Entropy Rate and Profitability of Technical Analysis: Experiments on the NYSE US 100 Stocks *

Nicolas NAVET

LORIA-INRIA
BP 239, 54506, Vandoeuvre, France
E-mail: nnavet@loria.fr

Shu-Heng CHEN

AI-ECON Research Center, NCCU
Taipei, Taiwan 11623
E-mail: chchen@nccu.edu.tw

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 to predict 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 financial trading rules is correlated with the entropy rates of the price time series. Experiments are conducted on EOD data of the stocks composing the NYSE US 100 index during period 2000-2006, with the use of genetic programming to induce the trading rules.

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

1. Entropy Estimate

Entropy estimate is a field of investigation that has been very active over the last 10 years, one of the reason 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 entropy rate can be roughly classified in two main classes:[4]

*This is a slightly edited version of a paper published at the 6th International Conference on Computational Intelligence in Economics and Finance (CIEF2007), Salt-Lake City, USA, July 18-24, 2007. Contact author: Nicolas Navet.

- "Plug-in" (or maximum-likelihood) estimators that basically consist in 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 for instance Ref. 3,8) or the Context-Tree Weighting algorithm (see Ref. 6,12). Both approaches have been shown[4,5] 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.

1.1. $\hat{h}_{SM}$ entropy rate estimator

In this study, we use an estimator belonging to the Lempel-Ziv class that has been proposed in Ref. 8 (estimator a) from Theorem 1 in Ref. 8 - as in Ref. 7, 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:

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

where Λ_i^n 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 $s_{i-n}, ..., s_{i-1}$.

This estimator, which is well known and often used in the literature (see, for instance, Ref. 7), has been shown in Ref. 8 to have better statistical properties and performances that 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 taking its value in the set $\{1, 2, ..., 8\}$. 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[10] , which demonstrates the quality of the estimator since 3 is the best that can be obtained with a "perfect" random generator.

1.2. *Entropy of NYSE US 100 stocks*

Here we estimate the entropy of the daily price time series of the stocks composing the NYSE US 100 index (composition of the index can be found at url `http://www.nyse.com/marketinfo/indexes/nyid_components.shtml`). The data is 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 boundary 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.[11]

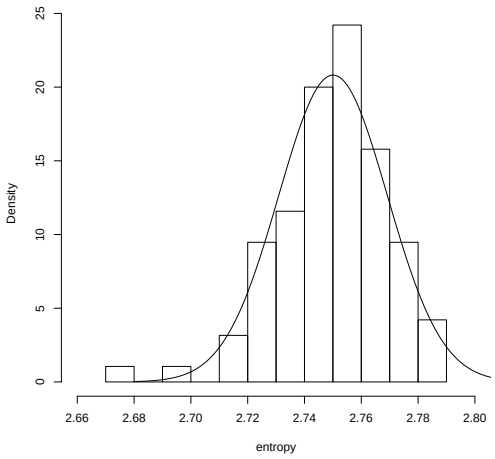

Fig. 1. Distribution of the entropy rate of the stocks composing the NYSE US 100 index (log. ratios of consecutive daily closing prices). A normal distribution of same mean and standard deviation is plotted for comparison. The reference period is $2000 - 2006$.

The distribution of entropy rate for the stocks of NYSE US 100 index

between from 01/01/2000 to 31/12/2006 is shown on figure 1. The minimum value is 2.68, the median 2.75, the mean 2.75 and the maximum value is 2.79. Time series have a high entropy since the theoretical upper bound is 3 and uniformly randomly generated sample achieve 2.90 with the same number of data points[a]. This is not very surprising per se since high entropy rates has been observed even at smaller time scales (see for instance Ref. 9). The 5 stocks from NYSE US 100 index of 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 of lowest entropy[b] 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 of the next section.

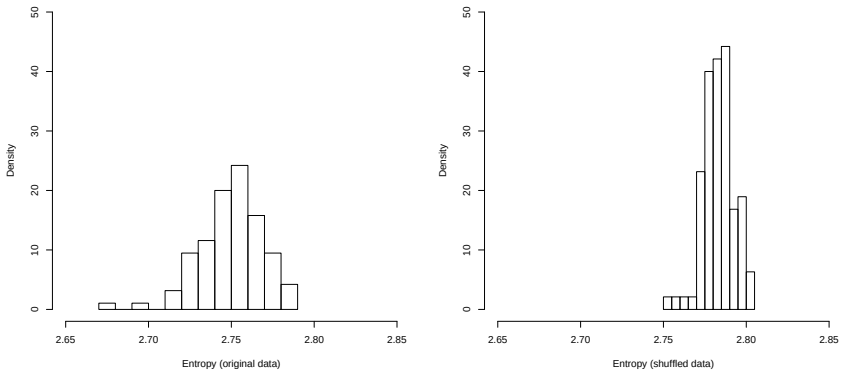

Fig. 2. Distribution of entropy rates of the original time series (left-hand graphics) and shuffled time series (right-hand graphics). x-axis ranges from 2.65 to 2.85 on both graphics.

Although the entropy is high, there is evidence that the original time series are not random. Indeed, we compared the entropy of the original time

[a]A value of 2.90 is obtained using 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 that the entropy value of some stocks.

[b]It is interesting to note that the autocorrelation of the log-returns is much larger for the lowest-entropy stocks than for the highest-entropy stocks. Up to a lag 100, there are on average 6 autocorrelations outside the 99% confidence bands for the lowest-entropy stocks versus 2 for the highest-entropy stocks. However, even for low-entropy stocks, the autocorrelation is too weak to be of any use for the purpose of forecasting.

series with the entropy of randomly shuffled variants of the original data (surrogate testing). Precisely, 100 shuffled time series of each original time series (after discretization step) are generated and their average entropy is measured. 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 on figure 2. This provides evidence that, at least for some stocks, there are (weak) temporal dependencies in the original time series. The question addressed in the next section is whether GP is able to take advantage of these temporal dependencies and produce profitable trading strategies.

2. Experiments with Genetic Programming

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 in making the investment decision randomly (*"Lottery Trading"*). We follow the methodology proposed in Ref. 2 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.

Experiments are conducted on 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, then a subset of top-performing rules are further selected on unseen data (validation set) and the best rule on the validation set is then evaluated on the out-of-sample test period. As classically done in the literature in terms of data-preprocessing, data are 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 loss stop (-5%), a profit target stop (10%) and a 90-days 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 100000\$ and the size of each position equal to 100% of the current equity. Functions, terminals and parameters of the GP runs are indicated in Appendix B.

From the results shown in tables 1 and 2, one should conclude that, with our experimental setup, selecting the stocks of 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 in Ref. 1, because

Table 1. Net return of GP and Lottery trading (LT) on the highest entropy stocks (rounded to the nearest 500$). First two columns are the average profit with GP (20 runs) and Lottery Trading (1000 runs). Third (resp. forth) column indicates whether we should reject the hypothesis that GP (resp. LT) does not outperform LT (resp. GP) at a 5% confidence level.

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

Table 2. Net return of GP and Lottery trading (LT) on the lowest entropy stocks (same settings as table 1). Experiments conducted with the possibility of selling short the stocks do not show significant improvements.

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

GP is often 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 stocks of lowest entropy as can be seen on figure A1. On the contrary, in the two cases (BAX and WAG) 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.

3. 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, by surrogate testing, that there are some weak temporal dependencies in the time series. Next step was to test the hypothesis that selecting the stocks of lowest entropy, the ones with the most predictable price time series, would lead to less risky investments. In the experiments, we did not observe that this hypothesis holds.

Recent studies (e.g. Ref. 4,7) have shown that Context Tree Weighting (CTW) entropy estimators often lead to faster convergence rates than Lempel-Ziv based estimators. Since, samples of daily data are of small size, the use of CTW may lead to some some improvements, although what is really crucial here is not the precise entropy estimate but the relative ordering between distinct time series.

Here, empirical evidences suggest to us that predictability is neither a necessary or sufficient condition for profitability. The predictability test only tells us the existence of temporal patterns, but 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 of 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 attribute. We think that further study about the relation between predictability and profitability should not rest upon only a single measure.

In this study we restrain ourselves to the stocks composing the NYSE US 100 because they are of primary interest for 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 not good candidates for a selection technique based on entropy. Future experiments should include stocks of lower entropy that do not belong to the NYSE U.S. 100, and other time scales should be considered. In particular, higher frequency data would enable us to study the variations of entropy over time.

References

1. S.-H. Chen, T.-W. Kuo, and K.-M. Hoi. 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, 2007. Forthcoming.
2. S.-H. Chen and N. Navet. Pretests for genetic-programming evolved trading programs: zero-intelligence strategies and lottery trading. In Irwin King, Jun Wang, Laiwan Chan, and DeLiang 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, October 3-6 2006. Springer.
3. M. Farach, M. Noordewier, S. Savari, L. Shepp, A. Wyner, and J. Ziv. 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, 1995. Society for Industrial and Applied Mathematics.
4. Y. Gao, I. Kontoyiannis, and E. Bienenstock. From the entropy to the statistical structure of spike trains. In *2006 IEEE International Symposium on Information Theory*, pages 645–649, July 2006.
5. Y. Gao, Y. Kontoyiannis, and E. Bienenstock. 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, December 2003.
6. M. B. Kennel and A. I. Mees. Context-tree modeling of observed symbolic dynamics. *Physical Review E*, 66(5):056209, Nov 2002.
7. M. B. Kennel, J. B. Shlens, H. D. I. Abarbanel, and E. J. Chichilnisky. Estimating entropy rates with bayesian confidence intervals. *Neural Computation*, 17(7):1531–1576, 2005.
8. I. Kontoyiannis, P. H. Algoet, Y. M. Suhov, and A. J. Wyner. Nonparametric entropy estimation for stationary processes and random fields, with applications to english text. *IEEE Transactions on Information Theory*, 44(3):1319–1327, 1998.
9. J. W. Lee, J. B. Park, H.-H. Jo, J.-S. Yang, and H.-T. Moon. 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)*, 2006.
10. J. Maurer. Boost random number library. Available at url `http://www.boost.org/libs/random/index.html`, 2007.
11. R. Steuer, L. Molgedey, W. Ebeling, and M.A. Jiménez-Montaño. Entropy and optimal partition for data analysis. *The European Physical Journal B - Condensed Matter and Complex Systems*, 19(2):265–269, February 2001.
12. F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens. The context-tree weighting method: basic properties. *IEEE Transactions on Information Theory*, 41(3):653–664, 1995.

Appendix A. Price Time Series Between Years 2000-2006

The price time series used in the experiments are shown in figures A1 and A2. The first third of the graphics is the training period, the second third the validation period and the last third is the test period.

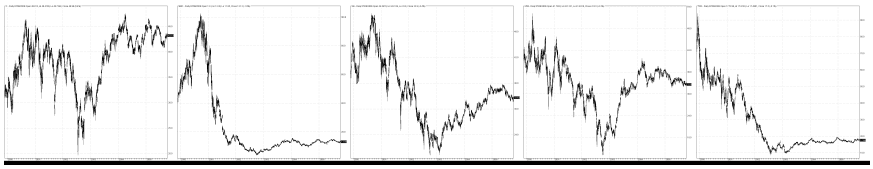

Fig. A1. Price time series of stocks having the lowest entropies : C, EMC, GE, JPM, TWX.

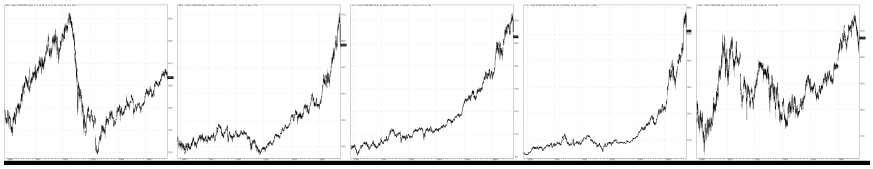

Fig. A2. Price time series of stocks having the highest entropies : BAX, MRO, OXY, VLO, WAG.

Appendix B. Genetic Programming Settings

The GP program implements strongly typed GP with the set of functions, terminals and parameters indicated below:

- Population size: 1000, number of generations: 50
- Minimum-maximum tree depth: 7
- Function set: +, -, *, /, norm, average, max, min, lag, and, or, not, >, <, if-then-else, true, false.
- Terminal set: price, 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
- Elitism: 10 best individuals are kept for the next generation
- Selection scheme: tournament selection of size 3
- Fitness function: net return
- Transaction costs: 0.1%
- Number of best trees saved for validation: 10