
On Predictability and Profitability: Would GP Induced Trading Rules be Sensitive to the Observed Entropy of Time Series?

Nicolas Navet^{1,2} and Shu-Heng Chen¹

¹ INRIA Lorraine, Campus-Scientifique, BP239, F-54506 Vandoeuvre, France, nnavet@loria.fr

² AI-ECON Research Center, Department of Economics, National Chengchi University, Taipei, Taiwan 11623, chchen@nccu.edu.tw

Summary. The entropy rate of a dynamic process measures the uncertainty that remains in the next information produced by the process given complete knowledge of the past. It is thus a natural measure of the difficulty faced in predicting the evolution of the process. The first question investigated here is whether stock price time series exhibit temporal dependencies that can be measured through entropy estimates. Then we study the extent to which the return of GP-induced financial trading rules is correlated with the entropy rates of the price time series. Experiments are conducted on EOD data of the stocks making up the NYSE US 100 index during the period 2000-2006, with genetic programming being used to induce the trading rules.

Key words: Entropy estimate, surrogate testing, genetic programming, financial trading rules, NYSE.

1 Introduction

One fundamental issue which remains unclear for both financial econometricians and financial engineers is the relationship between the predictability and profitability of financial time series. The literature, so far, has reached no conclusion with regard to the proposition that a time series is profitable if it is predictable, and vice versa. While this proposition may sound obvious, it is not. What makes it subtle is that the two groups of researchers have employed different approaches to tackle the financial time series, and have caused predictability and profitability to be two separate entities. Financial econometricians are more concerned with predictability. For this purpose, formal statistical or information-theoretic approaches are applied to measure the predictability of financial time series. On the other hand, financial engineers or financial practitioners are more concerned with profitability. For that purpose,

various heuristic trading algorithms have been used in an attempt to make proper market-timing decisions. There seems to have been a series of efforts made recently to bridge the gap between the two, but it is far from enough; more often than not what we see is that these two groups in the literature have developed without referring to or conforming to each other.

This chapter, therefore, purports to shed some light on this issue by further cross-referencing the empirical evidence. In other words, we shall connect the empirical results on predictability more closely with the empirical results on profitability. The approach taken by us is, first, to determine the predictability of some sampled financial time series, and, second, to gauge the profitability of time series with different predictability. To do so, we need to choose one principal measure of predictability as well as a trading algorithm. For the former, we choose an information-based measure, i.e., an entropy-based measure, and for the latter we use genetic programming to induce trading algorithms.

Using entropy to measure the degree of randomness and the predictability of a series has a long history, that goes back almost to the very beginning of the development of communication and information theory. Its significance has been introduced to economists since the 1960s. In Section 2, we shall give a brief review of the entropy measure and the associated estimator (Section 2.1) used in this paper. This set-up enables us to determine the degree of predictability of any time series coming later. In Section 2.2, the reasonable behavior (performance) of this proposed measure (estimated entropy) is further illustrated with pseudo random series and financial time series. However, to show that there is no unique measure of predictability, in Section 3 we further compare the results of our entropy-based measure with those from a well-known nonlinear dependence test, namely, the Brock, Dechert and Scheinkman (BDS) test (Section 3.1), and the linear dependence test based on the familiar auto-correlation function (Section 3.2). The purpose is to show that there are some discrepancies existing among different measures of predictability, which may become another obstacle to successfully establishing the connection between predictability and profitability.

Using genetic programming to evolve trading rules has gradually become a part of the practice of financial investment [Smith(1998)]. In this chapter, we continue this trend and use genetic programming to exploit the potential profitable opportunities. We start Section 4 with a simple review of genetic programming. The idea of using genetic programming to test the profitability performance is first established in [Chen and Navet(2006), Chen and Navet(2007)], where the random trading rule, known as lottery trading, is first formulated as a benchmark. This chapter applies the same idea to gauge the profitability of different financial time series. The experimental designs and results are given in Sections 4.1 and 4.2, respectively, followed by the concluding remarks (Section 5).

2 Entropy estimation

Entropy estimation is a field of investigation that has been very active over the last 10 years, one of the reasons being the crucial practical importance of information-theoretic techniques in the advances of neuroscience and, in particular, in the understanding of how the brain works. Methods for estimating the entropy rate can be roughly classified in two main classes [Gao *et al.*(2006)]:

- “Plug-in” (or maximum-likelihood) estimators that basically consist of evaluating the empirical distribution of all words of fixed length in the data, for instance by constructing an n -th order Markov Chain, and calculating the entropy of its distribution. Unfortunately, the sample size that is needed increases exponentially in the length of the words and, in practice, plug-in methods are not well suited to capture medium or long range dependencies. In the context of financial time-series, we cannot rule out that there are medium or long range dependencies, after all this is the assumption underlying many trading strategies, and thus we choose to not measure entropy with an estimator belonging to that family.
- Estimators based on data compression algorithms, either estimators based on Lempel-Ziv (ZV, see [Farach *et al.*(1995)] and [Kontoyiannis *et al.*(1998)]) or the Context-Tree Weighting algorithm (see [Willems *et al.*(1995)] and [Kennel and Mees(2002)]). Both approaches have been shown [Gao *et al.*(2006), Gao *et al.*(2003)] to have fast convergence rates (i.e., they are accurate even with a limited amount of observations) and to be able to capture medium and long-range dependencies.

2.1 \hat{h}_{SM} entropy rate estimator

In this study, we use an estimator belonging to the Lempel-Ziv class that has been proposed in [Kontoyiannis *et al.*(1998)] (estimator a) from Theorem 1 in [Kontoyiannis *et al.*(1998)] - as in [Kennel *et al.*(2005)], it will be named \hat{h}_{SM} in the following). Let n be the size of time series s and s_i the symbol at location i in s , the \hat{h}_{SM} estimator is defined as:

$$\hat{h}_{SM} = \left(\frac{1}{n} \sum_{i=1}^n A_i \right)^{-1} \log_2 n \quad (1)$$

where A_i is the length of the shortest substring starting at position s_i that does not appear as a contiguous substring of the previous i symbols s_0, \dots, s_{i-1} .

This estimator, which is well known and often used in the literature (see, for instance, [Kennel *et al.*(2005)]), has been shown in [Kontoyiannis *et al.*(1998)] to have better statistical properties and performances than earlier Lempel-Ziv estimators. To get further confidence in

the efficiency of \hat{h}_{SM} , we measured the entropy rate of a sample made of independent draws of a uniform random variable P that takes its value in the set $\{1, 2, \dots, 8\}$. The theoretical entropy is equal to $H(P) = -\sum_{i=1}^8 (1/8) \log_2(1/8) = 3$. The entropy estimate depends on the size of the sample, the quality of the random number generator and the efficiency of the entropy estimator. Using \hat{h}_{SM} with a sample of size 10000, the entropy estimate is equal to 2.96 with the random generator from the boost C++ library [Maurer(2007)], which demonstrates the quality of the estimator since 3 is the best that can be obtained with a “perfect” random generator.

2.2 Entropy of NYSE US 100 stocks

Here we estimate the entropy of the daily price time series of the stocks that make up the NYSE US 100 index (the composition of the index can be found at url http://www.nyse.com/marketinfo/indexes/nyid_components.shtml). The data are processed so that the data points are the log ratios between consecutive daily closing prices: $r_t = \ln(p_t/p_{t-1})$ and points are then further discretized into 8 distinct states. The boundaries between states are chosen so that each state is assigned the same number of data points (“homogeneous” partitioning). This design choice has the advantage that the model is parameter free and thus no heuristic decision that may change the conclusion reached is required. Furthermore, this experimental setup proved to be very efficient at revealing the randomness of the original data, which is the main quality criterion for partition schemes [Steuer *et al.*(2001)].

The distribution of the entropy rate for the stocks of NYSE US 100 index between from 01/01/2000 to 31/12/2006 is shown in Figure 1. The minimum value is 2.68, the median 2.75, the mean 2.75 and the maximum value is 2.79. The time series have a high entropy since the theoretical upper bound is 3 and uniformly randomly generated samples achieve 2.90 with the same number of data points³. This is not very surprising per se since high entropy rates have been observed even for smaller time scales (see for instance [Lee *et al.*(2006)]). The 5 stocks from NYSE US 100 index with the highest entropy, identified by their symbol, are *OXY* (2.789), *VLO* (2.787), *MRO* (2.785), *BAX* (2.78), *WAG* (2.776) and the five stocks with the lowest entropy are *TWX* (2.677), *EMC* (2.694), *C* (2.712), *JPM* (2.716), *GE* (2.723). These 10 stocks will be considered in the experiments in the next sections.

Although the entropy is high, there is evidence that the original time series are not random. Indeed, we compare the entropy of the original time series with the entropy of randomly shuffled variants of the original data (surrogate testing). Precisely, 100 shuffled time series for each original time series (after the discretization step) are generated and their average entropy is measured.

³ A value of 2.90 is obtained using the boost C++ random generator, but with the standard *rand()* function from the *C* library, the entropy rate achieved is as low as 2.77, which is less than the entropy value of some stocks.

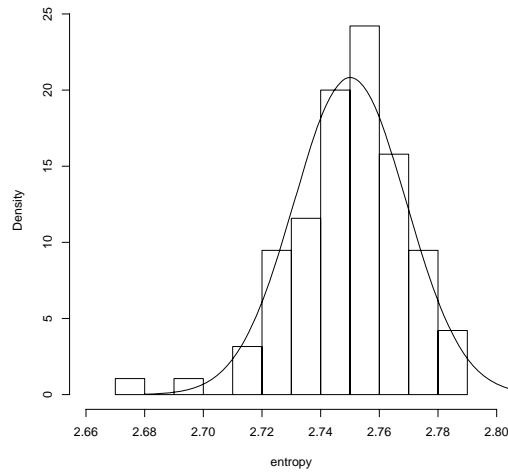


Fig. 1. Distribution of the entropy rate of the stocks that make up the NYSE US 100 index (log. ratios of consecutive daily closing prices). A normal distribution with the same mean and standard deviation is plotted for comparison. The reference period is 2000 – 2006.

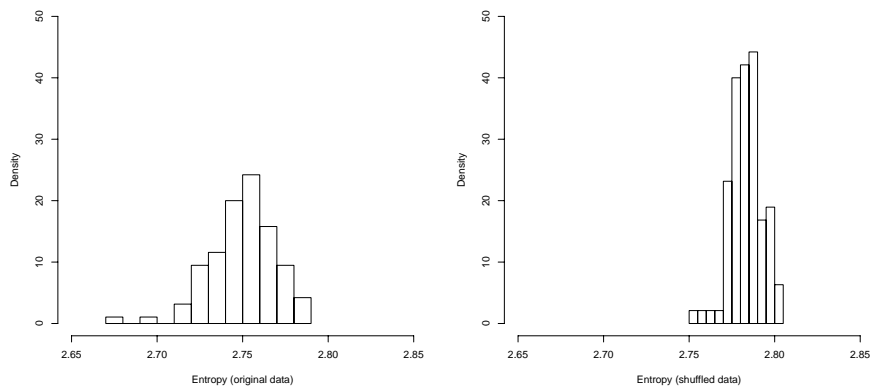


Fig. 2. Distribution of entropy rates of the original time series (left-hand graphics - rescaled with regard to Figure 1) and shuffled time series (right-hand graphics). The x-axis ranges from 2.65 to 2.85 on both graphics.

The complexity of the surrogate time series is greater (2.8 versus 2.75) with a lower standard deviation ($9 \cdot 10^{-3}$ versus $1.9 \cdot 10^{-2}$) and distributed differently as can be seen in Figure 2. This provides evidence that, at least for some stocks, there are (weak) temporal dependencies in the original time series.

3 Linear and nonlinear dependencies

Since some limited temporal dependences have been highlighted we now wish to estimate more precisely their extent through the Brock, Dechert and Scheinkman test (BDS), and to try to find out whether they are linear or nonlinear by performing an autocorrelation analysis.

3.1 BDS test statistics

We employ the BDS statistics (see [Brock *et al.*(1995)]) to test the null hypothesis that the daily log price changes are independent and identically distributed (i.i.d.). The BDS is a widely used and powerful test serving to identify departure from independence and identical distribution caused by non-stationarity, as well as linear and nonlinear dependencies. The test can be applied to raw data or residuals of an estimated model to test for omitted dynamics (see, for instance, [Caporale *et al.*(2004)]), and thus help to decide the model's relevance. The reader is referred to [Brock *et al.*(1995)] for the description of the test as well as an assessment of its power, and to [Hsieh(1993), Chu(2001)] for some applications to financial data.

The samples under study are sufficiently large for the test to be accurate (*i.e.*, with a sample size greater than 500, see [Hsieh(1993)]). The two parameters of importance are the “embedding dimension”, parameter m in what follows, and parameter δ , the maximal distance (expressed in terms of sample standard deviations) between points that are considered to be “close”. Both parameters are used in the test for the computation of the correlation integral, which is a measure of serial dependence originating from physics [Grassberger and Procaccia(1983)]. The parameter values used in the following analysis are classical in the BDS literature and conform to the recommendations given by the authors of the test.

Table 1 presents the BDS values of the daily log price changes between 01/01/2000 and 12/31/2006 for the highest entropy stocks, while Table 2 shows the BDS values for the lowest entropy stocks. The significance levels of the statistics are 1.645 (10%), 1.96 (5%) and 2.576 (1%). The first observation is that, regardless the stock and the BDS parameters, the null hypothesis that daily log price changes are i.i.d. should be rejected at the 1% significance level. This suggests that price time series are not stochastic i.i.d. processes.

What is striking however is that the values of the statistics are much larger for the lowest entropy stocks than for the highest. For a given set of parameters (e.g., $m = 2$ and $\delta = 0.5$), the smallest BDS value among the set

m	δ	<i>OXY</i>	<i>VLO</i>	<i>MRO</i>	<i>BAX</i>	<i>WAG</i>
2	0.5	5.68	4.23	6.60	7.38	6.78
3	0.5	6.40	5.38	9.52	11.17	7.79
5	0.5	9.86	7.18	13.62	17.73	9.94
2	1	5.66	4.17	6.69	8.13	7.45
3	1	6.61	5.35	9.40	11.11	8.89
5	1	9.04	6.88	13.08	15.31	11.17
2	1.5	5.34	4.01	6.17	8.33	7.63
3	1.5	6.53	5.34	8.75	10.46	9.52
5	1.5	8.66	6.86	12.08	13.30	11.55
2	2	4.81	3.48	5.32	7.29	6.82
3	2	6.08	4.80	7.97	8.57	9.00
5	2	8.24	6.39	10.86	10.32	10.64

Table 1. BDS test statistics of daily log price changes for the highest entropy stocks. All values are significant at the 1% level.

of lowest entropy stocks is larger than the highest value among the set of the highest entropy stocks. This shows that the departure from the i.i.d. property is much more important when the entropy is weaker, which conforms to what was expected, and provides evidence that the price dynamics are different between the stocks composing the two sets.

m	δ	<i>TWX</i>	<i>EMC</i>	<i>C</i>	<i>JPM</i>	<i>GE</i>
2	0.5	19.04	14.57	14.60	12.94	11.29
3	0.5	26.84	22.15	20.91	18.12	17.12
5	0.5	51.26	39.00	38.57	35.70	29.48
2	1	18.06	14.21	13.9	11.82	11.67
3	1	22.67	19.54	18.76	16.46	16.34
5	1	34.18	29.17	28.12	26.80	24.21
2	1.5	14.67	10.97	12.57	9.5	10.72
3	1.5	17.30	14.59	16.16	13.25	14.24
5	1.5	23.41	20.15	21.54	19.26	18.67
2	2	12.07	8.51	11.40	7.47	9.27
3	2	13.87	11.61	14.25	10.95	13.04
5	2	17.57	15.53	17.72	15.17	16.20

Table 2. BDS test statistics of daily log price changes for the lowest entropy stocks. All values are significant at the 1% level.

3.2 Autocorrelation analysis

The BDS test enables us to reject the null hypothesis that price changes are i.i.d. but it does not provide us with the precise cause of the rejection. In

particular, we would like to know whether the rejection is caused by linear or nonlinear dependencies in the time series because this would have implications in terms of forecasting and trading strategies. Here, to obtain some insight into this question, we analyze the autocorrelation of the daily log price changes.

Low Entropy Stocks		High Entropy Stocks	
<i>C</i>	8	<i>BAX</i>	2
<i>EMC</i>	5	<i>MRO</i>	1
<i>GE</i>	1	<i>OXY</i>	1
<i>JPM</i>	5	<i>VLO</i>	1
<i>TWX</i>	4	<i>WAG</i>	4

Fig. 3. Number of autocorrelation coefficients up to a lag of 100 that are significant at the 1% level.

What can be observed is that the autocorrelation of the log-returns is much larger for the lowest entropy stocks than for the highest entropy stocks. As shown in Table 3, up to a lag of 100, there are on average 4.6 autocorrelations that are significant at the 1% level for the lowest-entropy stocks versus 1.8 for the highest-entropy stocks. However, even for low-entropy stocks, the autocorrelation is very limited. Let us consider the CITIGROUP stock (symbol *C*) whose autocorrelation coefficients up to a lag 100 are shown in Figure 4. This stock has 8 autocorrelation coefficients that are significant at the 1% level, which is the highest count among all stocks under study. However, no

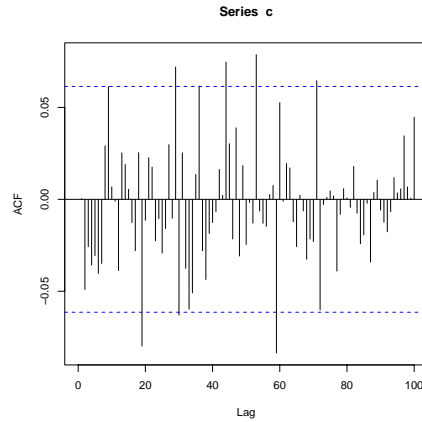


Fig. 4. Autocorrelation coefficients of the CITIGROUP stock (symbol *C*) daily log price changes up to a lag of 100. The points outside the region comprised of the area between the horizontal dotted lines are significant at the 1% level.

single coefficient is larger than 0.08, which is too weak to be of any value for the purpose of forecasting.

Given the limited serial correlation that can be found in the time series, we cannot conclude that the departure from i.i.d. is solely caused by linear dependencies, and, most likely, it should be explained by a combination of linear and nonlinear dependencies. The question addressed in the next section is whether GP is able to take advantage of these temporal dependencies, be they linear or nonlinear, and come up with profitable trading strategies.

4 Experiments with Genetic Programming

Genetic programming (GP) applies the idea of biological evolution to a society of computer programs. Specifically, in financial trading, each computer program represents a trading system - a decision rule - which when applied to the market provides trading recommendations. The society of computer programs evolves over the course of the successive generations until a termination criterion is fulfilled, usually a maximum number of generations or some property of the best individuals (e.g., stagnation for a certain number of generations, or a minimum performance threshold is reached). Classical genetic operators, namely, mutation, crossover and reproduction, are applied at each generation to a subset of individuals and the selection among the programs is biased towards the individuals that constitute the best solutions to the problem at hand. The reader may for instance refer to [Chen *et al.*(2007), Neely *et al.*(1997)] for GP applied to trading in foreign exchange markets, [Allen and Karjalainen(1999), Chen *et al.*(2007), Potvin *et al.*(2004)] in stock markets, [Wang(2000)] in futures markets and [Keber(1999), Chen *et al.*(1998)] for GP used for pricing options.

The aim of the experiments is to evaluate whether there is a link between the entropy of the time series and the profitability of the GP-induced trading rules. To assess its efficiency, GP is tested against a strategy that would consist of making the investment decision randomly (*“Lottery Trading”*). We follow the methodology proposed in [Chen and Navet(2006), Chen and Navet(2007)] and, in particular, we constrain the randomness so that the expected number of transactions for lottery trading is the same as for GP in order to allow a fair comparison. Hypothesis testing is performed with the Student’s t-test at a 95% confidence level.

4.1 Experimental setup

Experiments are conducted for the period 2000 – 2006, which is divided into three sections: the training (2000 – 2002), validation (2003 – 2004) and out-of-sample test periods (2005 – 2006). The trading rules are created by Genetic Programming on the training set, and a subset of top-performing rules are

further selected on the unseen data (validation set). The best rule on the validation set is then evaluated for the out-of-sample test period. As classically done in the literature in terms of data-preprocessing, data is normalized with a 100-day moving average. The individuals of GP are trading rules that decide when to enter a long position (no short selling allowed). Exits are decided by a maximum stop loss (-5%), a profit target stop (10%) and a 90-day stop (exit from a position that has been held for the last 90 days). The performance metric is the net profit, with a starting equity of \$100,000 and the size of each position equal to 100% of the current equity. The functions, terminals and parameters of the GP runs are described in Table 5.

Population size	1000
Number of generations	50
Maximum tree depth	7
Function set	+, -, *, /, norm, average, max, min, lag, and, or, not, >, <, if-then-else, true, false
Terminal set	daily closing prices, real and integer ephemeral constants
Value range for real constants	[-1,1]
Value range for integer constants	[0,1000]
Offsprings created by:	
crossover	50%
standard mutation	20%
swap mutation	15%
reproduction	10%
ephemeral constant mutation	5%
Initialization	ramp-half-and-half
Evolution scheme	generation replacement strategy
Elitism	10 best individuals are kept for the next generation
Selection scheme	tournament selection of size 3
Fitness function	accumulated return
Transaction costs	0.1%
Validation	
number of best trees saved	10 individuals per run are saved for validation

Fig. 5. GP control parameters

The price time series used in the experiments are shown in Figures 6 and 7. The first section of each graphic is the training period, the second section is the validation period and the last is the test period.

4.2 Experimental results

The outcomes of applying GP to the highest entropy stocks are shown in Table 3. What can be first observed is that GP always leads to profitable strategies but this is not very informative since the price time series of these stocks, except for *WAG*, exhibit a strong upward trend during the test period (see Figure 6) and short selling is not possible in these experiments. What is more significant is that GP outperforms Lottery trading (LT) 2 times out of the 5 experiments, while LT is never better than GP.

Table 4 shows the results of GP on the lowest entropy stocks. It turns out that GP is never better than LT while LT outperforms GP 2 times out of 5.

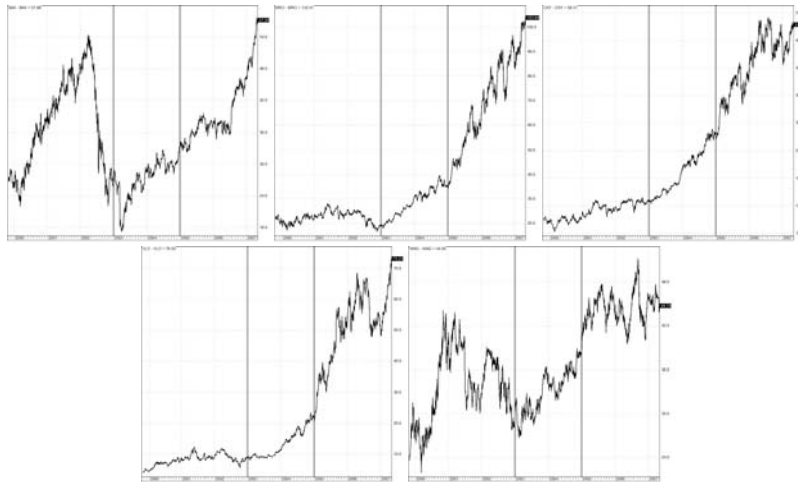


Fig. 6. Price time series of stocks having the highest entropies : *BAX*, *MRO*, *OXY*, *VLO*, *WAG*.

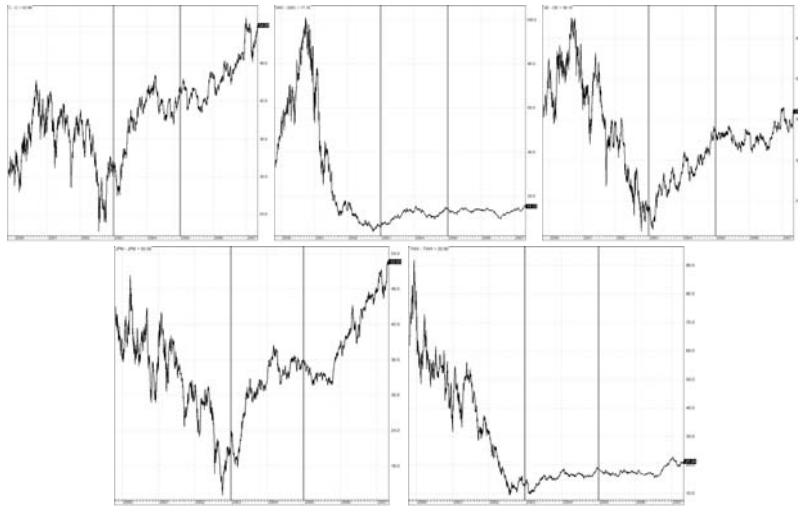


Fig. 7. Price time series of stocks having the lowest entropies : *C*, *EMC*, *GE*, *JPM*, *TWX*.

This suggests to us that the evolution process is not efficient here, and might even be detrimental.

From the results shown in Tables 3 and 4, it should be concluded that, with our experimental setup, selecting the stocks with the lowest entropy does not lead to a better profitability for the GP induced trading rules. We actually observe the opposite which can be explained, as highlighted

	GP net profits	LT net profits	GP>LT?	LT>GP?
<i>OXY</i>	15.5K\$	14K\$	No	No
<i>VLO</i>	7K\$	11.5K\$	No	No
<i>MRO</i>	15K\$	18.5K\$	No	No
<i>BAX</i>	24K\$	13K\$	Yes	No
<i>WAG</i>	6K\$	-0.5K\$	Yes	No

Table 3. Net return of GP and Lottery trading (LT) on the highest entropy stocks (rounded to the nearest 500\$). The first two columns are the average profit with GP (20 runs) and Lottery Trading (1000 runs). The third (resp. fourth) column indicates whether one should reject the hypothesis that GP (resp. LT) does not outperform LT (resp. GP) at the 95% confidence level.

	GP net profits	LT net profits	GP>LT?	LT>GP?
<i>TWX</i>	-9K\$	-1.5K\$	No	Yes
<i>EMC</i>	-16.5K\$	-11K\$	No	Yes
<i>C</i>	15K\$	18.5K\$	No	No
<i>JPM</i>	6K\$	10K\$	No	No
<i>GE</i>	-0.5K\$	0.5K\$	No	No

Table 4. Net return of GP and Lottery trading (LT) on the lowest entropy stocks (same settings as Table 3). Experiments conducted with the possibility of selling the stocks short do not show significant improvements.

in [Chen *et al.*(2007)], because GP is usually not efficient when the training interval exhibits a time series pattern which is significantly different from the out-of-sample period (e.g., “bull” versus “bear”, “sideways” versus “bull”, etc.). This is exactly what happens here for the lowest entropy stocks as can be seen in Figure 7.

On the contrary, in the two cases (*BAX* and *WAG*, see Figure 6) where the training period is very similar to the test period, GP clearly outperforms Lottery Trading. This suggests to us that improvements can be made by rethinking the data division scheme and coming up with criteria to select stocks that would integrate a measure of the dissimilarity between current and past market conditions. A contribution that might prove useful in that regard is given in [Struzik and Siebes(1999)] where the authors show that the Haar wavelet transform is suitable for providing estimates of similarity between time series.

5 Conclusion and Future Work

It has been shown that the EOD price time series of the NYSE U.S. 100 stocks do not all have equal entropies and, based on surrogate testing, that there are some weak temporal dependencies in the time series. The BDS statistics and the autocorrelation suggest that these temporal dependencies are both linear and nonlinear. The next step has been to test the hypothesis that selecting the stocks with the lowest entropy - the ones with the most predictable price time series - would lead to less risky investments. In the experiments, however, we did not observe that this hypothesis holds.

Recent studies (e.g., [Kennel *et al.*(2005), Gao *et al.*(2006)]) have shown that Context Tree Weighting (CTW) entropy estimators often lead to faster convergence rates than Lempel-Ziv-based estimators. Since the samples of daily data are small in size, the use of CTW may lead to some improvements, although what is really crucial here is not the precise entropy estimate but the relative ordering between distinct time series.

Here, the empirical evidence suggests that predictability is neither a necessary nor a sufficient condition for profitability. The predictability test only tells us about the existence of temporal patterns, but it does not give further information on how easy or difficult it is to discover the pattern. Therefore, predictability may not necessarily lead to profitability. On the other hand, we observed on two series with high entropy that it was possible to come up with efficient trading rules. As the large literature on the subject suggests, predictability has a multi-dimensional description, and only one measure of predictability may not be enough to capture all of its attributes. We think that further study regarding the relationship between predictability and profitability should not rest only upon a single measure.

In this study we limit ourselves to the stocks making up the NYSE US 100 because they are of primary interest to investors. The stocks are very liquid and have huge capitalizations (47% of the entire market capitalization of US companies). It is possible that the price time series of these stocks share many common structural characteristics, and so would not be good candidates for a selection technique based on entropy. Future experiments should include stocks of lower entropy that do not belong to the NYSE US 100, and other time scales should be considered. In particular, higher frequency data would enable us to study the variations in entropy over time.

Acknowledgement. We would like to thank the anonymous reviewers for their helpful suggestions. For the second author, the research support in the form of NSC grant No. NSC. 95-2415-H-004-002-MY3 is gratefully acknowledged.

References

[Allen and Karjalainen(1999)] Allen, F. and Karjalainen, R. (1999). Using genetic algorithms to find technical trading rules. *Journal of Financial Economics*, **51**,

- 245–271.
- [Brock *et al.*(1995)] Brock, W., Dechert, W., LeBaron, B., and Scheinkman, J. (1995). A test for independence based on the correlation dimension. Working papers 9520, Wisconsin Madison - Social Systems.
- [Caporale *et al.*(2004)] Caporale, G., Ntantamis, C., Pantelidis, T., and Pittis, N. (2004). The BDS test as a test for the adequacy of a GARCH(1,1) specification. a Monte Carlo study. *Institute for Advanced Studies*, **3**(2), 282–309.
- [Chen and Navet(2006)] Chen, S.-H. and Navet, N. (2006). Pretests for genetic-programming evolved trading programs: zero-intelligence strategies and lottery trading. In I. King, J. Wang, L. Chan, and D. L. Wang, editors, *Neural Information Processing, 13th International Conference, ICONIP 2006, Proceedings, Part III*, volume 4234 of *Lecture Notes in Computer Science*, pages 450–460, Hong Kong, China. Springer.
- [Chen and Navet(2007)] Chen, S.-H. and Navet, N. (2007). *Evolutionary Computation in Economics and Finance - Volume 2*, chapter Failure of Genetic-Programming Induced Trading Strategies: Distinguishing between Efficient Markets and Inefficient Algorithms. Springer.
- [Chen *et al.*(1998)] Chen, S.-H., Yeh, C.-H., and Lee, W.-C. (1998). Option pricing with genetic programming. In J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba, and R. Riolo, editors, *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 32–37, University of Wisconsin, Madison, Wisconsin, USA. Morgan Kaufmann.
- [Chen *et al.*(2007)] Chen, S.-H., Kuo, T.-W., and Hoi, K.-M. (2007). Genetic programming and financial trading: How much about "what we know". In C. Zopounidis, M. Doumpos, and P. M. Pardalos, editors, *Handbook of Financial Engineering*. Springer. Forthcoming.
- [Chu(2001)] Chu, P. (2001). Using BDS statistics to detect nonlinearity in time series. In *53rd session of the International Statistical Institute (ISI)*.
- [Farach *et al.*(1995)] Farach, M., Noordewier, M., Savari, S., Shepp, L., Wyner, A., and Ziv, J. (1995). On the entropy of DNA: algorithms and measurements based on memory and rapid convergence. In *SODA'95: Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 48–57, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.
- [Gao *et al.*(2003)] Gao, Y., Kontoyiannis, Y., and Bienenstock, E. (2003). Lempel-Ziv and CTW entropy estimators for spike trains. Slides presented at the NIPS03 Workshop on Estimation of entropy and information of undersampled probability distributions - Theory, algorithms, and applications to the neural code.
- [Gao *et al.*(2006)] Gao, Y., Kontoyiannis, I., and Bienenstock, E. (2006). From the entropy to the statistical structure of spike trains. In *2006 IEEE International Symposium on Information Theory*, pages 645–649.
- [Grassberger and Procaccia(1983)] Grassberger, P. and Procaccia, I. (1983). Measuring the strangeness of strange attractors. *Physica D*, **9**, 189–208.
- [Hsieh(1993)] Hsieh, D. (1993). Implications of nonlinear dynamics for financial risk management. *Journal of Financial and Quantitative Analysis*, **28**(1), 41–64.
- [Keber(1999)] Keber, C. (1999). Option pricing with the genetic programming approach. *Journal of Computational Intelligence*, **7**(6), 26–36.
- [Kennel and Mees(2002)] Kennel, M. B. and Mees, A. I. (2002). Context-tree modeling of observed symbolic dynamics. *Physical Review E*, **66**(5), 056209.

- [Kennel *et al.*(2005)] Kennel, M. B., Shlens, J. B., Abarbanel, H. D. I., and Chichilnisky, E. J. (2005). Estimating entropy rates with Bayesian confidence intervals. *Neural Computation*, **17**(7), 1531–1576.
- [Kontoyiannis *et al.*(1998)] Kontoyiannis, I., Algoet, P. H., Suhov, Y. M., and Wyner, A. J. (1998). Nonparametric entropy estimation for stationary processes and random fields, with applications to English text. *IEEE Transactions on Information Theory*, **44**(3), 1319–1327.
- [Lee *et al.*(2006)] Lee, J. W., Park, J. B., Jo, H.-H., Yang, J.-S., and Moon, H.-T. (2006). Complexity and entropy density analysis of the Korean stock market. In *Proceedings of the 5th International Conference on Computational Intelligence in Economics and Finance (CIEF2006)*.
- [Maurer(2007)] Maurer, J. (2007). Boost random number library. Available at url <http://www.boost.org/libs/random/index.html>.
- [Neely *et al.*(1997)] Neely, C., Weller, P., and Dittmar, R. (1997). Is technical analysis in the foreign exchange market profitable? A genetic programming approach. *Journal of Financial and Quantitative Analysis*, **32**(4), 405–427.
- [Potvin *et al.*(2004)] Potvin, J.-Y., Soriano, P., and Vallée, M. (2004). Generating trading rules on the stock markets with genetic programming. *Comput. Oper. Res.*, **31**(7), 1033–1047.
- [Smith(1998)] Smith, S. N. (1998). Trading applications of genetic programming. *Financial Engineering News*, **2**(6).
- [Steuer *et al.*(2001)] Steuer, R., Molgedey, L., Ebeling, W., and Jiménez-Montaño, M. (2001). Entropy and optimal partition for data analysis. *The European Physical Journal B - Condensed Matter and Complex Systems*, **19**(2), 265–269.
- [Struzik and Siebes(1999)] Struzik, Z. and Siebes, A. (1999). The Haar wavelet transform in the time series similarity paradigm. In *PKDD '99: Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*, pages 12–22, London, UK. Springer-Verlag.
- [Wang(2000)] Wang, J. (2000). Trading and hedging in S&P 500 spot and futures markets using genetic programming. *Journal of Futures Markets*, **20**(10), 911–942.
- [Willems *et al.*(1995)] Willems, F. M. J., Shtarkov, Y. M., and Tjalkens, T. J. (1995). The context-tree weighting method: basic properties. *IEEE Transactions on Information Theory*, **41**(3), 653–664.