Theory of Stochastic Process
Vol.0 (00), no.0 ,1999, pp.00-00

THOMAS HELLSTRÖM

DATA SNOOPING IN THE STOCK MARKET

This paper is a scientific introduction to technical stock analysis. The problems with performance evaluation of near-random-walk processes are illustrated with some examples and the consequences for algorithm development and performance evaluation are discussed. The prediction task is formalized. Existing benchmarks and testing metrics are surveyed, and some new measures are introduced.

AMS 1991 subject classifications. 90A09.

Key words and phrases. Returns, Trading rules, Trading simulations, Stock predictions, Benchmarks, Performance measures.

1. INTRODUCTION

Predicting stock prices is generally accepted to be a very difficult task. The stock prices behave very much like a random-walk process, both when investigating them statistically and when looking at the results from properly evaluated attempts to create prediction algorithms. This paper shows various ways to define the prediction problem, introduce performance metrics and suggest suitable benchmarks for performance evaluation. At first we take a look at how the near random-walk behavior has direct implications on the prediction and evaluation tasks.

2. PREDICTING AN ALMOST-RANDOM-WALK TIME SERIES

The prediction task is impossible, if the time series $y(t)$ that we attempt to predict is an absolute random walk. In such a case *any* algorithm for prediction of the sign of $\Delta y(t)$ produces a 50% hit rate in the long run. The best we can hope for is that $y(t)$ has a “near to” random-walk behavior, but with limited predictability. Degrees of accuracy of 54% hit rate in the predictions are often reported as satisfying results for stock predictions. See e.g. Tsibouris and Zeidenberg (1995) or Baestaens, van den Bergh, Vaudrey (1996).

The purpose of evaluating a prediction algorithm is to produce an answer "Yes" or "No" as to whether the algorithm really has predictive powers. The evaluation task is directly affected by the relatively low degree of accuracy that can be expected, even from a "successful" model. Some simple statistical exercises show the situations that may arise when trying to evaluate a prediction algorithm.

Scenario 1:

Assume that we are doing one-day predictions of a stock time series consisting of equal numbers of daily moves up and down during one year of 250 trading days. This is a realistic assumption. The average $up/(up+down)$ ratios for a large number of Swedish stocks is in Hellström (1998b) shown to be between 50% and 51%.

A totally random prediction algorithm is applied for each day. The algorithm simply produces a "1" (predicted move up in stock price) and a "0"(otherwise) totally at random. The probability that x predictions are correct is given by:

$$P(\textbf{hit rate} = x) \;=\; \begin{pmatrix} 250 \\ x \end{pmatrix} 0.5^x * 0.5^{250-x}. \tag{1}$$

This means that $P(\textbf{hit rate} \leq x)$ follows the binomial distribution $\textbf{binom}(250, 0.5)$ and therefore

$$P(\textbf{hit rate} > x) \;=\; 1 - P(\textbf{hit rate} \leq x) = \; 1 - \textbf{binom cdf}(x, 250, 0.5). \tag{2}$$

Here **binom cdf** denotes the binomial cumulative distribution function. Insertion of $x = 0.54 * 250 = 135$ yields $P(\textbf{hit rate} > 135) \; = 0.092$ as the probability that the random prediction algorithm gives a hit rate higher than 54%. Thus we are running a 9% risk of classifying the random algorithm as a useful predictive model. This corresponds to what statisticians call a "Type II error", i.e. accepting a false hypothesis. Lowering the required hit rate limit to 52% hit rate would increase the risk for a Type II error to $1-\textbf{binom cdf}(0.52 * 250, 250, 0.5) = 24\%$. Not a very advisable thing to do, apparently.

Scenario 2:

Assume that we are evaluating technical indicators that produce sell and buy signals once a week on average. We have selected one hundred different indicators and want to conduct a proper test, and decide if any of these indicators has predictive powers, which we define to be a hit rate of $> 55\%$. For a test period of 10 years we get 500 predictions from each indicator. They are compared to the actual stock chart, and then hit rates are computed for the indicators. What is the probability that a random indicator would slip through this test?

As before, we can compute the probability that a purely random indicator produces x or fewer correct signals (hits) when applied to a stock: $P(\textbf{hit rate} \leq x)$ follows the binomial distribution $\textbf{binom}(500, 0.5)$ and thus

$$P(\textbf{hit rate} > x) \; = \; 1 - P(\textbf{hit rate} \leq x) = \; 1 - \textbf{binom cdf}(x, 500, 0.5). \qquad (3)$$

We compute $P(\textbf{hit rate} > 0.55 * 500) = 0.0112$ as the probability that a random indicator gives a hit rate higher than 55%. But since we started off with 100 different indicators, we must calculate the probability that we falsely accept ANY of the 100 random indicators. This risk is $1 - (1 - 0.0112)^{100}$, which calculates to 68%. It should be noticed, that there are many hundreds of suggested indicators and rules, claimed to really predict future stock prices. In the light of what we have just seen, the mere selection of one of these indicators, based on its past performance, can be statistically totally unacceptable.

Why Do We Get these Results

The primary reason for obtaining results like the ones shown above, is that the limits set for accepted prediction hit rates are too low. Increasing them to, say 60% would provide considerably safer results. However, then the problem would be to find prediction algorithms that really produce such high hit rates for the required test period. The reason that so many papers present hit rates in the region below 55% may be simply that the prediction task is impossible (at least with the reported method,) and the only way to obtain results that seem to be significant is to keep the required hit rate on a level where even a random predictor would produce them.

Scenario 2 above also pinpoints another important issue to be borne in mind, especially when selecting the best performing algorithm or technical indicator. The dramatic increase to 68% in Scenario 2, is due to the fact that the selection is done *in sample*. Given enough different indicators, it would be possible to find one with **any** hit rate. It corresponds to overfitting a powerful model to a set of data points. The conclusion is that a test set of data points must be kept untouched during the *entire* parameter estimation and model selection process. Although this may sound trivial, it is in fact quite difficult to fulfill completely. Model selection is in effect even after the test results have been published, since good prediction results probably get more attention than bad ones. For a thorough analysis of related problems refer to Lo (1996).

3. AVAILABLE DATA

A prediction algorithm uses a set of known entities to produce a prediction of future values for the same or other entities. In the case of stock predictions, the entities can be divided into two categories: pure *technical* data and *fundamental* data.

Technical Data

The daily available data for each stock is represented in the following 4 time series, with data for each day of trading (intra-day data is often also available but is seldom used)[1]:

Close	The price of the last performed trade during the day
High	Highest traded price during the day
Low	Lowest traded price during the day
Volume	The total number of traded stocks during the day

Fundamental Data

Apart from the daily sampled data described above, there is a lot of information concerning the activities and financial situation of each company. Most companies quoted at a stock market are analyzed on a regular basis by the professional market analysts at the financial institutes. The analyses are often presented as numerical items, which are supposed to hint at the "true" value of the company's stock. The buy and sell recommendations are then formed, either intuitively or with some sort of mathematical analysis of these items.

4. PERFORMANCE EVALUATION

A way to measure the performance of a prediction algorithm is needed in two phases of the development cycle of the prediction system. First, during the modeling phase, where a model is selected or where optimal settings on unknown parameters in the model or in the trading rules have to be decided upon. Secondly, when the complete algorithm is put to test on historical data, to see if it serves the original goal of the development project.

Evaluation of prediction performance is an important, difficult, and often overlooked stage in the development of financial prediction algorithms. In the case of an algorithm based on trading rules, one problem is the comparatively low number of produced trades, which constitutes a somewhat weak statistical basis for our performance measures. Apart from this, we have the problem with overtraining and

[1] *Open*, the price for the first performed trade for the day is sometimes also available

selection bias. By tuning the parameters in the algorithm to maximize the performance on historical data, we always run a risk of fitting the algorithm too closely to the data set. Even for a rather "small" model (i.e. one with relatively few degrees of freedom), the problem must not be overlooked. The near-random-walk behavior in the data, combined with the comparatively low required prediction accuracy (a little higher than chance is normally regarded as sufficient,) make the situation very delicate. Illustrative examples were shown in section 2.

A general problem with financial predictions is the non-stationary nature of the process, i.e. the performance of a trading strategy varies over time, due to changing global conditions affecting the market as a whole. One strategy may work fine in a trending market, while another works best in a non-trending market. Even if this should really be taken care of by the modeling phase, it often causes problems in the final evaluation made on historical data.

From all this it should be clear that performance computation and evaluation are vital ingredients in a scientific approach to stock prediction. The evaluation part defines a number of performance metrics and suitable benchmarks to which the computed metrics can be compared in order to judge the prediction algorithm. Two properties are considered particularly important for metrics to be used: relevance to the prediction task (i.e. measure what we try to model) and the availability of a benchmark. The latter is extremely important, since we are dealing with near-random-walk processes, in which for a prediction system to be successful, it barely has to outperform pure chance. Furthermore, whether the performance is good or bad depends on the alternatives. Even a prediction algorithm that loses money sometimes, may turn out to be successful when compared to some of the alternatives.

We now gradually become more and more specific as we categorize prediction algorithms, performance metrics and benchmarks. Prediction algorithms for stock prices can be categorized in a number of ways. One categorization focuses on the way the points to predict are selected. Two broad classes can be identified: *The Time Series Approach* and *The Trading Rule Approach*. The methods, suitable performance metrics and benchmarks are described in separate sections below.

5. THE TIME SERIES APPROACH

The traditional way to define a stock prediction problem is to form a time series $y(t)$ from the stock prices $Close$. The most common way is to use the h-day *returns* $R_h(t)$ defined as

$$R_h(t) = \frac{Close(t) - Close(t-h)}{Close(t-h)}. \tag{4}$$

In this context the time series R_h is denoted y. y is furthermore assumed to be a function g of the k previous values in the same time series. To predict the return h days in the future, we thus assert

$$y(t+h) = g(y(t), y(t-1), ..., y(t-k)). \tag{5}$$

The task for the learning or modeling process is to find the function g that best approximates a given set of measured data. The unknown function g can be defined in many ways, e.g. as a linear autoregressive (AR) model or a feed-forward neural network. The unknown parameters in the model are normally computed by a learning (identification) algorithm that minimizes the root of the mean squared prediction error

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N} (g(t) - y(t+h))^2}. \tag{6}$$

It is most common to let the minimized RMSE measure (6) be the end point in the prediction task. However, in order to utilize the predictions, a decision-taking rule has to be created. A simple rule when predicting returns is:

$$D(t) = \begin{cases} \text{Buy} & : \text{ if } g(t) > \alpha \\ \text{Sell} & : \text{ if } g(t) < -\beta \\ \text{Do nothing} & : \text{ if } g(t) = 0 \end{cases} \tag{7}$$

where α and β are threshold parameters for buy and sell actions, depending on the predicted change in the stock price (by defining a decision rule such as (7), the Times Series oriented algorithm is transformed into a Trading Rule which is further described in section 6).

5.1 Performance metrics

Predictions according to the Time Series Approach are normally evaluated at a fixed horizon but, as was previously described (7), can be transformed into a trading rule. The metrics for testing algorithms at a fixed horizon work by comparing the predicted values to the actual outcome h days ahead. The predictions of stock prices for time t are expressed below by the time series $\{\hat{Close}(t), t = h+1, ..., N\}$. The actual prices are denoted by the time series $\{Close(t), t = 1, ..., N\}$. The predictions of the h-day return at time t are denoted by the time series $\{\hat{R}(t), t = h+1, ..., N\}$. The actual returns are denoted by the time series $\{R(t), t = h+1, ..., N\}$ and were defined in (4). We assume an h step horizon in the predictions. I.e.: The predictions $\hat{Close}(t)$ and $\hat{R}(t)$ are produced at time $t-h$.

Refenes (1995) provides a survey of a large number of measures of performance for financial predictions. Below we present the ones we have found essential, as well as some new metrics believed to be necessary.

RMSE

The RMSE for the predicted stock prices $Close$ is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=h+1}^{N} (Close(t) - \hat{C}lose(t))^2}. \tag{8}$$

The RMSE should be obviously as low as possible for a good prediction algorithm.

Hit Rate H_R

The hit rate of a return predictor indicates how often the sign of the return is correctly predicted. It is computed as the ratio between the number of correct non-zero predictions $\hat{R}(t)$ and the total number of non-zero moves in the stock time series. I.e.:

$$H_R = \frac{\left|\left\{R(t)\hat{R}(t) > 0\right\}_{h+1}^{N}\right|}{\left|\left\{R(t)\hat{R}(t) \neq 0\right\}_{h+1}^{N}\right|}. \tag{9}$$

The norm of the set in the definition is simply the number of elements in the set.

The reason for avoiding both zero predictions and zero returns in the computation of the hit rate is the following: if zeros were included, we would have to decide whether the following five combinations should be regarded as "hits" or not:

$\hat{R}(t) = 0$	$R(t) > 0$
$\hat{R}(t) = 0$	$R(t) < 0$
$\hat{R}(t) = 0$	$R(t) = 0$
$\hat{R}(t) > 0$	$R(t) = 0$
$\hat{R}(t) < 0$	$R(t) = 0$

Regardless of the choice made for the classification of these situations, the result is invariably an asymmetric treatment of the positive and negative returns. Since the zero-valued one-day returns can account for more than 20% of the samples in typical stock data, they would result in "Up fractions" arbitrarily either greater or less than 50%. Such a result would conceal the random-walk nature of the time series. By removing all zeros from both predictions and outcome, "Up fractions" very close to 50% are achieved. Therefore, in the case of one-day returns (i.e. $h = 1$), a hit rate H_R, significantly greater than 0.5, can be regarded as true predictions of the sign of the returns. Refer to Hellström (1998b) for more details.

There are two useful extensions to the general hit rate H_R defined above. It is often interesting to distinguish the ability to predict positive and negative returns R respectively. The measures H_{R+} and H_{R-} capture this. We define

$$H_{R+} = \frac{\left| \left\{ R(t) > 0 \ AND \ \hat{R}(t) > 0 \right\}_{h+1}^{N} \right|}{\left| \left\{ \hat{R}(t) > 0 \right\}_{h+1}^{N} \right|} \tag{10}$$

and

$$H_{R-} = \frac{\left| \left\{ R(t) < 0 \ AND \ \hat{R}(t) < 0 \right\}_{h+1}^{N} \right|}{\left| \left\{ \hat{R}(t) < 0 \right\}_{h+1}^{N} \right|}. \tag{11}$$

By demanding both $H_{R+} > 0.5$ and $H_{R-} > 0.5$ the misleading effects of long time trends in data are illuminated.

Net Profit

The ultimate measure of success for a prediction algorithm is its ability to produce profit if applied to real trading. This profit can be estimated for a Time Series Approach method, by assuming a trade at every time step in the direction of the predicted change. The net profit P_n for a time series prediction is computed as:

$$P_n = \sum_{t=h+1}^{N} (Close(t) - Close(t-h)) \cdot sign(\hat{Close}(t) - Close(t-h)). \tag{12}$$

5.2. Benchmarks

A benchmark should provide an alternative and a standardized way to produce predictions. The algorithm at test is then compared to the alternative in order to evaluate the performance. In this section benchmarks for predictions according to the Time Series Approach are presented. The $\varepsilon - increase\ prediction$ and *Naive Prediction of Returns* are introduced as extensions to the well-known *Naive Prediction of Stock Prices.* The measures T_r, HR_ϵ, HR_N, and P_r are suggested as testing metrics for performance evaluation relative to these benchmarks.

Naive Prediction of Stock Prices

The naive price predictor asserts that the best estimate $\hat{C}lose(t+h)$ of the future price $Close(t+h)$ is today's price $Close(t)$. This is a direct consequence of the random-walk hypothesis. It is always a good idea to measure the quality of a predictor in relation to this naive predictor. This is done in the *Theil coefficient of inequality* T_c, defined as the quotient between the $RMSE$ for the investigated predictor and the $RMSE$ for the naive price predictor . I.e.:

$$T_c = \frac{RMSE_y}{RMSE_{yN}} = \frac{\sqrt{\sum_{t=h+1}^{N}(Close(t) - \hat{C}lose(t))^2}}{\sqrt{\sum_{t=h+1}^{N}(Close(t) - Close(t-h))^2}}. \tag{13}$$

The Theil coefficient is often referred to as *the information coefficient* or the *t-test.* For $T_c > 1$ the predictor is worse than the naive price predictor, while $T_c < 1$ implies that the predictor is making better predictions.

$\varepsilon - increase$ **Prediction**

The $\epsilon - increase$ predictor asserts that the best estimate of the future price $Close(t+h)$ is today's price $Close(t) + \varepsilon$. The purpose of adding a small positive number ε to today's price is simply to enable computation of the hit rate for this predictor. The hit rate of this predictor provides a measure for the overall positive trend that "normally" makes an increase in price more likely than a decrease. The hit rate for the $\epsilon - increase$ predictor is defined as:

$$H_\epsilon = \frac{\left|\{R_h(t) > 0\}_{h+1}^{N}\right|}{\left|\{R_h(t) \neq 0\}_{h+1}^{N}\right|}. \tag{14}$$

The measure HR_ϵ is defined as the ratio between the predictor's hit rate H_R (as defined in (9)) and H_ϵ. I.e.:

$$HR_\epsilon = \frac{H_R}{H_\epsilon}. \tag{15}$$

HR_ϵ compares the hit rate of the predictor to that of the $\epsilon - increase$ predictor. For $HR_\epsilon < 1$ the predictor is worse than the $\epsilon - increase$ predictor, while $HR_\epsilon > 1$ implies that the predictor is making better predictions.

Naive Prediction of Returns

The Theil coefficient (13) compares the performance to that of the naive price predictor. We propose a similar measure to compare to the performance of the naive return predictor. The naive prediction of stock returns asserts today's return $R_h(t)$ (price increase since $t-h$) as the best estimate of $R_h(t+h)$. This naive prediction is formed from the observation of a one-step memory in the price generating process. In Hellström (1998b) the autocorrelation of stock returns was shown to exhibit a significant first lag component that indicates a correlation between adjacent returns. It is a good idea to measure the $RMSE$ of a return predictor in relation to the $RMSE$ for this naive return predictor. This is done in *Theil coefficient for returns* T_r defined as:

$$T_r = \frac{RMSE_r}{RMSE_{rN}} = \frac{\sqrt{\sum_{t=h+1}^{N}(R_h(t) - \hat{R}_h(t))^2}}{\sqrt{\sum_{t=h+1}^{N}(R_h(t) - R_h(t-h))^2}}. \tag{16}$$

For $T_r > 1$ the predictor is worse than the naive return predictor, while $T_r < 1$ implies that the predictor is making better predictions.

The Naive Prediction of Returns can also be compared on the basis of hit rate. This is the purpose of our suggested measure HR_N ("Hit rate relative to the Naive Predictor") and is described below. The hit rate H_N for the naive return predictor is first computed as:

$$H_N = \frac{\left|\{R_h(t)R_h(t-h) > 0\}_{h+1}^{N}\right|}{\left|\{R_h(t)R_h(t-h) \neq 0\}_{h+1}^{N}\right|}. \tag{17}$$

The Naive Prediction of Returns assumes, that an upswing is followed by yet another upswing the next day, and a downswing by yet another downswing.

The Relative Hit Rate HR_N is defined as the ratio between the hit rate of the predictor H_R (as defined in (9),) and that of the naive return predictor. I.e.:

$$HR_N = \frac{H_R}{H_N}. \tag{18}$$

HR_N compares the hit rate of the predictor relative to that of the naive return predictor. For $HR_N < 1$ the predictor is worse than the naive return predictor, while $HR_N > 1$ implies that the predictor is making better predictions.

Buy-and-Hold

The *Buy-and-Hold* profit P_b for a time period $\{1...N\}$ and one particular stock is defined as:

$$P_b = Close(N) - Close(1), \tag{19}$$

i.e. the profit made when buying at the start and selling at the end of the time period. The *Profit Relative to Buy-and-Hold* P_r is defined as the ratio between the predictor's *Net Profit* P_n, as defined in (12), and the *Buy-and-Hold* profit P_b. I.e.:

$$P_r = \frac{P_n}{P_b}. \tag{20}$$

For $P_r < 1$ the predictor's net profit is worse than the Buy-and-Hold alternative, while $P_r > 1$ implies that the predictor is making higher profits. This measure tests whether the net profit is due to real predictions or merely due to a general market trend.

5.3 Conclusions on evaluation

The evaluation of a prediction based on the Time Series Approach should be presented with annual figures for the following entities:

- If absolute prices *Close* are predicted: the Theil coefficient of inequality T_y (or $RMSE_y$ and $RMSE_{yN}$ separately)
- If returns R are predicted: the Theil coefficient for returns T_r (or $RMSE_r$ and $RMSE_{rN}$ separately)
- The hit rate relative to the ϵ-increase predictor HR_ϵ (or H_R and H_ϵ separately)
- The hit rate relative to the Naive Prediction of Returns HR_N (or H_R and H_N separately)
- The number of predictions
- The *Profit Relative to Buy-and-Hold* P_r (or P_n and P_b separately)

In addition to the annual values, mean values for the entire time period are also useful for fast comparison of methods.

The time series formulation based on the minimized RMSE measure is not always ideal for useful predictions of financial time series. Some reasons are:

1. The fixed prediction horizon h does not reflect the way in which financial predictions are being used. The ability of a model to predict should not be evaluated at one single fixed point in the future. A big increase in a stock value 14 days into the future is as good as the same increase 15 days into the future.

2. The RMSE treats all predictions, small and large, as equal. This is not always appropriate. Prediction points that would never be used for actual trading (i.e. price changes too small to be interesting) may cause higher residuals at the other points of more interest, to minimize the global RMSE.

3. A small predicted change in price, followed by a large real change in the same direction, is penalized by the RMSE measure. A trader is normally happy in this case, at least if, say, the small positive prediction was large enough to give a buy signal.

4. Several papers, e.g. Leitch and Tanner (1991) and Bengio (1997), report a poor correlation between the RMSE measure and the profit made by applying a prediction algorithm. A strategy that separates the modeling from the decision-taking rule is less optimal than modeling the decision taking directly (Moody (1992)). Both arguments 2 and 3 provide some explanations to these results.

6. THE TRADING RULE APPROACH

The other major type of prediction algorithms defines a trading rule as a time series $T(t)$ as

$$T(t) = \begin{cases} \text{Buy} & : \text{ if } g(t) = 1 \\ \text{Sell} & : \text{ if } g(t) = -1 \\ \text{Do nothing} & : \text{ if } g(t) = 0 \end{cases} \tag{21}$$

where g is a function f of the stock prices $Close$ or stock returns R (4) up to time t. E.g.:

$$g(t) = f(Close(t), Close(t-1), ..., Close(t-k)) \tag{22}$$

The function f determines the type of trading rule. Standard technical indicators such as the Stochastic Oscillator, the Relative Strength Index (RSI) or the Moving Average Convergence/Divergence (MACD) can all be described in this fashion. The task for the learning process in The Trading Rule Approach is to find the function f that maximizes the profit, when applying the rule on real data. Note the difference between this and The Time Series Approach, where the learning task is to find a function g that minimizes the $RMSE$ error (6) for the entire time series.

Example:

The function g is defined as:

$$g(t) = \begin{cases} 1: & \text{if } mav_S(t) > mav_L(t) \quad \text{AND} \quad mav_S(t-1) \leq mav_L(t-1) \\ -1: & \text{if } mav_S(t) < mav_L(t) \quad \text{AND} \quad mav_S(t-1) \geq mav_L(t-1) \\ 0: & \text{otherwise} \end{cases} \tag{23}$$

where $mav_k(t)$ is a moving average of length k. I.e.:

$$mav_k(t) = \frac{1}{k} \sum_{m=0}^{k-1} Close(t-m) \tag{24}$$

The trading rule is illustrated in Figure 1. The learning in this example consists of finding optimal values to specify the function g, i.e.: the length variables in the moving averages mav_L and mav_S. The trading rule (21) signals "Buy," if the short moving average mav_S crosses the long moving average mav_L from below. A "Sell" signal is issued when mav_S crosses the mav_L from above. The optimal settings for S and L are determined by the learning process.

The Trading Rule Approach avoids many of the problems previously described of the Time Series Approach but does indeed have problems of its own, primarily that of statistical significance. The trading rule $T(t)$ normally issues Buy or Sell signals only for a minor part of the points in the time series. While being one of the big advantages, it also presents serious statistical problems when computing levels of significance for the produced performance. It is easy to find a trading rule that historically outperforms any stock index, as long as it does not have to produce more than a few signals.

6.1 Performance Metrics

Trading-rule-based methods are normally evaluated by a trading simulation where the trading rule controls the buying and selling of one or several stocks for a period of time. It is however also possible to evaluate trading rules by treating them as a fixed horizon prediction.

Hit rate at a fixed horizon

By viewing the Buy and Sell rules separately, a trading rule can be evaluated in a fashion similar to the Time Series Approach. The hit rate H_B of a Buy rule indicates how often a buy signal is followed by a true increase in the stock price. We define H_B as

$$H_B = \frac{\left|\{g(t) = 1 \text{ AND } R_h(t+h) > 0\}_1^{N-h}\right|}{\left|\{g(t) = 1\}_1^{N-h}\right|} \tag{25}$$

where g is the function specifying the trade rule as described in (21).

The hit rate H_S of a Sell rule indicates how often a sell signal is followed by a true decrease in the stock price. We define H_S as

$$H_S = \frac{\left|\{g(t) = -1 \text{ AND } R_h(t+h) < 0\}_1^{N-h}\right|}{\left|\{g(t) = -1\}_1^{N-h}\right|}. \tag{26}$$

The prediction horizon h is in this case set arbitrarily and the performance is evaluated for the Buy and Sell part separately. The problem with this metric is the lack of an objective benchmark. By choosing a long enough prediction horizon h, most buy signals "result" in an increase in stock price and therefore in a very high hit rate H_B. The large h however also causes a correspondingly low hit rate H_S for the sell signals. By demanding both a high H_B and a high H_S, a good estimate of the overall performance is often possible. H_S and H_S are also useful for comparison between different prediction algorithms.

Profit at a fixed horizon

We can also compute the mean profit for the buy and sell signals instead of just the hit rate. The mean profit achieved if the Buy rule is obeyed is computed as

$$P_B = \frac{1}{|\{g(t) = 1\}|} \sum_{g(t)=1} R_h(t+h). \tag{27}$$

The mean profit for the sell rule P_S is defined correspondingly. A well-performing algorithm should give a large positive P_B and a large negative P_S. The lack of an objective benchmark is however obvious even for these profit measures. As in the case with hitrates, comparing P_B and P_S can often give a useful estimate of the overall performance for the algorithm.

Trading Simulation

A trading simulation implements a trading-rule-based prediction algorithm (as defined in (21),) in a system, where real trading is simulated as closely as possible. This means that the trading rule's Buy part initiates buy actions, and the Sell part initiates sell actions. In this way a more realistic situation is achieved than the previously described one of having a fixed horizon. Transaction costs for the trades can be also easily incorporated. Trading simulation can be either done stock by stock or multi-stock, in which case portfolio management also becomes an important issue. This approach is implemented in the ASTA system which is described in Hellström (1998a). The following metrics are relevant when evaluating trading rules both in stock-by-stock and multi-stock simulations.

Total Profit

An intuitively appealing and very common measure is the total wealth achieved by the trader, when simulating trading over the available training data period. In the case of multi-stock predictions the wealth is often presented as a function of time in a so-called *equity diagram.*

Profit per Year

Compute the annual profit achieved by applying the trading algorithm. The mean annual profit could be used as total measure for the entire time period. However, it is also important to pay attention to the performance in each individual year.

Fraction Profitable Trades

Even if a benchmark for this entity is hard to find (buying a stock in the beginning of the simulation and selling it 10 years later may generate for example an impressive fraction of 100% profitable trades,) this figure gives a very good feeling for how the system would work in reality. It can also provide information about where the trading result actually occurred. If we for example get a huge overall profit but a very low fraction of profitable trades, we should suspect a few lucky trades to be responsible for the good simulation result. Annual computation of the fraction profitable trades is therefore a very important part of the evaluation.

Number of Trades

The estimated profit from a Trading-Rule-based system has a very weak statistical significance, if the number of trades produced during the simulation is low. The number of trades is therefore of central importance, and should be presented along with the trading results.

6.2 Benchmarks

In the previous section the lack of proper benchmarks for the fixed horizon metrics in combination with Trading-Rule-based predictions were discussed. What remains is therefore a benchmark for the Trading Simulation metrics. We need different benchmarks for a multi-stock trading system and for a system that predicts and evaluates each stock separately.

Index

The natural benchmark for a multi-stock trading simulation is some kind of stock index, which is also the method most often used by professional brokers. It is a reasonable benchmark because it compares the performance to a very available alternative: that of buying a mutual fund instead of doing the trading ourselves.

It can be argued, that the change in the index underestimates the average profit achieved for the stocks that constitute that index. The reason for this would be that the dividends for the stocks are not taken into account in the calculation of the index[2]. On the other hand, the dividends are not included in the profit calculations for the tested prediction algorithms either, which means, that it is subject to the same underestimation. Therefore, the change in index is considered a relevant benchmark when comparing different prediction methods by simulated trading.

The result of a trading simulation should be presented as annual profits together with the increase in index. The mean difference between these two figures constitutes the net performance of the system. The results may be presented in table form or as a histogram as shown in the lower part of Figure 2. The profit may also be displayed in so-called equity diagrams, as shown in the upper part of Figure 2. The stock index is presented in the same diagram for comparison. The curves are scaled, so the leftmost point has a wealth of 1 for both the trader and the index. The values for other points along the date axis can be then interpreted as wealth relative to the one at the starting point. A value 2.10 means, for example, that the start capital has grown to 210% of its original value. Therefore, the final value at the very right side of the diagram is the net result after the trading for the entire time period has been completed. The major drawback of this method is that the trades in the beginning of the time period affect the end result more than the ones in the end of the time period. This is a consequence of the cumulative nature of the simulation. The profits in the beginning of the time period are being reinvested, and therefore appear "several times" in the total wealth resulting from trading during the entire time period. A histogram with annual profits for the algorithm and the index should therefore be used instead. The number of trades for each year is also highly important for the statistical significance of the results.

Buy-and-Hold When doing single stock predictions (or multi-stock predictions stock by stock) the stock itself should be used as benchmark instead of a global index. In this case we consider the price development for the stock much like we do for the Buy-and-Hold benchmark described for the Time Series Approach. The comparison and presentation of the performance can be done in the same way as in the multi-stock situation described in the previous section.

BIBLIOGRAPHY

1. Baestaens, D. J. E. ,van den Bergh, W. M. and Vaudrey, H., *Market Inefficiencies, Technical Trading and Neural Networks*, In Dunis, C., editor, *Forecasting Financial*

[2]For the Swedish stock market 1988-1997, this underestimation was around 3% per year.

Markets, 1996, John Wiley & Sons, Chichester, England, (1996), 245-260.

2. Bengio, Y., *Training A Neural Network with a Financial Criterion Rather than a Prediction Criterion*, In *Decision Technologies for Financial Engineering, Proceedings of the Fourth International Conference on Neural Networks in the Capital Markets, NNCM-96*,1997, World Scientific, Singapore, (1997), 36-48.

3. Gleith, G., Tanner, J., *Economic forecast evaluation: Profit versus the conventional error measures.,* In *The American Economic Review, 1991*, (1991), 580-590.

4. Hellström, T., *ASTA - a Test Bench and Development Tool for Trading Algorithms*, Technical Report UMINF 98.12 ISSN-0348-0542, Department of Computing Science Umeå University, Sweden, (1998a).

5. Hellström, T., *A Random Walk through the Stock Market*, Licentiate Thesis, UMINF 98.16 ISSN-0348-0542, Department of Computing Science Umeå University, Sweden, (1998b).

6. Lo, A. W., *Data Snooping and Other Selection Biases In Financial Econometrics, Tutorial NNCM-96, Neural Networks in the Capital Market, Pasadena*, 1996, (1996).

7. Moody, J. E., *Shooting craps in search of an optimal strategy for training connectionist pattern classifiers*, In Moody, J. E., Hanson, S. J. and Lippman,P., editor, *Advances in Neural Information Processing Systems 4, Proceedings of the 1991 NIPS Conference, 1992,* Morgan Kaufmann Publishers, San Mateo , CA, (1992).

8. Refenes, A. P., *Testing Strategies and Metrics*, In Refenes, A. P., editor, *Neural Networks in the Capital Markets*, 1995, John Wiley & Sons, Chichester, England, (1995), 67-76.

9. Tsibouris, G. and Zeidenberg, M., *Testing the Efficient Markets Hypothesis with Gradient Descent*, In Refenes, A. P., editors, *Neural Networks in the Capital Markets*, 1995, John Wiley & Sons, Chichester, England, (1995), 127-136.

Department of Computing Science, Umeå University, S-90187 Umeå, Sweden
E-mail: thomash@cs.umu.se
http://www.cs.umu.se/~thomash

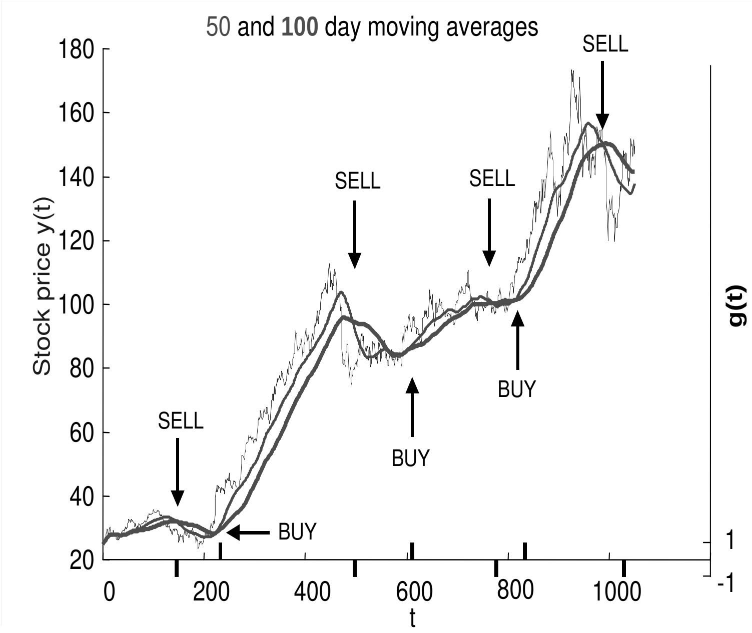

Figure 1: Example of a trading rule based on moving averages.

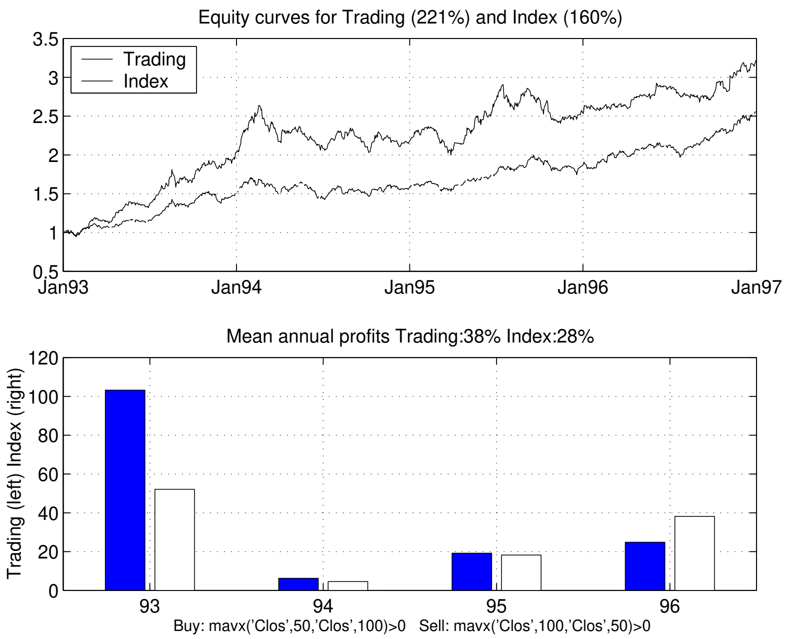

Figure 2: Presentation of performance for a trading-rule based system.