Using Fractal's Addition and GP for N-Steps Ahead Forecasting

Chao-Fu Hong[1] Chi-Liang Yang[2]

[1]Department of Information Management, Aletheia University
[2]Graduate School of Management Sciences, Aletheia University

Abstract

In the past decade there have been many researchers who have used the artificial intelligence and chaos theory to identify the time series to forecast the stock price. However, they have had a common problem: could not identify the limit of prediction. In this paper, we adopt the multi-acquiring cognizance as follows: the Genetic Programming (GP) is used to acquire the individual meaningful fractal from chaotic time series and the Run test is used to examine the residuary error. When the test result is not close to the randomness, we will run acquiring process again. In this paper, we compare the results of two models - firstly, our model with multi-acquiring and addition fractal algorithm in it, and secondly a simple GP model with embedding theory in it. Our model's SSE is about 246, but the simple GP's SSE is about 1252. It could prove our model has a good predictability. Finally, we extend our model to the n-steps ahead predicting, and the predictive precision is also the same as the one-step ahead.

Keywords: predictability, chaos, self-similarity, fractal, genetic programming.

1. Introduction

Many studies have been done on non-linear properties of the stock market data. Most studies have been tended to incorporate chaotic component of a time series Eric [1], Iba [4], Kaboudan [5][6], Kumar and Tan [7], Tan [11] and tended to use the continuative time Iba [3], Lee et al. [8], Neely and Weller [9][10] to improve a model's forecasting ability.

Kumar and Tan[7] investigated the performance of predictability on the statistic method and the artificial intelligence, the result indicated that the artificial intelligence had high precision. In addition, they also weaved embedding theory into ANN and the experimental results showed that the embedding theory could increase the predictive precision. From these results we find that if the chaos theory could identify the non-random time series, then it can increase the predicting precision. But, they did not explain what the predictive error was.

Many researchers run n-steps ahead predicting based on the chaotic trajectory. However, the n-steps ahead predicting is easily cumulate the predictive error; it will let the prediction become more difficult. Therefore, how to decrease predictive error and what the predictive error becomes an important problem.

2. Methodology

Because the fractal has the self-similarity effect, we can depend on the partial fractal and iterating with self-similarity to reconstruct the whole fractal. In this study, we adopt the backward thinking. When we have serial partial fractals, we integrate these values to construct the whole fractal to run the τ-steps ahead predicting. Our predictive model has three parts as follows:

Phase 0: Decompose a chaotic time series to τ Partial Fractal time series

Our issue is that run the τ-steps ahead predicting, it must have τ partial fractal time series to generate the τ forecasting functions and integrate every predicting value to reach the τ-steps ahead predicting. Therefore, we have to use the embedding theory first, the algorithm developed by Gautama et al. [2] to find the optimal period (delay-time τ) and create the τ partial fractal time series as formula (1).

$$X_{t+1} = f_1(x_t, x_{t-\tau}, x_{t-2\tau}, \ldots, x_{t-(\tau)\tau})$$
$$X_{(t+1)+1} = f_2(x_{(t+1)}, x_{(t+1)-\tau}, x_{(t+1)-2\tau}, \ldots, x_{(t+\tau-1)-(\tau)\tau})$$
$$\vdots$$
$$X_{(t+\tau-1)+1} = f_\tau(x_{(t+\tau-1)}, x_{(t+\tau-1)-\tau}, x_{(t+\tau-1)-2\tau}, \ldots, x_{(t+\tau-1)-(\tau)\tau}) \quad \ldots.. (1)$$

Phase 1: Use the Multi-Acquiring and Additional Fractal Algorithm to improve the Predictive Ability

From the FFT analysis (Fig.5) we can find that the chaotic time series is built by many independent wave vectors only to run the embedding analysis once; the optimal delay time maybe only have a ability to catch the mostly strong wave (major fractal), and it is reasonable that the major fractal is not completely equal to the whole fractal. From the view spot of backward thinking, substrate the major fractal from whole fractal is the residuary fractal. Then, the major fractal adds residuary fractal can increase the predictive ability. These processes will follow the formula (2) and Fig.1.

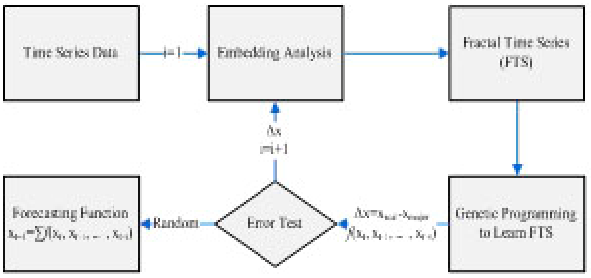

Fig. 1: Multi-acquiring and additional fractal algorithm

1st step: The genetic programming (GP) and sliding window (window length is τ) is used to learn the fractal and create the forecasting function. Then, these predictive values are able to build a new time series (x_{major}) and the x_{real} is the raw time series.

$$\Delta x = x_{real} - x_{major}$$
$$\Delta x_{t+1} = f_{\Delta}(\Delta x_t, \Delta x_{t-1}, \Delta x_{t-2}, ..., \Delta x_{t-\tau'})$$
$$x_{t+1} = f_{major}(x_t, x_{t-1}, x_{t-2}, ..., x_{t-\tau}) + f_{\Delta}(\Delta x_t, \Delta x_{t-1}, \Delta x_{t-2}, ..., \Delta x_{t-\tau'}) \quad ..... (2)$$

2nd step: follow the formula (2). Firstly, calculate the residuary error to get the error time series and that is used to training GP for constructing the residuary forecasting function. The final predictive value is the f_{major} add f_{Δ} like formula (2). This process will be continued until the residuary fractal can not sketch out any meaningful pattern. Then, the process will be stopped. It also assumes that the final predictive error may be as the white noise.

Phase 2: Analyze the Final Predictive Error

Two kinds of analysis methods are used to examine the final predictive error: one is the fractal analysis and the other is the Run test. Because the random time series has no any relation with neighboring data like formula (3) and its fractal will be a clutter as Fig.4, we can judge the error time series is the randomness or not. The other is the Run test, when the test result is high it also means the error time series is the randomness. These results are wanted to prove the final predictive error is randomness.

$$\Delta x_{t+1} \neq f_{\Delta}(\Delta x_t, \Delta x_{t-1}, \Delta x_{t-2}, ..., \Delta x_{t-\tau'}) \quad .. .. (3)$$

3. Experiment Results

The D-Link is our target stock, the experimental data is from TSEC (Taiwan Stock Exchange Corp.), and the period is from 01/10/2001 to 12/31/2003. The total data is 741 points, the top 590 points is used to training and remainder 151 points is for test. All data is the raw data.

Phase 0: Generate the Partial Fractal time series

The Gautama et al.'s [2] algorithm was used to analyze the delay time, the delay time was 8(τ=8). Following formula (1) to arrange training data and test data that the system could get 8 partial fractal time series. The similarity test's results are shown on table 1, their have good similarity, but every time series also has a unique part about 10%. This is why the simple embedding method is hardly getting good predictive precision.

Table. 1: the Similarity between every partial fractal

	2	3	4	5	6	7	8
1	.99	.99	.92	.90	.97	.90	.90
2		.99	.92	.90	.97	.90	.90
3			.92	.90	.97	.90	.90
4				.98	.94	.98	.98
5					.93	.99	1
6						.93	.93
7							.99

The experiment of **Phase 1**: the Multi-Acquiring and Additional Fractal Algorithm

1st step: Collected all predictive values to build a new time series (x_{major}), its phase diagram is shown in Fig. 2, the yellow and red mark are the predictive data (x_{major}) and the blue mark is raw data(x_{real}). From the graph their fractals are similar, but the result of Run test is 0.533. It is not very agreeable that the error time series is randomness, move to the step 2 to increase its predictability.

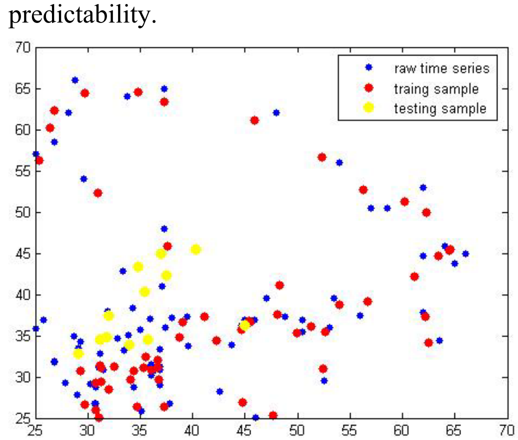

Fig. 2: simple GP's Phase Diagram.

2nd step: calculated the residuary error(Δx) and this error time series was also used to training GP for constructing the residuary forecasting function(f_Δ). Compare Fig.3 with Fig.2, it is clear that our model's fractal (add the residuary fractal to the major fractal) is very close the raw fractal, and from Fig. 6 our model's predicting error is about 2%. The FFT's result is shown in Fig.5. It also proves that our model is better than simple GP forecasting. But, the fractal of error time series is sketched in Fig.4. The result of the Run test is 0.849; it is very close to the randomness, and then stops the multi-acquiring and additional fractal process.

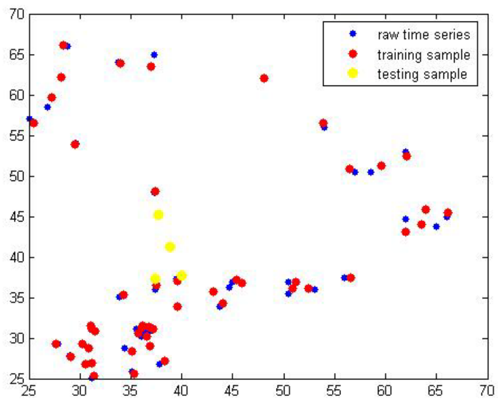

Fig. 3: Our model's Phase Diagram.

The experiment of **Phase 2**: Analyze the Final Predictive Error

Compare the final error's fractal (red mark) with simple GP's fractal (blue mark) in Fig.4, we can find that the error's fractal of our model is close to the clutter, and the Run test also proves this time series to be the randomness. These mean that our system is hardly to acquire information from the predictive error time series.

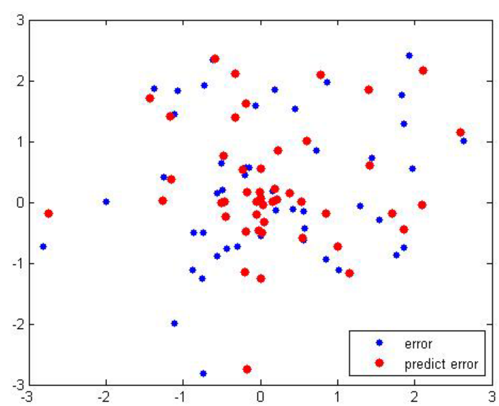

Fig. 4: the blue mark is primary error and red mark is residuary error.

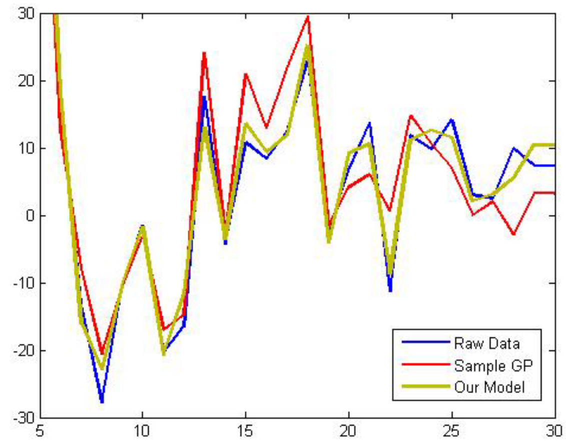

Fig. 5: FFT results belong to the raw data, simple GP and our model.

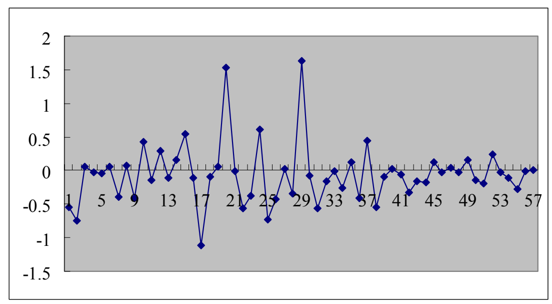

Fig. 6: Our model's predictive error.

The Kaboudan [5] had defined the predictability:

$$\eta = 100 * (1 - \overline{SSE}_Y \div \overline{SSE}_s),$$

$$\overline{SSE}_Y = k^{-1} * SSE_Y$$

Where k is the number of GP runs performed in search for the fittest equation, and $\overline{SSE}_S$ is computed similar to $\overline{SSE}_Y$ but for the shuffled data.

Following his forecasting method, we run 5 times to calculate the SSE_Y and$\eta$and the experimental results are shown on table 2. Their predictive precision is 98%, but the SSE_Y is above 1000. Compare our model (table 3) with simple GP's model (table 2), the $\eta$is the same, but the our model's SSE_Y is about 246 it small than the simple GP's SSE_Y. This result also can say our model has good predictability.

Table. 2: the Simple GP's SSE.

	SSE_Y	SSE_S	η
Run 1	1258.96	73851.44	0.983
Run 2	1303.52	78245.12	0.983
Run 3	1378.48	77587.36	0.982
Run 4	1331.6	80523.44	0.983
Run 5	988.64	79465.76	0.988

Table. 3: Our model's SSE

Partial Fractal	SSE_Y	SSE_S	η
1	11.28	8847.46	0.999
2	14.76	10456.83	0.999
3	34.56	11154	0.997
4	6.2	10677.5	0.999
5	28.66	11041.6	0.996
6	119.02	7813.63	0.985
7	14.34	10990.41	0.999
8	17.6	9941.01	0.998
total	246.42	80922.44	0.996955

4. Conclusions

In this study we have three issues. Firstly, the predicting values are generated by every partial forecasting function and integrate these values, the system could do the one to τ-step ahead forecasting. Secondly, our additional fractal method can build the complete partial fractal and increase the predictive precision to about 98%. But when we integrate every partial predictive error the whole system's SSE is 246, but the simple GP's SSE is 1252. It also indicates the forecasting performance of our model is very good. Finally, the residuary error is as the white noise. Because the multi-acquiring and additional fractal method can pick up all meaningful partial fractal, and the residuary error use the chaos analysis to draw its phase diagram as shown on Fig.4. It is as the clutter, and the Run test is 0.849. According to these results we deduce the residuary error is near the white noise.

5. References

[1] R. W. Eric, and M. B. John, "Evolving artificial neural networks to control chaotic systems", The American Physical Society, Vol. 56, No. 2, pp. 1531-1540, 1997.

[2] T. Gautama, D.P. Mandic, and M.M. Van Hulle, "A differential entropy based method for determining the optimal embedding parameters of a signal", Proceedings of the International Conference on Acoustics, Speech and Signal Processing, ICASSP 2003, pp. 29-32, 2003.

[3] H. Iba and T. Sasaki, "Using genetic programming to predict financial data", Proceedings of the Congress on Evolutionary Computation, vol. 1, pp. 244-251, 1999.

[4] H. Iba and N. I. Nikolaev, "Financial Data Prediction by Means of Genetic Programming", Proceedings of the Sixth International Conference on Computing in Economics and Finance, CEF2000, 2000.

[5] M. Kaboudan, "A measure of time series predictability using genetic programming applied to stock returns", Journal of Forecasting 18: 345-357.

[6] M. Kaboudan, "A GP approach to distinguish chaotic from noisy signals", Genetic Programming 1998: Proceedings of the Third Annual Conference, San Francisco. CA: Morgan Kaufmann, pp. 187-192, 1998.

[7] K. Kumar, C. Tan, and R. Ghosh, "Using chaotic components and ANN for forecasting exchange rates in foreign currency markets", Proceeding of the Queensland Finance Conference 1999, Brisbane, Queensland, Australia, September 30 – October 1, 1999.

[8] D. G. Lee, B. W. Lee, and S. H. Chang, "Genetic Programming model for long-term forecasting of electric power demand", Electric Power Systems Research, Vol. 40, pp.17-22, 1997.

[9] C. J. Neely, and P. A. Weller, "Predicting Exchange Rate Volatility: Genetic Programming Versus GARCH and RiskMetrics", Federal Reserve Bank of St. Louis Review, pp.43-54, 2002.

[10] C. J. Neely, and P. A. Weller, "Using a Genetic Program to Predict Exchange Rate Volatility", Genetic Algorithms and Genetic Programming in Computational Finance, edited by S. H. Chen, Kluwer Academic Publishers, pp. 263-279, 2002.

[11] C. Tan, "A Hybrid Financial Trading System Incorporating Chaos Theory, Statistical and Artificial Intelligence/Soft Computing Methods", at the Queensland Finance Conference 1999 – Computational Finance, Queensland, Australia, September 30 – October 1, 1999.