DATA PROCESSING IN THE IMITATIONAL AGENT-BASED MODEL OF FINANCIAL MARKET

D.V. Senko

Belarusian State University, Faculty of Applied Mathematics and Computer Science Institute for Mathematical Modelling and Data Analysis

Abstract. Knowledge of dynamic properties of processes, that take place in finance is important in applications for derivatives pricing and risk control. The widely used Black and Scholes Theory assumes geometric Brownian motion of underlying asset price. However, an attempt to provide a model, which better describes price movements often leads to impossibility of studying the results analytically. Approach, which uses imitational models to study the dynamics in a complex system is a promising one. This paper presents the imitational model of the financial market, which is agent based, and provides a posibility to apply pattern recognition algorithms to build and calibrate a decision making support system in the financial market.

Stochastic models in finance

The basis of the widely studied and used Black and Scholes Theory is the assumption, that
movements of market prices can be described by Geometric Brownian Motion

$$\frac{ds(t)}{s(t)} = \mu dt + \sigma dW(t), \qquad (1)$$

where $W(t)$ – Brownian motion.

This model was a breakthrough in the financial analysis and engineering of derivatives, and was studied thoroughly. Several improvements and generalization of this process were proposed, and studied. The key parameters in this model are drift and volatility. The compound theory of derivatives pricing bases on this formula.

Imitational micromodel

The derivation of Random Walk formula applied to financial time series is based on the assumption of market efficiency. Random Walk means that price increments at different times $\Delta s(t),...,\Delta s(t-n)$ are independent and identically distributed random variables. Many derivative pricing and portfolio selection problems, can be solved analytically. It's simplicity is also the reason of the model's drawbacks. The attempts to make the model more suitable for real data result in mathematical equations, that can not be simplified and solved analytically. For example more real is the case, that decisions of market participants depend on the past price development. This is the case, because many of them use technical analysis for their decision making.

$$\Delta s(t) = f(\omega, s(t-1),..., s(t-n)) \qquad (2)$$

Analytical form of f is unknown, and is better described by the pattern selection and matching process in technical analysis. If historical price movement matches technical pattern, which forecasts price increase, then $\Delta s(t) > 0$ is the more likely outcome.

To better study and understand the underlying market dynamics the imitational model is proposed, which implements the price calculation algorithm, based on the processing of many orders from various market participants. The participants are modelled to make their decisions, basing on the historic market movements, they search for the predefined patterns, and make their investment decisions accordingly. The model is implemented in java programming language, which is chosen because of the simplicity to write big projects and reuse the code.

Model of market agent

Agent is capable of investing money into one product, or he can have money as cash. He is described by the state vector $(p^i(t), c^i(t))$ and parameters (λ, C), where $p^i(t)$ - number of products, he possesses $p^i(t) \geq 0$, $c^i(t)$ - cash available for further investments, $p^i(0) = 0, c^i(0) = 0$, $C>0$ – credit line: $c^i(t) \geq -C$. At times $(t_1^i, t_2^i, ...)$, with $\Delta t_k^i = t_{k+1}^i - t_k^i$, modelled with exponential distribution with parameter λ, agent A_i is active. He analyzes the Matrix of historical prices:

$P(t) = \begin{bmatrix} t_1 .. t_n \\ s_1 .. s_n \end{bmatrix}$, where s_i is product price at time t_i, $t_i < t$, $(t_1, ..., t_n) = \bigcup_{i=1}^{N} (t_1^i, ..., t_k^i)$, so, that $t_i \leq t_{i+1}$. His analysis of this matrix should result into a decision to do nothing or into an order $o = (a, p_{\min}, p_{\max}, t_{start}, t_{end})$, which numerically reflects his willingness to buy or sell one product. The order should only be performed, if price lays between $(p_{\min}, p_{\max})$, and date restrictions are satisfied.

Agent tries to recognise a defined pattern in price movements by using correlation coeficient in integral form. The integrals are calculated numerically.

$$r = \frac{\int_0^T (f(t) - a)(g(t) - b)dt}{\sqrt{\int_0^T (f(t) - a)^2 dt \int_0^T (g(t) - b)^2 dt}}, \tag{3}$$

where $a = \int_0^T f(t)dt$ и $b = \int_0^T g(t)dt$, $f(t)$ - price, $g(t)$ - pattern. The pattern can be stretched in accordance with studied time frame.

Price calculation

The order is then conducted into orderbook which is cleared according to the algorithm described in and price is calculated. The algorithm is restricted to basic ordertype "Limit Order" with time and price limitations.

Orders are placed and stored in the orderbook $(b_1, .., b_n, s_1, .., s_m)$, before they can be matched, *b* means buy orders, *s* – sell orders. Reference price is the price of the last deal.

When new buy order b_{n+1} arrives, algorithm searches $\{s_i\}$ and chooses an order which complies with conditions:

1. time and price compliance:

   $S' = \{s \in \{s_i\} \mid (p^b_{\min}, p^b_{\max}) \cap (p^s_{\min}, p^s_{\max}) \neq \phi$,

   and $(t^b_{start}, t^b_{end}) \cap (t^s_{start}, t^s_{end}) \neq \phi\}$,

2. price priority $p^s_{\min} = \min_{s \in S'} p^s_{\min}$,
3. if there are several of those, time priority is essential

Matched price is derived from the condition

$$p_{new} = \underset{p \in (p^b_{\min}, p^b_{\max}) \cap (p^s_{\min}, p^s_{\max})}{\arg\min} |p - p_{old}| \tag{4}$$

Computational experiments

Following 3 experiments were conducted:

1. N_A=0, N_R=200
2. N_A=6, N_R=194
3. N_A=9, N_R=191

where N_R - number of agents, taking random decisions, N_A- number of agents, taking decisions in accordance with technical pattern recognition on figure 1.

Fig. 1 Pattern for recognition

This pattern represents slowing of price increase, with future possible trend reversal. In the computational experiments several parameters can be varied, λ - agent activity, τ - average time of order validity, number of agents of different types, agent credit line and product restrictions, time frame for analysis, algorithm for risk control. When changing λ for each agent in same proportion, process is stretched along time axis. When increasing τ chart becomes smoother. Figure 2 shows the results of interactions of Random Agents.

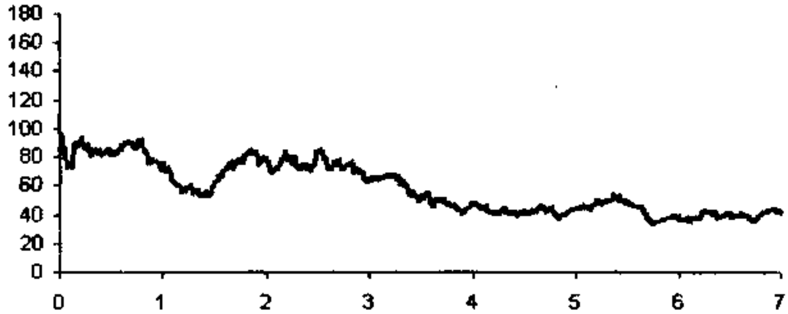

Fig. 2 Price movements with random agents. N_R =200, λ =1/240, τ = 0.0005

When adding some small number of deterministic agents A price movements contain intervals, where quick price movements take place Fig 3. These intervals correspond to activity times of agents, that take same decisions.

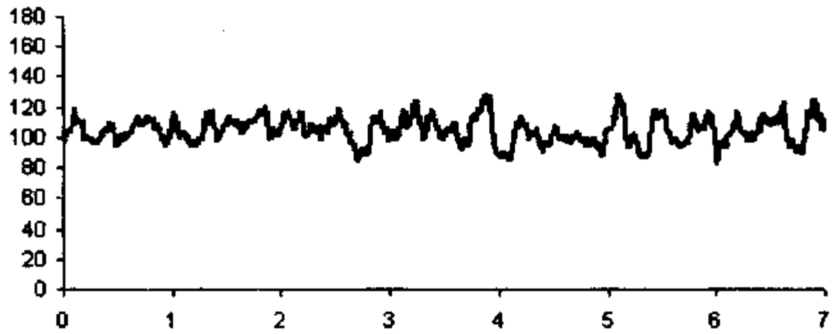

Fig. 3. Price movements, with small number of deterministic agents
$N_A = 6,\ \lambda = 1/240,\ \tau = 0.005,\ l = 0.5;\ N_R = 194,\ \lambda = 1/240,\ \tau = 0.0005$

Bigger number of deterministic agents result in bigger price oscillations, whereas the period comparable with time frame.

References

[1] Bak P., Paczuski M., Shubik M. Price variations in a stock market with many agents// Physica. - 1997. - A 246. - P. 430-453.
[2] Balsara N. Money Management Strategies for Futures and Options Traders. - John Wiley & Sons Inc., 1992. - 256 p.
[3] Gourieroux C., Monfort A. Simulation-Based Econometric Methods. - Oxford University Press, 1996. - 174 p.
[4] Paul W., Baschnagel J. Stochastic Processes: From Physics to Finance. - Berlin: Springer, 1999. - 231 p.
[5] Rosenberg M.R. Currency forecasting: methods and models for predicting exchange rate movements. - McGraw-Hill, 1996. - 388 p.
[6] Sandmann K. Einfuehrung in die Stochastik der Finanzmaerkte. - Berlin: Springer, 2001. - 522 p.
[7] Xetra ® Marktmodell Aktien Release 4.0 Herausgeber: Deutsche Boerse Frankfurt am Main 2000. - 51 S.