Optimization of Trading Rules with a Penalty Term

Thomas Hellström
Department of Computing Science
Umeå University
901 87 Umeå Sweden
thomash@cs.umu.se
http://www.cs.umu.se/~thomash

Abstract: A big problem when working with models for financial prediction is the estimation of out-of-sample performance for the obtained models or trading rules. In particular, it is very easy to jump into conclusions regarding trading rules that exhibit extremely profitable behavior, when tested on historical data. These misjudgments are often caused by the rules covering too few examples in the examined data. This paper deals with the problem in conjunction with nonconvex global optimization of trading rules by adding a constraint in the problem formulation. The effect is a regularization, where solutions covering too few examples are rejected. The modeling is performed with a sliding-window technique and generates different parameters for the optimized trading rules in each time window. Test results from the Swedish stock market show superior generalization ability in terms of risk-adjusted hit rates for the rules generated with the proposed method. Furthermore, the results show that the high hit rates achieved, to a large extent are a result of the adaptive modeling with sliding windows.

Keywords: financial series, forecasting, trading rules, optimization stock prediction.

1 Introduction

In recent years a variety of models for forecasting changes in stock market prices have been introduced. In this paper we consider the special kind of models, which produce predictions only for a very limited fraction of all observations, and a 'do not know' for the rest. In the trading community these models are often referred to as *technical indicators,* but can also include more complex models as well as models with many kinds of input, such as interest rates, stock indexes, and exchange rates. The general notion is that it might be easier to find a model allowed to output a 'do not know' result, than try to model non-existing dependencies in large parts of the input space.

The trading rules often are optimized with respect to parameters that control when the rules generate trading signals. This optimization is performed using historical data. Examples of this can be seen in Hellström, Holmström [9], where a technical indicator is tuned to give maximum profit in a simulated trading environment. A related problem is found in Iglehart, Voessner [11], where genetic algorithms are utilized to optimize a trading system designed to trade the Standard & Poors (S&P) 500 Index.

One of the most serious problems when optimizing trading rules is the lack of generalization for the obtained rules. It is usually a simple task to generate a trading rule that performs excellently on the training data, but has a very modest performance when tested on previously unseen data. One certain way to achieve this undesirable situation is to allow low support for the generated rules, i.e. accept rules that produce a very small number of trading signals for the training data. This increases the risk for undesirable data snooping, and normally gives very poor generalization performance. Moreover, the statistical uncertainties in the computed performance measures increase drastically as the size of the data set decreases. In this paper a solution for the problem is implemented as part of a numerical optimization routine that tunes parameters in trading rules for stock trading. Section 2 introduces the concept of Trading Rules, and defines the specific rules used in the study. The relevant performance measures are also discussed and defined. The learning algorithm is implemented as a nonconvex global optimization problem, and is described in Section 3. The results of applying the trading rules and learning strategies on data from the Swedish stock market are presented in Section 4. Section 5 closes with conclusions and suggestions for future research.

2 Trading Rules

A general way to formulate strategies for stock trading is to define a trading rule as a time series $T(t)$ such as

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

where g is a function of the previous stock prices *Close*:

$$g: \{Close(t), ..., Close(t-k)\} \curvearrowright \{-1, 0, 1\}. \quad (2)$$

Trading rule (1) is designed to serve as decision support in actual stock trading, as indicated by the labels *Buy, Sell,* and *Do Nothing.* Function g determines the type of the trading rule. By extending expression (2) with the input variables *High* (highest-paid price), *Low* (lowest-paid price), *Open* (first price) and *Volume* (number of traded stocks), most standard technical indicators, such as the *Stochastic Oscillator*, the *Relative Strength Index (RSI), Moving Average Convergence/Divergence (MACD)* etc. [14], can be described in this fashion. It is also clear that more sophisticated prediction algorithms based on traditional time-series analysis, neural networks, etc., can be formulated in the same general structure. In addition to the price and volume variables, such fundamental variables as interest rates and exchange rates may also be included as input to function g in (2). Quite often the buy and sell decisions are controlled by separate expressions and the trading rules are then denoted *Buy rule* and *Sell rule* respectively. Hereinafter we use the notation g_s to denote a trading rule applied to one specific stock s.

Example 1:

Function g is defined as

$$g(t) = \begin{cases} 1: & \text{if } mav_S(t) > mav_L(t) \wedge \\ & mav_S(t-1) \leq mav_L(t-1) \\ -1: & \text{if } mav_S(t) < mav_L(t) \wedge \\ & mav_S(t-1) \geq mav_L(t-1) \\ 0: & \text{otherwise} \end{cases} \quad (3)$$

where $mav_k(t)$ is a moving average of length k defined as

$$mav_k(t) = \frac{1}{k} \sum_{m=0}^{k-1} Close(t-m). \quad (4)$$

The trading rule (3) 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 can be determined by a learning process. For a general trading rule, a learning task can be defined as finding the optimal function g. In the case of standard technical indicators function g is normally parameterized with a few parameters that have to be determined. In the example above, the optimal settings for the moving-average lengths S and L are unknown and should be determined by a learning algorithm. This learning task is described in more detail in Section 3.

In this paper, the moving-average rule described above, as well as two other trading rules for generating *Buy* signals, are used to demonstrate the techniques with constrained optimization. All three are based on standard technical indicators, well-known by the trading community. For a thorough introduction to the subject, refer to Kaufman [14]. However, the standard indicators have been augmented with a term that includes the traded volume. This too is in accordance with common practice among traders. Trading signals are often given higher significance if they are accompanied by a high-traded volume for the stock in question. The dependence between traded volume and future stock prices has also been analyzed in academic research. The reliability of past stock returns is modeled taking the volume into account in Blume, Easley and O'Hara [2]. In Campbell, Grossman and Wang [4], the interaction between different types of market actors is analyzed with respect to the traded volume and the reversed returns. A survey of research dealing with the relation between traded volume and stock returns can be found in Karpov [13]. We include the traded volume as a term in all our technical trading rules. To facilitate a uniform modeling for *all* stocks in the market, a normalized measure has to be defined.

2.1 Gaussian Volume

The Gaussian volume $V_n(t)$ is a transformation of the traded volume (number of stocks) $V(t)$ defined as

$$V_n(t) = (V(t) - m_V(t))/\sigma_V(t), \quad (5)$$

where $m_V(t)$ and $\sigma_V(t)$ are computed in a running window of length n as

$$m_V(t) = \frac{1}{n} \sum_{i=1}^{n} V(t-i) \quad (6)$$

and

$$\sigma_V = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (V(t-i) - m_V(t))^2}. \quad (7)$$

V_n expresses the number of standard deviations, by which the volume differs from its running mean. The

normalization makes it possible to compare values of V_n for different stocks and also for different times. In this paper the Gaussian volume V_{10} is used and is denoted by $gvol10$, since this is the name of the ASTA implementation of the function (see Hellström [8]).

2.2 Crossing Moving-Averages

This trading rule is based on the principle shown in the example in Section 2. Moving averages have been used, for a long time, both by practitioners and academics investigating the predictability of financial markets. Levich and Thomas [18] find that moving-average rules produce statistically significant average returns for future contracts on currencies. Brock, Lakonishok and LeBaron [3] find the same behavior for the US stock market. A theoretical analysis of when and why moving-average trading rules work, is found in Kuo [16]. In this paper we define the *Buy* rule *mav* as

$$mav(x_1, x_2, x_3) = Mavx(x_1, x_2) \wedge gvol10 > x_3, \quad (8)$$

where

$$\begin{aligned} Mavx(x_1, x_2) = \quad & mav_{x_1}(t) > mav_{x_2}(t) \wedge \\ & mav_{x_1}(t-1) \leq mav_{x_2}(t-1) \end{aligned} \quad (9)$$

and mav_{x_1} and mav_{x_2} are given by (4).

2.3 Trading Channel Breakout

The main part of this trading rule is what is popularly known as *Bollinger Bands* (see e.g. page 91 in Kaufman [14]). The complete trading rule is defined as

$$break(x_1, x_2, x_3) = breakout(x_1, x_2) \wedge gvol10 > x_3, \quad (10)$$

where the *breakout* function is defined as

$$\begin{aligned} & breakout(x_1, x_2) = \\ & Close(t) > (mav_{x_1}(t) + x_2 \cdot \sigma_{x_1}(t)) \quad \wedge \\ & Close(t-1) \leq (mav_{x_1}(t) + x_2 \cdot \sigma_{x_1}(t)) \end{aligned} \quad (11)$$

and $mav_{x_1}(t)$ is given by (4). Function $\sigma_{x_1}(t)$ computes the standard deviation of the *Close* as

$$\sigma_{x_1}(t) = \sqrt{\frac{1}{x_1 - 1} \sum_{i=0}^{x_1 - 1} (Close(t-i) - mav_{x_1}(t))^2}. \quad (12)$$

The idea is to define an upper boundary for a trading channel and generate a *Buy* signal when the *Close* penetrates this boundary from below. This upper boundary is defined as the sum of a moving average mav_{x_1} and x_2 times an estimate of the standard deviation σ_{x_1}. The *breakout* function is illustrated in Figure 1 with $x_1 = 120$ and $x_2 = 3.5$.

2.4 Level of Resistance

The trading rule *Level of Resistance*, in this paper denoted *resist*, is based on a technique commonly executed by manual inspection of the stock charts. The general idea is to identify *peaks* in a window backwards, where the *Close* price is roughly the same. When such peaks are found, a *Buy* signal is generated if the *Close* price crosses from below the level for the found peaks. We define the trading rule *resist* as

$$\begin{aligned} & resist(x_1, x_2, x_3, x_4) = \\ & xresist(x_1, x_2, x_3) \wedge gvol10 > x_4 \end{aligned} \quad (13)$$

where

$$\begin{aligned} xresist(x_1, x_2, x_3) = \quad & Close(t) > plevel \quad \wedge \\ & Close(t-1) \leq plevel \end{aligned} \quad (14)$$

and

$$plevel = \begin{cases} l: & \text{if at least } x_2 \text{ peaks in } Close \text{ that} \\ & \text{differs by less than } x_3\% \text{ can} \\ & \text{be identified at level } l \text{ in an } x_1 \text{ -} \\ & \text{day-long window backwards.} \\ 0: & \text{otherwise} \end{cases} \quad (15)$$

The *xresist* function is illustrated in Figure 2 with $x_1 = 160, x_2 = 2$ and $x_3 = 1$.

2.5 Performance Evaluation

Performance evaluation for a trading rule is needed in two stages of the process. First, in the learning phase, when optimal parameters for the trading rule have to be determined. The second stage is when the final trading rule is evaluated on the test data set previously unseen. For more information about performance evaluation refer to Hellström [6] or Refenes [19].

Trading-rule-based methods are normally evaluated by trading simulation, where the trading rule controls the buying and selling of one or several stocks over a period of time. Examples of this approach in conjunction with optimization can be found in Hellström, Holmström [9]. However, it is also possible to evaluate a trading rule with a fixed prediction horizon, of which the advantage is that all situations where the trading rules fire (i.e.: $T(t) \neq$ *Do Nothing* in (1)) are evaluated. When performing a trading simulation, this is normally not the case, since the simulated trader is bounded by the real-world constraint of a limited amount of money. This prevents the trader

from executing some of the *Buy* signals that the trading rule produces. Since the fraction of left-out trades can be as high as 80-90%, a scheme with randomization and repeated simulations is normally required to produce reliable performance measures for the trading rules. Therefore in this study we evaluate trading rules at fixed prediction horizons. The measure of interest is the correctness of the sign of the price change from the time of the prediction to 5 days ahead. This way of evaluating predictions has gained increased interest in recent years as an alternative to the more conventional way of minimizing the error of the level prediction. A comparative study of sign and level methods can be found in Leung, Daouk and Chen [17], where the presented experiments suggest that methods predicting the sign provide higher profits than methods predicting the level for a number of investigated stock indexes.

For a time period $[1, ..., T]$ and a set of stocks S, the h-day positive hit rate for a *Buy* rule g is defined as

$$H_g^+ = \frac{card\{(t,s)|R_h^s(t+h) > 0, g_s(t) = 1, (t,s) \in A\}}{card\{(t,s)|R_h^s(t+h) \neq 0, g_s(t) = 1, (t,s) \in A\}} \tag{16}$$

where g_s is the function specifying the trading rule as described in (1) and $A = \{(t,s)|1 \leq t \leq T-h, s \in S\}$. The return R_h^s is the relative change in price and is defined as

$$R_h^s(t) = 100 \cdot \frac{Close_s(t) - Close_s(t-h)}{Close_s(t-h)} \tag{17}$$

where $Close_s(t)$ is the price for a stock s at the end of day t. The hit rate H_g^+ for a *Buy* rule g indicates how often a *Buy* signal is followed by a true increase in the stock price. The hit rate H_g^- for a *Sell* rule is defined correspondingly but with returns $R_h < 0$.

3 Optimization

For a general trading rule, finding the optimal function g in (2) can be defined as an inductive learning task that can be solved using historical data. In the case of standard technical indicators, the function g is normally parameterized with a few parameters x that have to be determined in order to maximize the chosen performance measure on the historical data. To express this parameterization, the notation $g[x]$ will be used in this section.

One big problem about trading rules in general and optimizing them in particular is the statistical significance of the estimated performance. The trading rule (1) normally issues *Buy* or *Sell* signals only for a minor part of the points in the time series. This results in low levels of significance for the produced performance measures. It is often easy to find a trading rule that historically outperforms any benchmark, as long as it does not have to produce more than a few signals per year. However, the performance on previously unseen data is most often very bad in these situations. We therefore formulate a constrained optimization problem for a *Buy rule* g (*Sell* rules can be treated in a similar way) as

$$\begin{aligned} &\arg\ \underset{x}{max} H_{g[x]}^+ \\ &\text{s.t.} \\ &card\{(s,t)|g_s[x](t) = 1, t \leq T-h,\ s \in S\} \geq N_0, \\ &x_L \leq x \leq x_H \end{aligned} \tag{18}$$

where x_L and x_H are lower and upper bounds for the unknown parameters and the other constraint is the total number of *Buy* signals. The hit rate $H_{g[x]}^+$ is given by definition (16). With the introduced notation, $g_s[x](t)$ denotes the trading rule g parameterized with parameters x and applied to stock s for time t. The optimization routine performs simulations up to time T to compute the hit rate and number of trading signals for a given $g[x]$. The purpose is to maximize the hit rate $H_{g[x]}^+$ by altering the variables x that parameterize the function g. The final performance measure is the out-of-sample hit rate $H_{g[x]}^+$, computed for time $t > T$ with the optimal estimated parameters x.

Using a 'hard' constraint in the optimization problem in (18) leads to a nonsmooth problem. Because of the uncertainty in the choice of the 'most' suitable value of N_0, it is reasonable to reformulate the problem using a 'soft' constraint approach that generates a smooth problem. The approach uses a sigmoid function to smoothly model the behavior of the added constraint and is inspired by the membership-function concept used in fuzzy logic (Klir, Yuan [15]). The new problem formulation, in which the objective function in (18) is weighted with the output of a sigmoid, is

$$\begin{aligned} &\arg\ \underset{x}{max}\ \ H_{g[x]}^+ \cdot \\ &s_{N_0}\left(card\{(s,t)|g_s[x](t) = 1, t \leq T-h, s \in S\}\right) \\ &\text{s.t.} \\ &x_L \leq x \leq x_H \end{aligned} \tag{19}$$

where s_{N_0} is given by the sigmoid function

$$s_{N_0}(n) = \frac{1}{1 + e^{-\alpha(n-\beta)}}. \tag{20}$$

The parameters α and β are computed to fulfill the equations $s_{N_0}(N_0) = 0.99$ and $s_{N_0}(N_0 \cdot 0.5) = 0.01$. This ensures a smooth penalty for trading rules that generate less than N_0 trading signals. If more than

N_0 trading signals are generated, the s_{N_0} function returns essentially 1 and hence does not affect the search for an optimal function g. The constraint acts like a regularizer, since the search space for the function g is reduced by requiring a minimum number of trading signals. This improves the statistical significance of the estimated performance and the generalizability of the found solution (i.e. the achieved hit rate on previously unseen data). The choice of the cut-off value N_0 is a trade-off between the achieved hit rate on the training data and the generalizability.

The optimization problem (19) is a box-bounded nonconvex global optimization problem. It is suitable to use derivative free methods, since no analytical expressions for $g[x]$ and $H^+_{g[x]}$ are available. In our tests we are using the *DIRECT* algorithm by Jones [12] as implemented in the TOMLAB optimization environment [1, 10]. The *DIRECT* algorithm operates on a grid covering the input space. The Lipschitz constant is viewed as a weighting parameter that indicates how much emphasis to put on global versus local search.

Technical analysis of stocks is normally based on the premise that the market's behavior does not change much over time. While future movements in stock prices are never copies of the past, the market's way of responding to new situations is assumed to be similar to the way it has handled them in the past. Since this is not necessarily a valid assumption the optimization will be performed with a sliding window technique as described in the next Section.

4 Empirical Tests

The hit rate H^+_g in the object function (18) is computed using the non-interactive version of the ASTA system, which performs market simulations of trading rules given in symbolic form. The ASTA system is written in Matlab and has a large number of technical indicators implemented. The system is thoroughly described in Hellström [8]. Examples of usage are found in Hellström [7].

4.1 Test Procedure

The test is utilizing a sliding-window technique with a 2-year training data period followed by a 1-year test period. The starting point of the training period is moved between 1990 and 1995 in 1-year steps. This results in six separate modeling/test periods. The presented performance is the total for the six test periods (1992,...,1997). The purpose of using sliding windows in the optimization is twofold. First, the stability in the performance can be studied since we get six performance measures instead of one. Second, the trading rules are allowed to adapt to time-varying market conditions such as volatility, long-term trends etc. The 32 largest Swedish stocks are included in the test, which provides a total number of data points of around 35000 (not all stocks have data for the entire period). The trading rules select a small fraction of these points (date and stock) as suggested opportunities to buy stocks.

4.2 Test Results

The results for 5-day prediction horizon are presented in Table 1 , with positive hit rate H^+ and number of points N where a trading signal is generated. Separate measures for training data and test data are presented in the columns labeled H_{tr}, N_{tr}, H_{te} and N_{te}. The rightmost column shows the lower 90% confidence limit[1] for the hit rate H_{te}. The cut-off value N_0, used for the regularization, is set to 100. Each of the eight rows represents a prediction method. The first three rows show the results for the trading rules $resist_{100}$, $break_{100}$ and mav_{100} described in Section 2. The parameters $x_1, x_2, ...$ are optimized for best performance on the training data, using the regularization described above ($N_0 = 100$). The following three rows show the same trading rules as above, but with no regularization to control the number of generated trading signals ($N_0 = 1$): $resist_1$, $break_1$ and mav_1. Performance for the benchmark methods Naive-5_+ and Naive-ε are also reported. The Naive-5_+ predictor of the returns for a stock s asserts today's return $R^s_5(t)$ (price increase since $t-5$) as the prediction of $R^s_5(t+5)$. The Naive-ε prediction of prices for a stock s asserts today's price $Close_s(t)$ as the best estimate of $Close_s(t+5)$. To enable comparison of hit rate predictions, the naive predictor is modified so the best estimate of today's price is assumed to be $Close_s(t+5)+\varepsilon$. This means that the predicted returns R^s_5 are always positive. This naive predictor is denoted below *Naive-ϵ.*

As expected, the optimized trading rules perform much better for the training data than for the test data. This effect is much more emphasized for the non-regularized trading rules than for the regularized ones. The difference can be understood as over-fitting of data that can be controlled by the regularization. The out-of-sample hit rates H_{te} show no systematic difference between the two kinds of predictors. The small observed differences should be seen rather as stochastic fluctuations caused by the low accuracy in the estimation of the hit rates for the non-regularized trading rules. The lower 90% confidence limit reveals

[1] The lower boundary for a 90% confidence interval.

how uncertain the hit rates H_{te} are for these rules. This uncertainty comes from the low number of predictions generated. None of these trading rules can be said to significantly outperform the benchmark predictors, while all the regularized predictors exhibit a significantly higher hit rate than the benchmarks.

4.3 Stability of the Found Optima

The experimental setup with sliding windows gives a stable evaluation of the trading rules. In this section an additional test of the stability and relevance of the optimized trading rules is performed. In Table 2, the three regularized trading rules optimized with data from 1990-1991 are applied not only for 1992 but also for the following years up to 1997. This means that the optimized rules are regarded as globally valid instead of valid only for the year following the optimization period. The results show that the average hit rate for the trading rules for the six years, is clearly lower than the one achieved by the sliding-window approach, as shown in Table 1 (the relevant value for comparison is shown in column H_{te}). Furthermore, the individual results for each year show no clear tendency and can be regarded as random variations. These observations give further credibility to the sliding-window results and show that the optimizations really are catching patterns and regularities in the data and not only spurious local optima in random and noisy object functions.

5 Conclusions

The optimized trading rules give a significantly higher hit rate than the benchmark methods when tested out-of-sample. The three regularized 5-day predictors give hit rates for the sign between 59% and 64%, while the benchmark methods give less than 53%. The 3 non-regularized predictors produce comparable results, but the risk-compensated hit rate is much lower and not significantly better than the benchmarks.

The constrained optimization that avoids too few selected points is essential both for practical reasons (since we want to get assistance in our buy and sell decisions more than a few times per year), and for a reasonably safe estimate of the expected hit rate out-of-sample. Without safeguarding against too few points, the found optima gives excellent performance on the training data, but no significant improvement relative to pure chance on the test data. Furthermore, the results show that the high hit rates achieved, to a large extent are a result of the adaptive modeling with sliding windows. Techniques such as the ones shown in this paper are therefore a vital part of both development and evaluation of optimized trading strategies.

References

[1] M. Björkman and K. Holmström. Global Optimization Using the DIRECT Algorithm in Matlab. *Advanced Modeling and Optimization*, 1(2):17–37, 1999.

[2] L. Blume, D. Easley, and M. O'Hara. Market statistics and technical analysis: The role of volume. *Journal of Finance*, 49:153–181, 1994.

[3] W. Brock, J. Lakonishok, and B. LeBaron. Simple technical rules and the stochastic propertites of stock returns. *Journal of Finance*, 47:1731–1764, 1992.

[4] J. Y. Campbell, S. J. Grossman, and J. Wang. Trading volume and serial correlation in stock returns. *Quarterly Journal of Economics*, 108:905–940, 1993.

[5] T. Hellström. *A Random Walk through the Stock Market*. Licentiate thesis, Umeå University, Umeå Sweden, 1998.

[6] T. Hellström. Data Snooping in the Stock Market. *Theory of Stochastic Processes*, 5(21)(1-2):33–50, 1999.

[7] T. Hellström. ASTA - a Tool for Development of Stock Prediction Algorithms. *Theory of Stochastic Processes*, 5(21)(1-2):22–32, 1999.

[8] T. Hellström. ASTA - User's Reference Guide. Technical Report UMINF-00.16 ISSN-0348-0542, Department of Computing Science Umeå University, Umeå Sweden, 2000.

[9] T. Hellström and K. Holmström. Parameter Tuning in Trading Algorithms using ASTA. In Y. S. Abu-Mostafa, B. LeBaron, A. W. Lo, and A. S. Weigend, editors, *Computational Finance 1999*, pages 343–357, Cambridge, MA, 1999. MIT Press.

[10] K. Holmström. The TOMLAB Optimization Environment in Matlab. *Advanced Modeling and Optimization*, 1(1):47–69, 1999.

[11] Donald L. Iglehart and Siegfried Voessner. Optimization of a trading system using global search techniques and local optimization. *Journal of Computational Intelligence in Finance*, 6:36–46, 1998.

[12] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, October 1993.

[13] J. M. Karpov. The relation between price changes and traded volume. *Journal of Financial and Quantitative Analysis*, 22:109–126, 1987.

[14] Perry J. Kaufman. *Trading Systems and Methods*. John Wiley and Sons, New York, 1998.

[15] G.J. Klir and Bo Yuan. *Fuzzy Sets and Fuzzy Logic. Theory and Applications*. Prentice-Hall, Inc, New Jersey, USA, 1995.

[16] G. W. Kuo. Some exact results for moving-average trading rules with applications to UK indices. In E. Acar and S. Satchell, editors, *Advanced Trading Rules*, pages 81–102. Butterworth Heinemann, Oxford, 1998.

[17] Mark T. Leung, Hazem Daouk, and An-Sing Chen. Forecasting stock indices: a comparison of classification and level estimation methods. *International Journal of Forecasting*, 16:173–190, 2000.

[18] R. M. Levich and L. R. Thomas. The significance of technical trading-rule profits in the foreign exchange market: a bootstrap approach. *Journal of International Money and Finance*, 12:451–474, 1993.

[19] Apostolos-Paul Refenes. Testing strategies and metrics. In Apostolos-Paul Refenes, editor, *Neural Networks in the Capital Markets*, chapter 5, pages 67–76. John Wiley & Sons, Chichester, England, 1995.

Table 1: Hit rate and number of selected points for optimized trading rules. Totals from 6 1-year test periods (1992-1997) with the preceding 2 years for training. 5-day prediction horizon.

Method	H_{tr}	N_{tr}	H_{te}	N_{te}	**90%−low** H_{te}
$resist_{100}$	63.61	753	58.72	516	55.03
$break_{100}$	68.36	708	64.22	450	60.33
mav_{100}	65.62	701	61.21	397	57.01
$resist_{1}$	86.27	51	64.29	28	47.00
$break_{1}$	73.56	295	56.32	190	50.09
mav_{1}	85.11	94	61.70	47	48.67
$Naive-\varepsilon$	50.83	80456	52.35	42372	51.95
$Naive-5_{+}$	52.01	37519	52.17	20539	51.59

Table 2: Hit rate for trading rules optimized with data from 1990-1991. 5-day prediction horizon.

Method	92	93	94	95	96	97	Average	H_{te}
$resist_{100}$	54.7	58.7	51.0	56.6	52.5	53.5	54.6	58.72
$break_{100}$	72.2	66.0	62.7	51.1	58.3	55.0	59.8	64.22
mav_{100}	56.7	70.3	53.9	51.2	61.5	59.7	58.7	61.21

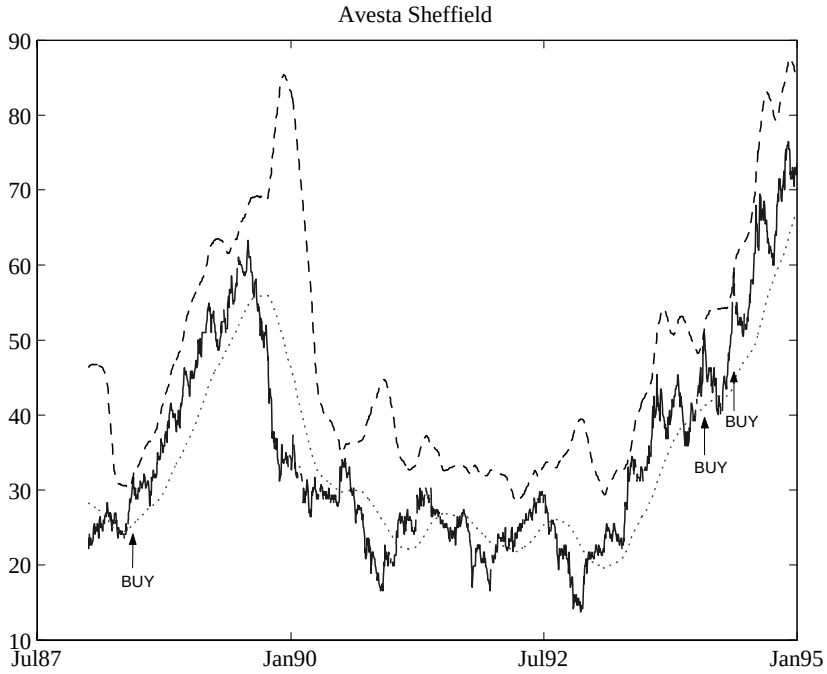

Figure 1: A trading rule based on breakout of trading channels. The solid line is stock price. The dotted line is a 120 days moving average and the dashed line is the moving average plus 3.5 standard deviations.

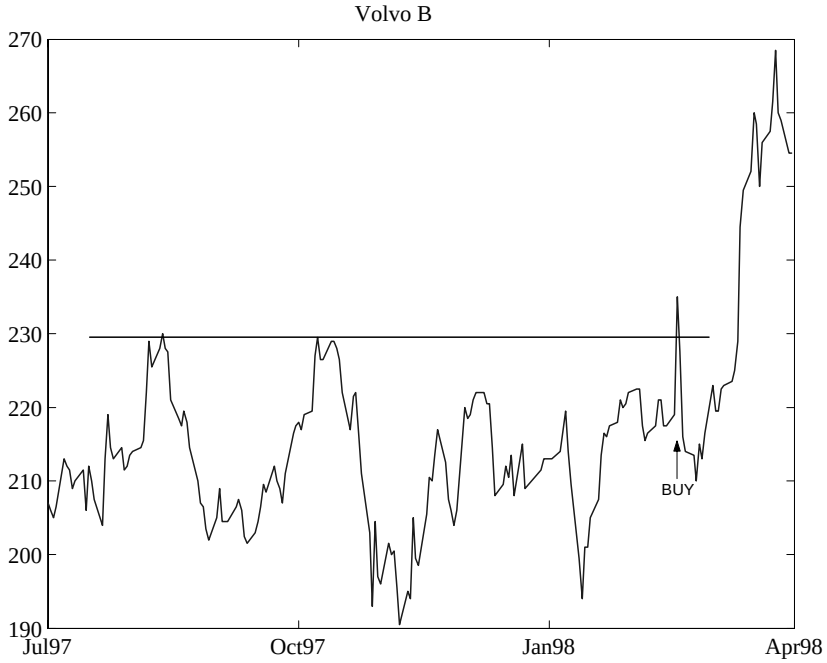

Figure 2: The shown trading rule generates Buy signals when the price crosses a level which has acted as a resistance twice during the last 160 days.