Interactive Estimation of Agent-Based Financial Markets Models

Ihsan Ecemis[1] Eric Bonabeau[1,2] Trent Ashburn[3]

[1]Icosystem Corporation, 10 Fawcett Street, Cambridge, MA 02138, USA
[2]eric@icosystem.com
[3]Tiger, 10 Fawcett Street, Cambridge, MA 02138, USA

Abstract

In this paper we show how an interactive evolutionary computation (IEC) method can be used to perform interactive estimation of agent-based market models.

Keywords: agent-based modeling, interactive evolution.

1. Introduction

There have been a number of attempts over the last decade to model financial markets with agent-based modeling (ABM) [5,6,7]. The dynamics of the stock market results from the behavior of many interacting agents, leading to emergent phenomena that can be understood using a bottom-up, ABM approach. One of the main issue in financial markets ABM is calibration and validation: how can one evaluate the quality of a model, from both a structural perspective (how sound is the model?) and from an econometric perspective (how well does that model reproduce the data quantitatively?). One difficulty is that the use of purely numerical scoring methods to evaluate data fit constrains the search path so dramatically that no good fitting model is found or the best fit is generated by a low-plausibility model. In many situations the fitness function for a model cannot be practically formulated mathematically. This problem can be overcome if subjective factors can guide the search for "good" models, enabling ABM users to integrate financial economics expertise into their models. Boschetti and Moresi [2] have proposed to replace the numerical evaluation of data fit by a subjective evaluation. A technique originally developed to generate "interesting" images and pieces of art [3,8] is used to perform model inversion by integrating subjective knowledge into the evaluation process. The technique, Interactive evolutionary computation (IEC) (see [9] for a review) is a directed search evolutionary algorithm which requires human input to evaluate the fitness of a pattern (here, the fitness might be how well the model reproduces the data *qualitatively*) and uses evolutionary operators such as mutation and crossover to breed the individual-level rules that produce collective-level patterns. In this paper IEC is used to discover the parameters of an agent-based model of a financial market from aggregate observations. The user operates a visualization tool to navigate a parameter space using selection and variation operators. Parameter values define traders and their trading strategies which, in turn, generate a synthetic price history. The user's goal is to find a combination of parameter values that can reproduce the *qualitative features* a target price history. The experiment uses a target price history from a market frenzy that occurred on the London Stock Exchange in September 2002.

2. Agent-Based Model

A simple model of a financial market is used. It has an order management and clearing mechanism, traders operating trading strategies, a market maker posting orders on the book which are matched with traders' orders, and a price history. After each time step, the arithmetic mean of the best bid and best ask is appended to a simulation's price history. The order book matches orders in a continuous double-sided auction. Upon order submission, the book immediately clears any and all clearable orders. Orders are market orders: an offer to buy at the best ask or sell at the best bid. At each time step, traders take turns trading according to their strategies. Traders trade once per time step, each trader submits only market orders, and traders' strategies (described below) are based upon a set of initial conditions and the accumulated price history of the stock. To ensure liquidity in the market, there is one market maker in the simulation, tightening the spread and maintaining order depth. Traders in the model come in four flavors: fundamentalists, chartists, noise traders, and a whoops trader. Each has unlimited buying power, and unlimited shorting power. Only one trade can be made per time step, per trader.

Fundamentalists. Fundamentalists calculate a "true value" of the underlying stock by averaging the price history a certain number of minutes back – a moving average of the form $T=\sum_{x=r}^{r-l} p_x$ where T is the true price, p is a historical price, r is the time of the most recently recorded historical price, and l is the number of time steps used in the computation. The trader then computes an upper threshold, U, and a lower threshold, L. Each is a percentage t away (50%=0.50) from the true price: $U=T(1+t)$ and $L=T(1-t)$. Once the price action has passed through one of the thresholds, a number of time steps must pass (greater than the fundamentalist's reaction time) before the fundamentalist is "awake" and able to trade. Once this reaction time threshold has been passed, the fundamentalist makes one trade per time step according to the rules:

- *Buy* if $A \le L$ and a long position is not held
- *Sell* if $B \ge T$ and a long position is held
- *Short* if $B \ge U$ and a short position is not held
- *Cover* if $A \le T$ and a short position is held

where B is the best bid, and A is the best ask. The trader holds a long position if he has bought, and he no long holds it if he sells; likewise for the short side. Characteristically, a fundamentalist appears to anchor the price history and dampen volatility.

Chartists. Chartists' trading is triggered by momentum – they trade in the direction of a trend when the trend is steep enough. The momentum M is computed: $M=\left(p_r / p_{r-l}\right)/l$, where l is the number of time steps used in the calculation, p_r is the most recent historical price, and p_{r-l} is the price l time steps before p_r. A chartist's trading rules are:

If the number of my previous trades is less than the maximum allowed, then:

- *Buy* if $M \ge p_r(1+T)$ and the trader does not hold a long position
- *Sell* if $M \le 0$ and the trader holds a short position *Short* if $B \ge U$ and a short position is not held
- *Short* if $M \le p_r(1-T)$ and the trader does not hold a short position
- *Cover* if $M \ge 0$ and the trader holds a short position

where p_r is the most recent historical price and T is the threshold (5%=0.05). Buying, selling, shorting, and covering each count as one trade. Characteristically, chartists appear to reinforce trends and enhance both volatility and waves.

Noise Traders. Noise traders either buy or sell with equal probability in each time step. Noise traders are meant to reflect the apparent randomness in the real markets and act to move the price action around, in effect "tripping" the strategies of the other traders.

Whoops Trader. The whoops trader places one 10,000-stock order to buy at the market at 10:15am which represents the alleged trading mistake that happened on September 20, 2002 at 10:10am. While the trading mistake is reported to have occurred at 10:10am [7], orders on the London Stock Exchange may take many minutes before they are observed by other traders who then may take minutes more to act on that information. Because our synthetic traders immediately see information and react just as quickly, the simulation time of the mistaken order is adjusted to 10:15am.

Each simulation begins with the market maker quoting a best bid at \$3820, best ask at \$3821. All the market maker's orders are in lots of 100, and each simulation is run for the duration of the real dataset – from 9:00am to 11:30am in 1-minute increments.

3. Interactive Evolution

The IEC search method employs a genetic algorithm and a graphical user interface to facilitate the user acting as the fitness function. A small initial population of agent based models is generated with random parameter values. The resulting price histories are generated by running the models, and then shown to the human observer. The observer selects interesting patterns according to whatever objective and subjective criteria the observer may be using to visually compare candidates with the target. These selections are considered the fittest individuals of the generation. The user configures a set of operators - elitism, mutation, and crossover – which are used to produce a new generation of models from the user-selected "fittest" individuals in the previous generation. The new generation is then simulated and the resulting price histories are again presented to the observer. This procedure is iterated until interesting patterns emerge from the search that more closely match the target. Over multiple generations the population may converge toward the target.

The user interface is a critical component of the method, which depends crucially upon the user's ability to evaluate visualizations of the candidate solutions [9] –obviously this method can only work if the population size is kept small and if interesting patterns emerge after a reasonably small number of generations. The user interface shows the price histories of each candidate model in the left window, each overlaid on top of the target price history. On the right are two controls panels: (1) the main IEC control panel configures the size of the left-hand visualization grid, toggles operators, controls the chance that a gene is mutated, and controls the proportion of crossover versus mutation; the evolve button produces the next generation; (2) the second control panel controls model drawing.

Genotype. Each model simulation and its price history can be considered a phenotype, generated from the set of trading strategies and their parameters used in the simulation (the genotype). The parameters of these strategies vary across genotype, and it is the composition of these genotypes that we are interactively evolving. The genotype of a model has genes that are numbers – initially random numbers. Each number codes for a specific aspect of the set of traders (how many of each kind of trader, parameters of each strategy) and has bounds specific to the aspect it codes for. Genes are numbers that code for random variables – either how many of one kind of trader (the random variable is a constant in this case) or the α's and β's for a Beta distribution. Random values drawn from the Beta distribution serve as specific parameters to traders. In this way, only α and β describe a distribution of values used to parameterize any number of traders, compressing the genotype. The Beta probability distribution has non-zero values and takes the form: $p(x) = \left(\Gamma(\alpha+\beta)/\Gamma(\alpha)\Gamma(\beta)\right)x^{\alpha-1}(1-x)^{\beta-1}$, where x is real number in the interval **[**0,1**]**, $p(x)$ is the probability at x, and Γ(n) is Euler's gamma function. The Beta distribution was selected because it is a bounded distribution that can take many shapes, it is nicely parameterized by just α and β, and it is easily scaled to cover each variable's legal bounds. Note also that a beta distribution with α=1 and β=1 is equivalent to a uniform distribution. Alphas and betas are each bounded between 1 and 100, and each parameter has an upper and lower bound for values drawn from this distribution which scales the corresponding beta distribution. Constants, α's, and β's comprise a genotype, and the parameter values drawn from these random variables and their bounds define the search space. The bounds and distributions for the parameters are defined as follows:

Fundamentalists: # of traders is a constant between 10 and 50; Trade size, Reaction time, Moving average length and threshold percentage are all drawn from Beta distributions within bounds [100,500], [1,50], [20,100] and [0%,5%], respectively.

Chartists: # of traders is a constant between 10 and 100; Trade size, Maximum # of trades, Momentum look back and Momentum threshold percentage are all drawn from Beta distributions within bounds [100,1000], [1,200], [1,5] and [0.1%,1%], respectively.

Noise traders: # of traders is a constant between 1 and 10; Trade size is drawn from a Beta distribution within bounds [100,200].

Selection. Candidates are either selected or not, and non-selected candidates are discarded. The user may turn on and off each of three operators to produce next generation members - elitism, mutation, and crossover. Elitism is applied first; then crossover and mutation are performed on randomly chosen selected members to produce remaining members. The number of new members produced by crossover versus the number produced by mutation is a proportion chosen by the user.

Elitism. Selected candidates are copied to the new generation.

Mutation. The user controls the chance that each gene is mutated - between 0% and 100%. As the mutation algorithm iterates through each gene, a random number between 0.0 and 1.0 is chosen. If that number is less than the mutation percentage (expressed as a decimal, 50% is 0.50), that gene is mutated to a new value within the bounds of that variable. Mutation occurs on each gene if rand<chance, where rand is a random number in the interval [0,1] and chance is the user-selected chance that a gene is mutated. New values for a given gene are chosen as follows: v_n=*min+rand*(max-min)*, where v_n is the new value, *min* is the lower bound for this gene, *max* is the upper bound for this gene, and *rand* is a random number in the interval [0,1].

Crossover. Double-point crossover chooses crossover points in a uniform random fashion and produces a new candidate by recombining two parent genotypes from among the selected candidates.

4. Experiment and Results

As an example application, we use the market frenzy that occurred on September 20, 2002 at the London Stock Exchange and lasted for about 20 minutes [1]. The event began at 10:10 am and within 5 minutes the FTSE100 index rose from 3,860 to 4,060. Within another few minutes, the index fell to 3,755, before returning to a value slightly above its original level at the end of the 20 minutes. In that context, our formulation of the problem we wish to address is: what is the nature of the initial trigger and what are the likely behaviors of the different players that can generate such a deviation from the market's stationary behavior? Our model will assume that one large order is being submitted, producing the necessary trigger to push the market out of its stationary state "without affecting its fundamental dynamical parameters", following Muchnik and Solomon's plausible assumption [7]. The goal of IEC here is to determine the nature of the herding behavior that could produce the event. We were able to qualitatively match the target price history in an average of 10 generations, from a variety of starting populations. While "qualitative match" is an imprecise term, the characteristics we attempted to match are the amplitude, period, phase, and damping rate of the approximate wave – and of course the size, shape, and location of the price history. We matched reasonably well on each of these dimensions (Figure 1).

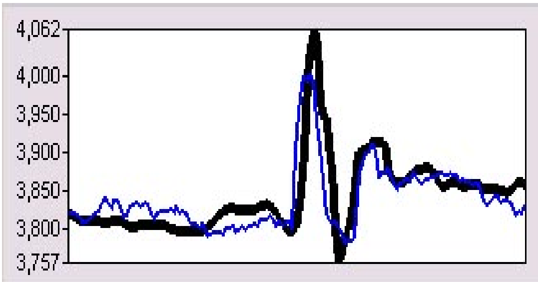

Fig. 1: Best solution with market frenzy data to be fitted (thick curve).

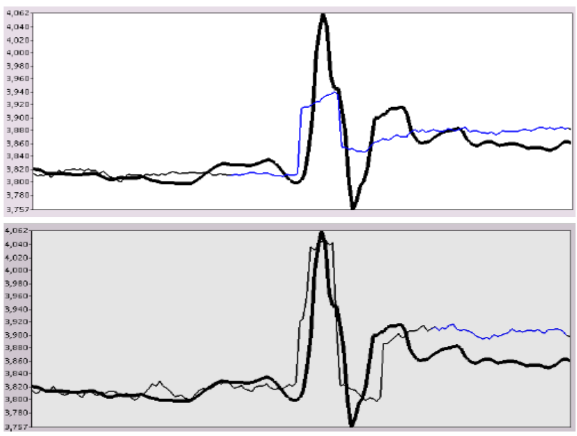

Fig. 2: Best solution with data to be fitted (thick curve). Top: GA best solution. Bottom: IE best solution.

One of the first questions one may ask about the example described in the previous sections is, "why can't a straight genetic algorithm come up with similar results in a short amount of time?" In order to apply a GA with an automated rather than user-defined fitness function, one has to define one such fitness function. The easiest and most natural one here is simply the inverse of a fitting error function $E=\left(N^{-1}\sum_{t=0}^{N}(p_t-a_t)^2\right)^{1/2}$, where p_t and a_t represent the model-predicted and actual price at time t, respectively. We used a simple GA, where the probability to be selected for each mutation or crossover operation is proportional to $1/(E+0.0001)$. Figure 2 shows the fittest phenotype the GA found after 50 generations (top) together with a good match found with IEC. Although the IEC and GA solutions have similar E values, he phenotype discovered by the GA does not satisfy the requirements of the experiment, namely it does not match the amplitude and shape of the first wave and dampens so fast as to almost not exhibit the second wave. This result is not entirely surprising since these qualitative features were not built into the error function E and that is why GA could not "see" them but a human could. We repeated the experiment with GA more than 10 times with different initial conditions but could not get a good match to the true price curve. E is not a good measure of performance in this experiment.

5. Conclusion

We have shown with a simple example how IEC can be used for ABM estimation. IEC nicely combines human expertise with evolutionary computation and improves the speed and accuracy of search in the fields of model inversion and design, particularly when goal evaluation is multi-variate, complex, or qualitative. A GA with a least-mean squares objective function does a reasonable job but IEC enables the user to estimate models without having to define a problem-specific, *ad hoc* objective function.

6. References

[1] Ball, P. (2002) Stock market shock explained. Physicists model recent trading frenzy. Nature Science Update,

[2] Boschetti, F., Moresi, L. 2001. Interactive inversion in geosciences. *Geophysics* 66, 1226-1234.

[3] Dawkins. R. 1987. *The Blind Watchmaker*. W. W. Norton, New York.

[4] Goldberg, D. E. 1989. Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing.

[5] LeBaron, B. (2001) A builder's guide to agent-based financial markets. *Quantitative Finance* **1**, 254-261

[6] Lux, T. & Marchesi, M. (1999). Scaling and criticality in a stochastic multi-agent model of a financial market. *Nature* **397**, 498-500

[7] Muchnik, L. & Solomon, S. (2003). Statistical mechanics of conventional traders may lead to non-conventional market behavior. *Physica Scripta* **T106**, 41-47.

[8] Sims, K. 1991. Artificial evolution for computer graphics. *Computer Graphics* **25**: 319-328.

[9] Takagi, H. 2001. Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. *Proc. IEEE* **89**: 1275-1296.