A Genetic Algorithm System for Predicting the OEX

Deniz Yuret
Michael de la Maza

1 Introduction

Many have discovered that designing a successful automatic trading system is a Sisyphean task. Powerful new artificial intelligence methods, such as neural networks, have the potential to simplify this task. To that end, this article describes an application of genetic algorithms to predicting the OEX.

2 What is a genetic algorithm?

Genetic algorithms were invented over twenty years ago by John Holland who drew upon sources in biology and economics to develop a general optimization algorithm that proved to be adept at finding good solutions to problems ranging from jet propeller design to protein folding. Although genetic algorithms were developed some time ago, it is only during the last five years that they have achieved widespread popularity within the computer science community and only now are they being applied to stock market prediction.

The basic genetic algorithm maintains a set of individuals which can be thought of as potential solutions to a problem. The best individuals in this set undergo crossover and mutation to produce new individuals that are expected to be better solutions to the problem. After this process is repeated several times, the algorithm stops. The basic genetic algorithm is shown in Table 1 and a stylized flow chart of the procedure is shown in Figure 1.

The application of the genetic algorithm to a particular problem is not a straightforward plug-and-play procedure. It requires the specification of four sub-components:

- A representation of the individuals (potential solutions to the problem). For the application of predicting the OEX, the representation

t = 0
Initialize P(t)
Until done
    Assign fitnesses to the individuals in P(t)
    Select P(t+1) from P(t)
    Apply mutation and crossover operators to P(t+1)
    t = t + 1

Table 1: The basic genetic algorithm. P(t) is the set of individuals at step t. Initialization randomly generates a starting set of individuals.

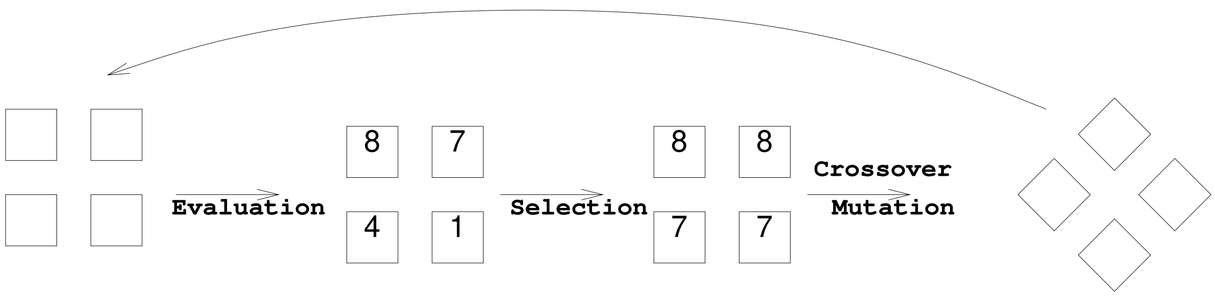

Figure 1: A flow chart of the genetic algorithm. Each box represents an individual. In this example, the selection procedure makes two copies of the top two individuals and discards the worst two. These two individuals then undergo crossover and mutation to form a new population of individuals. This process is then repeated.

will determine the expressivity and complexity of the calculations done in order to make the predictions.

- A fitness criterion that indicates how good an individual is. The higher the fitness, the better the individual.
- Mutation and crossover operators that generate new individuals by modifying the present set of individuals.
- A selection procedure that uses fitness to determine which individuals will participate in mutation and crossover.

In this application, the genetic algorithm generates individuals that are good at predicting one day changes in the OEX using the high, low, and close prices for the previous ten days. With this in mind, let's consider each of these four sub-components in turn.

3 Representation

There are many possible representations for the individuals. Each individual could be a neural network or an if-then rule or a parametric equation or one of any number of other things. We decided upon a very simple if-then rule representation that only compares the values of the data using greater-than (>) and less-than (<) boolean operators. Thus, the genetic algorithm cannot produce a trading system that, as part of its decision procedure, calculates the percentage change in the OEX because the representation does not allow for subtraction or division. An individual, then, is composed of a list of conditions that must be met.

Here is an example of an individual:

((> (C 3) (H 4)) (< (L 2) (C 0)))

C stands for the closing price, H for the high, and L for the low. The number after the letter indicates what day's price is to be used. So, (C 3) means "The closing price three days ago" and (> (C 3) (H 4)) means "The closing price three days ago must be higher than the high price four days ago." The entire individual can be read as: "The closing price three days ago must be higher than the high price four days ago and the low price two days ago must be lower than today's close." Although individuals could be generated that predict that the OEX will go up or down, throughout this article we adopt the convention that all individuals predict that the OEX will rise on the following day.

This individual is tested by using it to make predictions on a contiguous sequence of data. On most days, it would not produce a prediction because its conditions are not met (for example, the closing price three days ago may not be higher than the high four days ago). When its conditions are met the performance statistics, such as the average OEX move on the following day, are updated for this individual.

4 Fitness

Much thought has been put into quantifying the quality of a trading system. Although any one of these formulas could have been used, this genetic algorithm system uses a simple fitness criteria that balances the number of trades with the average move of the OEX when the conditions of the individual are met. The fitness formula is:

$$(\text{number of trades})^{0.7} * (\text{average move per trade})$$

For example, an individual that makes 10 trades and that averages 0.5 OEX points per trade has a fitness of $10^{0.7} * 0.5 \simeq 5 * 0.5 = 2.5$.

Importantly, the fitness of an individual is not computed using the same data that it is tested on, thus preventing overfitting. This will be discussed in more detail in the results section.

5 Mutation and crossover

Mutation and crossover are the mechanisms in genetic algorithms that improve the fitness of individuals. Mutation randomly chooses one of the conditions in an individual and randomly changes it to another one. For example, the individual ((> (C 3) (H 4)) (< (L 2) (C 0))) could be mutated into ((> (C 3) (H 4)) (> (H 9) (L 8))) by mutating the second condition.

Crossover combines two individuals by exchanging their conditions to produce two other individuals, called offspring. A split point is chosen and the conditions to the left of that split point in the first individual and to the right of that split point in the second individual form one of the offspring. The conditions to the right of the split point in the first individual and to the left of the split point in the second individual form the second offspring.

For example, if the two original individuals are ((> (C 1) (C 2)) (> (C 3) (C 4)) (> (C 5) (> C 6))) and ((> (L 1) (L 2)) (> (L 3) (L 4)) (> (L 5) (> L 6))) and the split point is between the first two conditions, then the two offspring are ((> (C 1) (C 2)) (> (L 3) (L 4)) (> (L 5) (> L 6))) and ((> (L 1) (L 2)) (> (C 3) (C 4)) (> (C 5) (> C 6))). The crossover operator can be made more versatile by using two split points, one in each individual. In this case, the offspring may have a different number of conditions than the original individuals.

Why should mutation and crossover lead to better individuals? Mutation increases the diversity of the conditions. It allows conditions that are not already contained in one of the individuals to be introduced. Crossover has the potential to combine the best conditions of two good individuals to produce a better individual.

6 Selection

Selection insures that only the best individuals participate in mutation and crossover. The strongest selection procedure selects only the best individual, while the weakest selection procedure selects all of the individuals. It is best to choose a selection procedure between these two extremes in order to compromise between the desire to fine-tune the best trading system found so far and the need to explore radically different trading systems.

The most common selection procedure replicates an individual a number of times that is equal to its fitness divided by the average fitness:

$$\text{number of copies of individual } i = f_i/F$$

In this equation, f_i is the fitness of individual i and F is the average fitness of all of the individuals. For example, if the fitness of an individual is 24 and the average fitness of the population is 8, then $24/8 = 3$ copies of this individual would be made. When this division does not produce an integer, it is rounded to the nearest integer.

Note that individuals with very low fitness are eliminated and that individuals with very high fitness are copied multiple times. Mutation and crossover take place on this modified set of individuals to produce a new set of individuals.

7 Results

When tested on the last 500 days of a database that ends on June 15, 1993, the accuracy of the best individual is 55.1% and the average OEX move is 0.39 points. The best individual trades a total of 127 times (out of a possible 500). This should be compared to a strategy which always predicts that the OEX will rise on the next day. This strategy is correct 48.8% of the time and the average OEX move is 0.12 points over this same 500 day period. Thus, although the prediction accuracy is about the same, the genetic algorithm produces a strategy that predicts a move in the OEX that is, on average, three times higher.

If only the last 100 days are used, then the accuracy of the best individual improves to 70.4% and the average OEX move increases by 0.49 to 0.88. The best individual makes a total of 24 trades during this 100 day period. This compares favorably to the strategy that always predicts that the OEX will rise, which has an accuracy of 53.5%. The average move for this strategy is 0.2 OEX points.

These results suggest that the genetic algorithm is finding better than average solutions. These promising results are due not only to a carefully designed genetic algorithm, but also to a testing strategy that reduces the negative effects of overfitting. The genetic algorithm is run on a training set of 1000 data points and then a tuning set of 500 data points is used to select the best individual. Finally, statistics are collected on the performance of the best individual on a testing set of 10 days. This procedure, which is diagrammed in Figure 7, is then repeated, each time shifting the training, tuning, and testing sets by 10 days. The results reported above for 500 and 100 day periods are the averages over 50 shifts and 10 shifts. For example, the first training set consists of the first 1000 days in the data base, the second training set consists of days 10 to 1009 and so forth.

Note that the individuals are being trained and tested on different sets of data. When the same set of data points is used to create a trading system

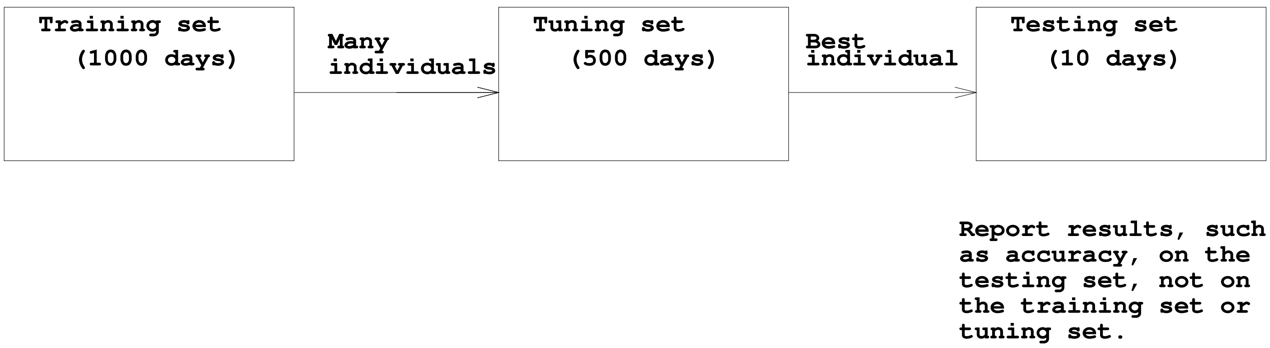

Figure 2: Three non-intersecting data sets reduce the negative effects of overfitting. The genetic algorithm searches for good individuals on the training set. These individuals are then used to predict the OEX on the tuning set, the best one is selected, and its final statistics are computed on the testing set.

and also to test it, the results are sometimes grossly misleading. Using this method, we have found individuals that are accurate over 90% of the time and predict moves that are, on average, over two OEX points! What traders want from a testing system is a good estimate of future profits and that is best done by keeping the training and testing separate. The tuning set functions to reduce overfitting on the training set.

Of course, a trader must find a security that turns these predictions into profits. Currently, we trade just-in-the-money nearest term OEX options. For example, during the middle of January with the OEX between 435 and 440, we would trade January 435 calls and January 440 puts. An obvious extension of this work is to use S&P 500 data, instead of OEX data, and then buy and sell the nearest term contract.

8 Real-time trading

The genetic algorithm system described here was part of a system that we used during the 1993 U.S. Investing Championship (options division). Between September and December the system generated seven trades, all of them profitable, which produced a total return of 43.9%. With the help of this system we are now participating in the 1994 U.S. Investing Championship (futures division) and our account is up 9.8% through January 21.

We expect that genetic algorithms will become part of the arsenal of many traders and believe that they will produce significant profits for astute market participants. We are interested in hearing from readers about their experiences with genetic algorithms and are making a LISP implementation of this system available in order to facilitate experimenting with genetic algorithms.

References

[1] L. Davis, editor. *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York, 1991.

[2] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading, MA, 1989.

[3] J.H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, 1975.