Learning Financial Agent based Simulator: Simulated Annealing as an Optimizer for Simulation Parameters

Filippo Neri

Dept. of Computer Science, University of Naples, 80125 Naples, Italy
filipponeri@yahoo.com

Abstract. Integrating agent based modeling with machine learning results in a promising methodology to model the behavior of financial markets. We report in this paper an experimental study of our learning system L-FABS showing how it can acquire models for financial time series.

Keywords: Agent based modeling, simulated annealing, financial markets, prediction of the SP500 and DJIA time series.

1 Introduction

Financial markets can be viewed as complex systems whose basic entities and interactions can be easily described. However no theory or method is able to explain or predict with certainty what will happen during their future evolution. The state-of-the-art literature shows that agent based modeling (ABM) is a promising methodology for simulating several types of domains, when interpreted as complex systems, like, for instance, consumer markets, economies or societies [1–7]. Thus ABM could be a promising candidate for investigating the behavior of financial markets as well. For researchers active in the ABM community, computational simulation takes the form of agent based simulators where hypothesis about the decision making process of the individual and about the relationships occurring among them could be tested. One of the main difficulties encountered by these works stays in tuning the model so that the simulated behavior approximate the observed one. With this respect machine learning algorithms, and simulated annealing [8] in the case of our research, can provide a useful tool to learn the main parameters governing the simulation process [5, 9].

In this paper, we comments some of the results obtained by our Learning Financial Agent Based Simulator L-FABS, that has been described in [5, 9], showing that it can approximate the time series of the SP500 and DJIA indexes under a variety of experimental setups.

2 The learning simulator L-FABS

The abstraction of financial markets that we used to design L-FABS [5, 9] takes into account that the behavior of the investors is driven by two main factors:

a) the propensity to take some investment risks today, by buying a financial asset, in exchange for a future uncertain reward, when selling the asset; and
b) the common consensus (the market sentiment or just sentiment) about the future behavior of the market itself. If the people believe that the economic outlook will be negative, each individual will tend to sell some of her/his assets.

The basic component of L-FABS is the Financial Agent Simulator or FAS. FAS is made up of a number of investor-agents that functionally reproduce the decision making process of human investors by deciding, round by round, if to buy additional assets, sell some or stay put. FAS manages the time period to be simulated by imposing turns over the many investor-agents acting in the system. Each investor-agents decides what to do on the base of a probabilistic process influenced by its risk-reward rate. Learning in L-FABS consists of finding the vector of risk/reward propensity rates that approximates a given time series with a minimum error. This learning setting allows for combining FAS, the simulation engine, with any of the many machine learning algorithms able to find a vector of values that minimizes a given error function. Examples of suitable machine learning algorithms include genetic algorithms [10, 11], decision trees [12], neural networks, simulated annealing [8] and many others. In our study, we decided to use simulated annealing because probabilistic search methods proved to be robust and well performing across several domains [8, 10, 11] and an individual oriented method is less computationally expensive than a population oriented one. Then L-FABS consists of running Simulated Annealing to search the vector space of the risk/reward propensity rates in order to find one that minimizes the error function.

3 Empirical analysis

For the empirical evaluation of L-FABS, we selected some financial time series to work with. As usual with learning systems, we will train L-FABS on a part of the dataset, the learning set, and then we will use the remaining part of the dataset as test set to assess the performances of the learned model. The selected datasets are:
Dataset 1 - learning set: SP500 from 3 Jan 1994 to 17 Dec 2003 and test set: SP500 from 18 Dec 2003 to 23 Oct 2006.
Dataset 2 - learning set: DJIA from 3 Jan 1994 to 17 Dec 2003 and test set: DJIA from 18 Dec 2003 to 23 Oct 2006.
All the datasets contain the daily closing values of the indexes and have been freely acquired from the finance section of yahoo.com. The reported results are averaged over 10 runs of the same experimental setting and the shown forecast errors are measured on test sets. In Table 1, the performances of L-FABS are shown when run on the Datasets 1 and 2. The columns in the table stand for: "Day to predict" indicates the number of days ahead for which a prediction of the time series is made, "Sentiment" indicates if the Sentiment index is calculated with modality S1 (taking into account the previous day close) or S5 (taking into account the close values of the previous 5 days), and, finally, the measured errors

on the test set are reported in terms of the MAPE (Mean Absolute Percentage Error) and Standard Error. The Sentiment value captures the public short term expectation about the market behavior. The findings suggests that the model learned by L-FABS has captured the intrinsic dynamics of the target time series. Moreover, from Table 1, it appears that the predictions of the next day values are more accurate than the predictions made for the seven days ahead values. This result confirms the intuitive experience that the farther a prediction is moved into the future, the less accurate it will be.

Table 1: Experimental findings on Datasets 1 and 2 relative to different periods of the SP500 and DJIA time series.

Experimental results on Dataset 1

Day to predict	Sentiment	MAPE	StdErr
1	S1	0.76	16.55
1	S5	0.70	14.31
7	S1	1.46	25.16
7	S5	1.42	24.29

Experimental results on Dataset 2

Day to predict	Sentiment	MAPE	StdErr
1	S1	0.76	136.52
1	S5	0.74	131.73
7	S1	1.48	215.26
7	S5	1.50	221.83

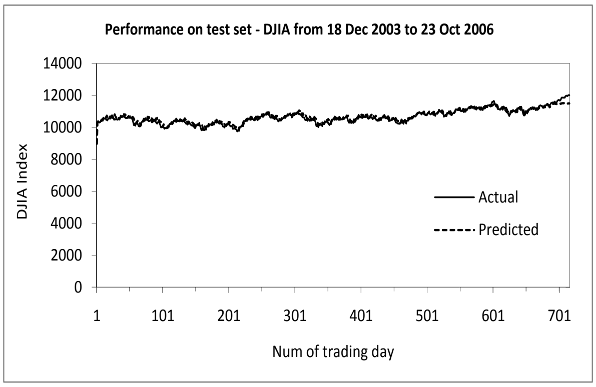

Fig. 1: The actual and predicted DJIA time series when testing L-FABS on Dataset 2 with settings: S1, Day to predict: 1.

For the sake of completeness, we also show the graphs of two time series as predicted by L-FABS in fig. 1 which is an example of the results obtained over several runs of L-FABS[1]. The predicted time series are compared with the actual ones. As it can be seen from the graphs, the solid line (actual) and the

[1] Note for a reviewer: at the end of the graph there is a divergence between the predicted value and estimated value because the agents-investors have reached their maximum investment capability. In this case, there is no possibility for the system to

dotted line (predicted) are very close confirming the error figures that have been reported in the tables.

4 Conclusions

We have briefly reported an experimental study about our system LFABS, which uses machine learning and agent based simulations to model financial time series, and we also reported an empirical study to investigating the behavior of L-FABS when modeling market indexes. We believe that our experimental findings support the research hypothesis that agent based simulation together with machine learning could result in a methodology fit to model financial time series.

References

1. Epstein, J.M., Axtell, R.: Growing artificial societies: social science from the bottom up. The Brookings Institution, Washington, DC, USA (1996)
2. Axelrod, R.: Agent-based modeling as a bridge between disciplines. In Judd, K.L., Tesfatsion, L., eds.: Handbook of Computational Economics, Vol. 2: Agent-Based Computational Economics, Handbooks in Economics Series. North-Holland, Amsterdam, The Nederlands (2006) 1562–1583
3. Bruun, C.: The economy as an agent-based whole simulating schumpeterian dynamics. Industry and Innovation **10** (2003)
4. Tesfatsion, L.: Agent-based computational economics: Growing economies from the bottom up. Artif. Life **8** (2002) 55–82
5. Neri, F.: Learning and predicting financial time series by combining natural computation and agent simulation. In Di Chio et al., ed.: EvoApplications (2). Volume LNCS 6625. Springer (2011) 111–119
6. Neri, F.: Learning and predicting financial time series by combining evolutionary computation and agent simulation. Transactions on Computational Collective Intelligence. Nguyen N. (editor in chief), Springer Berlin / Heidelberg **6** (2012) 202–221
7. Niola, V., Neri, F., Mastorakis, N., et al., eds.: New Aspects of Automatic Control, Modelling and Simulation. Europment Press (Sofia, Bulgaria) (2010)
8. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220** (1983) 671–680
9. Neri, F.: How to investigate the decision making behavior of investors in financial markets by means of software agents. In: 12th WSEAS International Conference on Automatic Control, Modelling and Simulation, Catania, Italy, Europment Press (2010) 118–124
10. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Ma (1989)
11. Neri, F.: Traffic packet based intrusion detection: decision trees and generic based learning evaluation. WSEAS Transaction on Computers **4** (2005) 1017–1024
12. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, California (1993)

---

predict an higher value for the index. The maximum investment capability is fixed before starting the systems is proportional to the total amount of money available.