other ways to improve the algorithm would include playing around with the mutation rate of the genetic algortihm.

6.1 Software

I will be using Java for this program, because it is supported by almost every operating system and therefore can expose the program to more people.

References

[1] Review Paper. A Survey of Mathematical Finance David Hobson Proceedings: Mathematical, Physical and Engineering Sciences, Vol. 460, No. 2052 (Dec. 8, 2004), pp. 3369-3401 Published by: The Royal Society Stable URL: http://www.jstor.org/stable/4143245

[2] On the Structure of Proper Black-Scholes Formulae, Peter Whittle, Journal of Applied Probability, Vol. 38, Probability, Statistics and Seismology (2001), pp. 243-248, Published by: Applied Probability Trust, Stable URL: http://www.jstor.org/stable/3215883

[3] Optimal and Near-Optimal Advection-Diffusion Finite-Difference Schemes III. Black-Scholes Equation, Ronald Smith, Proceedings: Mathematical, Physical and Engineering Sciences, Vol. 456, No. 1997 (May 8, 2000), pp. 1019-1028, Published by: The Royal Society, Stable URL: http://www.jstor.org/stable/2665477

[4] Far Field Boundary Conditions for Black-Scholes Equations, Raul Kangro, Roy Nicolaides, SIAM Journal on Numerical Analysis, Vol. 38, No. 4 (2001), pp. 1357-1368, Published by: Society for Industrial and Applied Mathematics, Stable URL: http://www.jstor.org/stable/3061926

[5] Black, Merton and Scholes: Their Work and Its Consequences, Ajay Shah, Economic and Political Weekly, Vol. 32, No. 52 (Dec. 27, 1997 - Jan. 2, 1998), pp. 3337-3342, Published by: Economic and Political Weekly, Stable URL: http://www.jstor.org/stable/4406240