Predicting Stock Returns with Genetic Programming: Do the Short-Term Nonlinear Regularities Exist?

Shu-Heng Chen
Department of Economics
National Chengchi University
Taipei, Taiwan, 11623
R.O.C
E-mail: NCCUT171@TWNMOE10.BITNET

Chia-Hsuan Yeh
Department of Economics
National Chengchi University
Taipei, Taiwan, 11623
R.O.C
E-mail: NCCUS449@TWNMOE10.BITNET

October, 1994

ABSTRACT

This paper is devoted to applying the genetic programming paradigm to the test of the captial market efficiency hypothesis. How this paradigm is distinguished from the existing statistical approaches is briefly reviewed. Instead of using the large-sample analysis prevailing in the literature, this research rests on a small-sample analysis to inquire the existence of short-term non-linear regularities. By Rissanen's MDLP (Minimum Description Length Principle), the sample period with the highest complexity is chosen. Since our simulation results, which are based on Koza's genetic programming paradigm (KGP) and its Bayesian modification (BGP), show that it is not easy to outperform AR(1) and is extremely difficult to beat random walk, the nonlinear regularities, while might exist, is very difficult to be found. Therefore, the capital market efficiency hypothesis can, at least, sustain from this perspective.

1 Rethinking Predictability via Genetic Programming

The purpose of this paper is to apply the *genetic programming paradigm* developed by Holland(1975) and Koza(1992) to testing the *capital market efficiency hypothesis* by using the data of Taiwan's stock market. Traditionally, the test of this hypothesis is based on the concept of *probabilistic independence*, that is, to prove that the σ-algebra generated by the history of the rates of return will tell us nothing about the present or future rates of return. Therefore, technically, the rate of return R_t should be independent of any Borel functions of R_s ($s < t$)[1]. However, the major problem behind this idea is that there is no way we can effectively construct the evidence of independence by trying *all* Borel functions of R_t. So, in practice, only limited sets of function are included as the candidates, and the test gradually proceeds from *linear independence* to *non-linear independence*.

[1] In the literature, this is the so-called *weak-form efficiency*.

In this paper, we shall show that genetic programming provides us with a new framework to revisit this issue. While it agrees with the definition of market efficiency based on probabilistic independence in some essential aspects such as the unpredictability of the future rates of return, genetic programming captures the meaning of ***unpredictability*** in a more natural way, i.e., instead of asking whether or not the future rates of return are predictable, the genetic programming (GP) approach asks how difficult it is to predict; instead of asking whether the capital market is efficient, GP asks how efficient or inefficient the capital market is. Thus, the genetic programming approach transforms an intractable (and undecidable) yes-or-no issue into a more-or-less one. The reasons why genetic programming enables us to do so are stated as follows.

The intuitive meaning of *hard to predict* or *very hard to predict* can be considered equivalent to *(very) hard to find a rule under intensive search from past experiences which can help predict the future.* However, a fruitless intensive search does not imply there is no rule, nor does it mean that it is difficult to find one given the fact that it exists. Rather, it depends on how the search is implemented. Blind random search might not be qualified to decide whether the rules are hard to find. Highly organized search has its problems too, because to implement organized search, we must know something about the world which we have not started to explore yet. Fruitless intensive search can evidence "hard-to-predictness" only if the implementation of search is neither too random nor too organized. But it is very difficult to pinpoint the balance point. The relevant concern here is not *"Is the search structure optimal?"* but rather *"Is the search structure acceptable?"*[2]. This is where genetic programming comes into play.

By following Darwin's evolutionary principle and the operations of reproduction, crossover and mutation, GP can be considered an effective search principle. First of all, it makes a compromise between blind random search and organized selective search. Initially (Generation 0) , it can start from a totally random search. Then, generation after generation, the operation of reproduction and crossover based on the fitness criterion makes the search more selective and organized[3]. Secondly, it starts from simple forecasting rules and, unless necessary, the chance of jumping into complex forecasting rules is rare. In the spirit of Occam's razor or the information theoretic *Minimum Description Length Criterion*[4], this is certainly a very desirable feature. Thus, GP can effectively[5] find better forecasting rules. This explains why GP can serve as an effective search structure and as a foundation upon which one may judge whether it is hard to predict the rates of return.

In the literature of financial economics, the efficient market hypothesis can be translated as

$$E[R_t \mid \Sigma_{t-1}] = 0 \tag{1}$$

where Σ_{t-1} is the $\sigma-algebra$ generated by all past publicly available information. One of the tests of this hypothesis is based on the model

$$\begin{aligned} E[R_t \mid \Sigma_{t-1}] &= g(R_{t-1}, \cdots, R_{t-k}) \\ &= \beta_0 + \sum_{i=1}^{k} \beta_i R_{t-i} + \phi(R_{t-1}, \cdots, R_{t-k}) \end{aligned} \tag{2}$$

[2]This is especially true under the influence of Godel's incompleteness theorem in mathematical logic.

[3]It shares some similar features with the simulated annealing in the numerical analysis.

[4]See Rissanen (1989).

[5]I.e., in the sense of both time complexity and algorithmic complexity. However, the mathematics that underlies GP is not easy. Some work has been done by Holland(1986), but we need more contribution in this aspect.

with the null hypothesis

$$H_0 : \beta_i = 0, \quad i = 0, 1, \cdots, k \tag{3}$$

and

$$\phi(R_{t-1}, \cdots, R_{t-k}) = 0 \tag{4}$$

where $\phi()$ is a nonlinear Borel function of $R_{t-1}, \cdots, R_{t-k}$. While it is easy to test (3) by restricting the model to a linear version of Model (2), a general test of both (3) and (4) under Model (2) is difficult. We therefore use the following two-stage procedure to perform this test.

At the first stage, we use genetic programming to find the possible functional form g, i.e., $\hat{g}(.)$, by using the SSE (sum of squared errors) as the criterion of fitness. This enables us to know whether there is any simple nonlinear function which can fit the data better than the linear autoregressive models (LARMs). However, by doing so, we might run the risk of overfitting. Therefore, At the second stage, we use $\hat{g}(.)$ to forecast and to see whether it can forecast better than the LARMs. Only in the case where $\hat{g}$ can both fit and forecast better, do we conclude that a nonlinear relationship exists.

Before we can proceed further, however, a few questions need to be addressed. Firstly, since the search directed by genetic programming is *random*, what matters is the *probability* of finding models better than LARMs. As is clearly shown in Chen and Yeh (1994a), genetic programming can be mathematized as a random search model parameterized by first-order Markov transition probabilities which are determined by the design of the *evolution operators*[6]. Hence, given a target, a particular LARM for instance, we can ask the probability, namely, Π, of finding the *forest*[7] which includes at least one *tree* whose performance in terms of both fitting and forecasting is superior to that of the target[8]. We propose this Π as an objective measure of the difficulty of prediction. The higher the Π, the easier it is to predict. Unfortunately, a direct calculation of Π is extremely difficult. Nevertheless, an estimated $\hat{\Pi}$ can be obtained by a large-scale simulation.

2 Nonlinearity, Complexity and the Choice of Data

The second issue concerns the choice of the data set, especially the sample size. While the efficient market hypothesis puts no restriction on the sample size, the application of GP to different sizes of sample does require lots of thought. This is because GP aims at finding the potential existence of nonlinear regularities. The requirement for the sample size varies with different periodicities. For the time-invariant long-term nonlinear relation, a large sample size is needed. However, if the stock market encounters a sequence of short-term time-variant nonlinear relations, a large sample size may average out all these relations. In this case, a smaller sample size is desirable. Thus, lots of combinations need to be checked before any mature conclusion can be reached. Since this is the very beginning of our research on the efficient market hypothesis, we would like to see some preliminary results as soon as possible, and the choice of small sample

[6]This can be done by using the language LISP (List Programming) to encode functions.

[7]A forest is a collection of trees. Since in the language of LISP, each function can be represented by a tree, a forest is, in fact, a collection of functions or models.

[8]More precisely, Π is a function of the length of evolution taken, i.e., the number of generations, n, and should be denoted by $\Pi(n)$.

size is justified by this consideration. In this paper, the sample size is set to be 55, and the data are divided into the in-sample period and out-sample period in accordance with the following ratio[9].

$$\frac{p}{q} = \frac{\#\{in - sample\ period\}}{\#\{out - sample\ period\}} = 10 \qquad (5)$$

Furthermore, we use Rissanen's MDLP (minimun description length principle) criterion to pick out the most complex 55 observations[10] as our data set. A detailed description of this procedure and its meaning can be found in Chen and Tan (1994)[11]. By this criterion, the Taiwan stock market index used is during the period from 11/27/90 to 2/5/91[12]. The in-sample period is from 11/27/90 to 1/30/91 and the out-of-sample from 1/31/91 to 2/5/91.

3 The Emprical Results of Koza's Genetic Programming

To implement genetic programming, the program GP-Pascal is written in terms of Pascal 4.0 by following the instruction given in Koza (1992)[13]. The chosen parameters to run GP-Pascal are given in Table 1. To derived $\tilde{\pi}$, 72 simulations were run under Table 1.

Table 1: Tableau for Simulation 1a-72a and 1b-50b

Population size	500
The number of tree created by complete growth	50
The number of tree created by partial growth	50
Functional set	$\{+, -, \times, sin, cos, \%, EXP, RLOG\}$
Terminal set	$\{R_{t-1}, R_{t-2}, \cdots, R_{t-10}\}$
The number of tree generated by reproduction	50
The number of new lives	50
The number of trees generated by mutation	100
The probability of mutation	0.2
The maximum length of tree	17
The probability of leaf selection under crossover	0.5
The number of generations	100
The maximum number in the domain of Exp	1700

For each of the simulation, the sum of squared errors (SSE) is calculated for the in-sample period and the sum of squared prediction errors (SSPE) for the out-of-sample period under Gen (generation)=0, 50, 100, 150 and 200[14]. We

[9] Since we only consider the capability of GP to learn the possible existence of the short-term nonlinearity, it is natural not to test its performance by using too many out-sample observations.

[10] I.e., the sample period with the highest MDL.

[11] Briefly speaking, we first transform the orginal sequence of $\{R_t\}$ from 1/5/71 to 1/27/94 into a 0-and-1 sequence based on the sign of R_t. Then MDL is computed for each of the 50 consecutive observations in the 0-and-1 sequence by choosing the Bernoulli class and Markov class as our model classes.

[12] The MDL for this period is 37.807. The lowest MDL which is 12.126 is observed in the period form 5/23/85 to 6/1/85.

[13] A detailed description of this program can be found in Chen, Lin and Yeh (1994).

[14] For details, we refer to Chen and Yeh (1994b).

then compared these results with those derived from the best model chosen from LARMs. The LARMs we consider are composed of Model AR(1) to AR(10) and random walks. The best model chosen by the AICC criterion is AR(1) whose SSE and SSPE are 0.06726 and 0.00331 respectively[15] and the results of $\hat{\Pi(n)}$ are given in Table 2.

Table 2: The Estimated $\Pi(n)$ Given That the Target Is AR(1)

n	$\hat{\Pi}(n)$	The simulation which beats AR(1)
50	0.3750	4, 6, 8,12,14,16,17,21,24,25,26,27,29,30,31,33,37,41,48,49,51,53,55,57,58,62,72
100	0.2083	4,10,16,27,29,31,41,45,47,48,49,51,52,53,64
150	0.1806	4,10,15,27,31,35,41,42,45,49,51,52,64
200	0.1528	4, 6,10,13,27,31,41,42,45,49,63

In terms of both *SSE* and *SSPE*, the chance that GP can beat AR(1) is less than one half; hence, generally speaking, GP does not perform better than AR(1). This is especially true when we allow the evolution to take longer. This may reflect that overfitting will become a serious problem when evolution takes too long. The third column of Table 2 lists all the simulations that beat AR(1). There are five simulations which perform consistently better than AR(1), namely, Simulations 4, 27, 31, 41, and 49. The best model selected from these five simulations are written as Equations (6) to (10).

Simulation 4a :

$$\begin{aligned} R_t &= ((LogR_{t-4} * (R_{t-1} * R_{t-5}))\%(((ExpR_{t-4} * (R_{t-10}\%R_{t-8})) \\ &\quad -Log(R_{t-5} + R_{t-1})) * (R_{t-10} + (R_{t-7}\%(R_{t-10} + R_{t-8}))))) \end{aligned} \tag{6}$$

Simulation 27a :

$$\begin{aligned} R_t &= (Sin(R_{t-9} * R_{t-5})\%((((Sin6.51634822 - R_{t-4}) + R_{t-7}) \\ &\quad -R_{t-1}) + SinR_{t-3})) \end{aligned} \tag{7}$$

Simulation 31a :

$$R_t = (R_{t-8}\%Log((R_{t-6} * R_{t-6})\%R_{t-9})) \tag{8}$$

Simulation 41a :

$$\begin{aligned} R_t &= (LogLogLog(R_{t-10}\%R_{t-9}) * (LogR_{t-1} * (ExpLog(R_{t-2} + \\ &\quad R_{t-6})\%((R_{t-9}\%R_{t-9})\%R_{t-3})))) \end{aligned} \tag{9}$$

Simulation 49a :

$$\begin{aligned} R_t &= (Log((Log((R_{t-10} * (R_{t-9} - R_{t-10})) * R_{t-7}) * ((Log((R_{t-1} * \\ &\quad (R_{t-10} * R_{t-4})) * R_{t-7}) * (R_{t-6} * R_{t-8})) * R_{t-6})) * R_{t-6}) * \\ &\quad (R_{t-4} * R_{t-2})) \end{aligned} \tag{10}$$

In order to understand whether the model selected by GP can outperform random walks (RWs), we now turn to Table 3. It can be seen that the chance that GP beats RWs is extremely low. While Simulations 4 and 41 can perform consistently better than RWs, the difference is very limited[16].

[15]The SSE and SSPE for *random walks* are 0.0068 and 0.0027.

[16]See Figure 1 and 2.

Table 3: The Estimated $\Pi(n)$ Given That the Target Is Random Walks

n	$\hat{\Pi}(n)$	The simulation which beats Random Walks
50	0.0416	4,16,41
100	0.0555	4,16,41,45
150	0.0555	4,10,41,51
200	0.0555	4,10,13,41,

4 The Emprical Results of Bayesian Genetic Programming

In our second experiment, we consider a modified version of genetic programming, i.e., *Bayesian genetic programming* (BGP). BGP modifies the original version of genetic programming by adding our prior knowledge to Generation 0. By doing this, we are asking whether GP can enable us to forecast better given prior knowledge. In this experiment, we include all LARMs, i.e., from AR(1) to AR(10), into Generation 0. 50 simulations were run under the same chosen parameters given in Table 1 and the results are summarized in Tables 4 and 5.

Tabel 4 is the average SSE over all simulations under KGP and BGP. We can see that adding LARMs does enhance our performance of learning from experience. On average, the SSE has been reduced from 0.055 to 0.046 at Gen=200. Unfortunately, the nonliear regularities learned in this way have very poor capability of being generalized. From Table 5, we can see that with LARMs as the initial knowledge, the forecasting performance of GP deteriorates. Only Simulation 11 consistently performs better than AR(1), and none of the simulations forecast better than random walks.

Table 4: The Comparision of Fitness Performance between KGP and BGP

n	KGP's SSE	BGP's SSE
50	0.059	0.053
100	0.057	0.050
150	0.056	0.048
200	0.055	0.046

Table 5: The Estimated $\Pi(n)$ Given That the Target Is AR(1):BGP

n	$\hat{\Pi}(n)$	The simulation which beats AR(1)
50	0.02	11
100	0.04	11, 40
150	0.02	11
200	0.02	11

5 Concluding Remarks: Complex Turning Points and Long-Term Regularity

The application of genetic programming to the most complex 50-day period of Taiwan's stock index shows that, while the short-term nonliearities might exist, it is extremely difficult to find them. It is in this sense that the weak-form efficiecncy hypothesis is accepted. However, to generalize this result, two questions need to be addressed. First, will the same result hold for other 50-day periods whose turning point pattern is less complex. Second, are there any long-term nonlinearities in stock returns series? These issues merit further studies.

References

[1] Chen, S., C. Lin and C. Yeh (1994), "On the Model Selection and its Stability of the Natural Vancancy Rate of Housing: An Application of Genetic Programming", Paper presented in the *2nd Conference on Inductrial Economics*, Tamkang University, Taipei.

[2] Chen, S. and C. Yeh (1994a), "Genetic Programming and Technology-Oriednted Forecasting", in C. C. Yuan (eds.), *Proceedings of the 1994 National Conference on the Management of Technology*, National Chiao Tung University, Hsin Chu. pp. 377-391.

[3] Chen, S. and C. Yeh (1994b), "Predicting Stock Returns with Genetic Programming: Do the Short-Term Regularities Exist?", mimeo, Department of Economics, National Chengchi University.

[4] Chen, S and C, Tan (1994), "On the Stochastic Complexity of Taiwan Stock Returns", mimeo, Department of Economics, National Chengchi University.

[5] Holland, John H (1975), *Adaptation in Natural and Artificial Systems.* University of Michigan Press.

[6] Holland (1986), "A Mathematical Framework for Studying Learning in Classifier Systems", *Physica 22D*, pp.307–317.

[7] Koza, J. R. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* The MIT Press.

[8] Rissanen, J.(1989), *Stochastic Complexity savein Statistical Inquiry.* World Science, Singapore.