Post processing trading signals for improved trading performance

Thomas Hellström[1], Luis Torgo[2]
[1]*Department of Computing Science,*
Umeå University, Sweden.
[2]*LIACC/FEP, University of Porto, Portugal.*

Abstract

A trading strategy is an algorithm that provides decision support for a trader. An ideal system suggests which stocks to buy and sell at every moment. Limited but still very useful trading strategies suggest stocks to buy, but leave the sell decisions and the decision of proportions of different stocks to the trader, or to another automatic decision mechanisim. In this paper we use a previously introduced method of predicting rank variables to produce both buy and sell decisions. The rank variables are predicted by neural networks, and provide an efficient way to produce daily buy (and also sell) suggestions. This should be seen in contrast to "ordinary" technical indicators that often give very few signals, or buy/sell signals for many stocks at the same time. The produced buy signals are further processed in a classification module that aims at identifying which of the numerous buy signals one should trust, and of which ones one should discard. The classification reduces the number of buy signals and also increases both hitrate and overall profit for a simulated trader. Data from the US stock market for 1992-2001 is used in the tests of the system, and the results show how a trading system's performance can be significantly improved by adding a post-processing classification layer between the generation of trading signals and the actual decision making.

1 Introduction

Technical analysis is a commonly used method among practitioners, to generate buy and sell signals for stock trading. Statistical studies of the relevance of technical analysis have most often shown negative results, and the dominating academic view is that technical analysis simply does not work. However, there are also papers claiming that technical analysis, under certain circumstances, can be used for successful trading. In [6], technical indicators are dynamically optimized with a sliding window technique, and the results show that the performance is significantly improved by looking beyond static technical indicators. In the present paper we use a previously introduced method of rank predictions [8] to produce buy and sell signals for stock trading. The rank variables are predicted by neural networks and provide an efficient way to produce daily buy (and also sell) signals. To improve the quality of the buy signals, they are further processed in a classification module that tries to identify which ones of the buy signals one should trust, and which ones one should scrap.

The selected buy rules are combined with sell rules to a complete trading system, which is evaluated in simulated trading with the ASTA system. A technical overview of ASTA can be found in Hellström [8] and examples of usage in Hellström [5] and Hellström, Holmström [9]. More information can also be found at http://www.cs.umu.se/~thomash.

The data in the study comes from 599 current or former S&P 500 stocks on the American stock market, from January 1, 1992 till December 31, 2001. The results show how a trading system's performance can be significantly improved by adding a post processing classification layer between the generation of trading signals and the actual decision making.

In Section 2 the rank measure is introduced. In Section 3, neural models for prediction of the rank are defined, and historical data is used to estimate the weights in the neural nets. Results from time series predictions are presented. The predictions are used as a basis for a Decision Support System for stock picking, described in Section 4. The trading signals are filtered in a classifier system, based on regression trees, and is described in Section 4.1. Section 5 contains a summary of the results together with ideas for future research.

2 Stock Ranks

The returns of individual securities are the primary targets in most research that deal with the predictability of financial markets. The rank concept focuses on the observation that a real trading situation involves not only attempts to predict the individual returns for a set of interesting securities, but also a comparison and selection among the produced predictions. What an investor really wants to have is not a large number of predictions for individual returns, but rather a grading of the securities in question. Even if this can be achieved by grading the individual predictions of returns, it is not obvious that it will yield an optimal decision based on a limited amount of noisy data.

The k-day return $R_k(t)$ for a stock m with close prices $y^m(1), ..., y^m(t_1)$ is defined for $t \in [k+1, ..., t_1]$ as

$$R_k^m(t) = \frac{y^m(t) - y^m(t-k)}{y^m(t-k)}. \tag{1}$$

We introduce a rank concept A_k^m, based on the k-day return R_k as follows: The k-day rank A_k^m for a stock s_m in the set $\{s_1, ..., s_N\}$ is computed by ranking the N stocks in the order of the k-day returns R_k. The ranking orders are then normalized, so the stock with the lowest R_k is ranked -0.5 and the stock with the highest R_k is ranked 0.5. The definition of the k-day rank A_k^m for a stock m belonging to a set of stocks $\{s_1, ..., s_N\}$, can thus be written as

$$A_k^m(t) = \frac{\#\{R_k^i(t) | R_k^m(t) \geq R_k^i(t), 1 \leq i \leq N\} - 1}{N - 1} - 0.5 \qquad (2)$$

where the $\#$ function returns the number of elements in the argument set. This is an integer between 1 and N. R_k^m is the k-day returns, computed for stock m. The scaling between -0.5 and $+0.5$ assigns the stock with the median value on R_k the rank 0. A positive rank A_k^m means that stock m performs better than this median stock, and a negative rank means that it performs worse. This new measure gives an indication of how each individual stock has developed relatively to the other stocks, viewed on a time scale set by the value of k.

The scaling around $zero$ is convenient when defining a prediction task for the rank. It is clear that an ability to identify, at time t, a stock m, for which $A_h^m(t + h) > 0, h > 0$, means an opportunity to make excess profit relative to the market. A method that can identify stocks m and times t with a mean value of $A_h^m(t + h) > 0, h > 0$, can be used as a trading strategy that can do better than the average stock. The hitrate for the predictions can be defined as the fraction of times, for which the sign of the predicted rank $A_h^m(t + h)$ is correct. A value greater than 50% means that true predictions have been achieved. The following advantages compared to predicting returns $R_h(t + h)$ can be noticed:

1. The benchmark for predictions of ranks $A_h^m(t+h)$ becomes clearly defined: A hitrate $> 50\%$ for the predictions of the sign of $A_h^m(t + h)$ means that we are doing better than chance. When predicting returns $R_h(t+h)$, the general positive drift in the market causes more than 50% of the returns to be > 0, which means that it is hard to define a good benchmark. Furthermore, a positive mean value for predicted positive ranks $A_h(t + h)$ (and a negative mean value for predicted negative ranks) means that we are doing better than chance. When predicting returns $R_h(t + h)$, the general positive drift in the market causes the returns to have a mean value > 0. Therefore, a mere positive mean return for predicted positive returns does not imply any useful predicting ability.
2. The rank values $A_k^1(t), ..., A_k^N(t)$, for time t and a set of stocks $1, ..., N$ are uniformly distributed between -0.5 and 0.5 provided no return values are equal. Returns R_k^m, on the other hand, are distributed with sparsely populated tails for the extreme low and high values. This makes the statistical analysis of rank predictions safer and easier than predictions of returns.
3. The effect of global events gets automatically incorporated into the predictor variables. The analysis becomes totally focused on identifying deviations from the average stock, instead of trying to model the global economic situation.

3 Predicting the Ranks

For a stock m, we attempt to predict the h-day-rank h days ahead by fitting a function g_m so that

$$\hat{A}_h^m(t+h) = g_m(I_t) \tag{3}$$

where I_t is the information available at time t. I_t may, for example, include stock returns $R_k^m(t)$, ranks $A_k^m(t)$, traded volume, etc. The prediction problem 3 is as general as the corresponding problem for stock returns, and can be attacked, of course, in a variety of ways. Our choice in this first formulation of the problem assumes a dependence between the future rank $A_h^m(t+h)$ and current ranks $A_k^m(t)$ for different values of k. I.e.: a stock's tendency to be a *winner* in the future depends on its *winner* property in the past, computed for different time horizons. This assumption is inspired by the autocorrelation analysis in Hellström [7], and also by previous work by De Bondt, Thaler [1] and Hellström [5] showing how these dependencies can be exploited for prediction and trading. Confining our analysis to $1-, 2-, 5-$,and $20-$day-horizons, the prediction model 3 is refined to

$$\hat{A}_h^m(t+h) = g_m(A_1^m(t), A_2^m(t), A_5^m(t), A_{20}^m(t)). \tag{4}$$

The choice of function g_m in this paper is a feed-forward neural network. The network is trained using historical data. For a market with N stocks, N separate networks are built, each one denoted by the index m. The h-day rank A_h^m for time $t+h$ is predicted from the 1-day, 2-day, 5-day and 20-day ranks, computed at time t. To facilitate further comparison of the m produced predictions, they are ranked in the same way as in the definition of the ranks themselves:

$$\hat{A}_h^m(t+h) \leftarrow \frac{\#\{\hat{A}_h^i(t) | \hat{A}_h^m(t) \geq \hat{A}_h^i(t), 1 \leq i \leq N\} - 1}{N-1} - 0.5. \tag{5}$$

In this way the N predictions $\hat{A}_h^m(t+h), m = 1, ..., N$, get values uniformly distributed between -0.5 and 0.5 with the lowest prediction having the value -0.5 and the highest one the value 0.5.

The complete data set covers the years 1992-2001. We have used a sliding window technique, where 500 points (about 2 trading years) are used for training and the following 100 are used for prediction. The window is then moved 100 days (approx. 5 trading months) ahead and the procedure is repeated until end of data. Since 500 points are needed for the modeling, the predictions are produced for the years 1994-2001.

3.1 Evaluation of the Rank Predictions

The computed models $g_m, m = 1, ..., N$ at each time step t produce N predictions of the future ranks $A_h^m(t+h)$ for the N stocks. The N predictions $\hat{A}_h^m, m = 1, ..., N$, are evenly distributed by transformation (5) in $[-0.5, ..., 0.5]$. As we shall see in the following section, we can construct a successful trading system utilizing only a few of the N predictions. Furthermore, even viewed as N separate predictions, we have the freedom of rejecting predictions if they are not viewed as reliable or profitable. By introducing a cut-off value γ, a selection of predictions can be made. For example, $\gamma = 0.49$ means that we are only considering predictions $\hat{A}_h^m(t+h)$ with $|\hat{A}_h^m(t+h)| > 0.49$.

The results for 1-day predictions of 1-day ranks $\hat{A}_1^m(t+1)$ for γ between 0 and 0.5 are presented in Figures 2 and 3. Figure 2 shows the hitrate for the positive (solid curve) and negative (dotted curve) rank predictions. The hitrate for the positive predictions is the fraction of predictions $\hat{A}_h^m(t+h) > \gamma$, with the correct sign. A value significantly higher than 50% means that we are able to identify higher-than-average performing stocks better than chance. The hitrate for the negative predictions is the fraction of predictions $\hat{A}_h^m(t+h) < -\gamma$, with the correct sign. A value significantly higher than 50% means that we are able to identify lower-than-average performing stocks better than chance. For cut-off value $\gamma = 0$, the hitrate is 51.5% for both positive and negative rank predictions. By increasing γ to 0.49, the hitrate goes up to 57% for predicted negative ranks.

Figure 3 shows the 1-day return for the positive (solid curve) and negative (dotted curve) rank predictions. The 1-day return for the positive predictions is defined as $100\cdot$Mean value of the 1-day returns $R_1^m(t+1)$ for predictions $\hat{A}_1^m(t+1) > \gamma$. The difference between the mean returns for positive and negative rank predictions shows that the sign of the rank prediction really separates the returns significantly. For cut-off value $\gamma = 0.49$, positive predictions of ranks in average are followed by a return of $+0.5\%$, while a negative rank prediction in average is followed by a return of -0.33%. All presented values are average values over time t and over all involved stocks m.

4 Building a Trading System

The rank predictions are used as basis for a Decision Support System for stock picking. The layout of the trading system is given by the two top blocks in Figure 1. The 1-day predictions $\hat{A}_1^m(t+1)$ are fed into a decision maker that generates buy and sell signals. The module "Generation of trading signals" uses the 1-day rank predictions $\hat{A}_1^m(t+1)$ in combination with the cut-off value γ set to 0.49. A buy signal is generated if $\hat{A}_1^m(t+1) > 0.49$, and otherwise a sell signal is generated (of course provided the stock in question is in the portfolio). This results in 5 to 6 buy signals being generated every day, which may be used directly as trading signals. The result of a simulated trading, based on the described buy and sell rules, is presented in the top three lines of Table 1. For the simulation, the ASTA system [8] is used. The system uses daily data, where the trading decision is based on each day's close price, and the actual trading is done using the open price for the following day. The commission is set to 50 USD per trade and the trader's initial capital is set to 100.000 USD. To enable a comparison with an extended strategy, the simulation is carried out for the years 1999-2001. As we can see from Table 1, the result is far from satisfying. The trader actually goes bankrupt after a little bit more than one year. An analysis shows that the high commission combined with the high number of trades (2075 for the year 1999) are the reasons for this failure. The performance improves if the initial capital is increased, but for a limited amount of capital it is clear that a very high number of trades is not always a desirable property of a trading system. In the next section, this problem is addressed by a classification module operating on the produced buy signals.

To improve the performance we now try to reduce the high number of trades by adding a classifier to the trading system (Figure 1). The classifier described in Section 4.1, uses additional technical information to filter out those buy signals that really result in positive returns. In this way, the hitrate increases and the total number of buy signals is reduced.

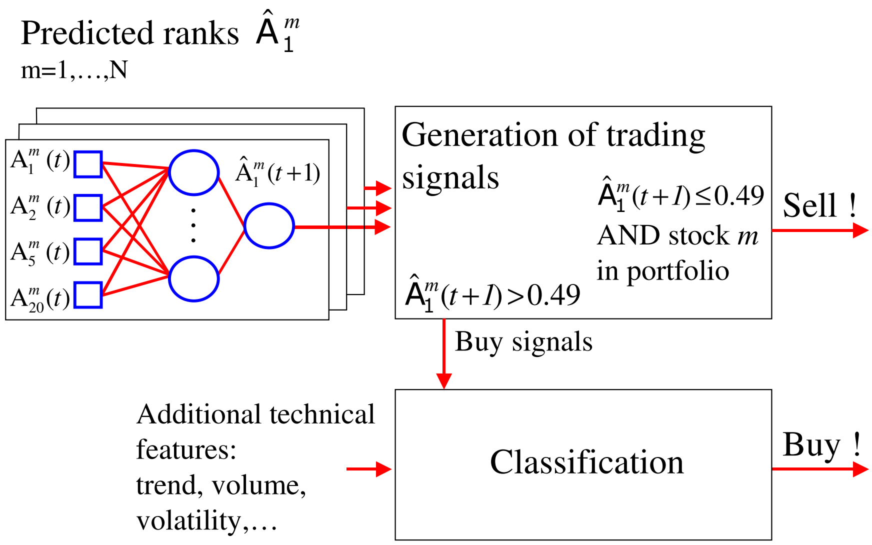


Figure 1: Architecture for trading system based on rank predictions. The stocks with the highest predicted ranks are passed on to a classifier. The classifier attempts to identify the succesful buy signals and blocks the other signals.

4.1 Classifying Buy Signals through Partial Linear Trees

The classification schema used to reduce the high number of trades coming out of the rank-based predictions is based on partial linear trees [13, 14]. The method used to select the “trustable” buy signals uses a prediction of 1-day returns, $\hat{R}_1^m(t+1)$, for the particular stock, which got a 1-day rank prediction higher than 0.49, *i.e.*

$\hat{A}_1^m(t+1) > 0.49$. If this predicted 1-day return is higher than a certain threshold, we consider the signal "trustable", otherwise we do not buy the security.

Partial linear trees result from the integration of partial linear models in the leaves of a regression tree. The main motivation for this integration is to try to achieve the predictive accuracy of partial linear regression (*e.g.* [11, 3]), whilst maintaining the interpretability of tree-based models. Further details on both the method of integration and the obtained results in terms of accuracy and interpretability can be found in Torgo [13] and Torgo [14].

Partial linear regression [11] is a semiparametric regression technique that integrates a standard least-squares linear polynomial with a kernel smoother [10, 15]. Given a query case for which we want a prediction, these models obtain it by summing the value predicted by a linear polynomial with the value resulting from smoothing (averaging) the errors of the polynomial in the neighboring training sample observations (*i.e.* the most similar training cases). The more inadequate the linear polynomial is versus the given training sample, the larger the importance of the smoothing component. In the extreme case where the polynomial perfectly fits the training data, a partial linear model is reduced to a standard least squares linear polynomial.

Regression trees (*e.g.* [2]) handle multiple regression methods, obtaining models that have proven to be quite interpretable and having competitive predictive accuracy. Moreover, these models can be obtained with a computational efficiency that hardly has parallel in competitive approaches. A regression tree can be seen as a kind of additive regression model [4] of the form,

$$rt\,(x) = \sum_{i=1}^{l} k_i \times I\,(x \in D_i) \tag{6}$$

where $k_i's$ are constants; $I\,(.)$ is an indicator function returning 1 if its argument is true and 0 otherwise; and $D_i's$ are disjoint partitions of the training data D so that $\bigcup_{i=1}^{l} D_i = D$ and $\bigcap_{i=1}^{l} = \phi$.

These models are sometimes called piecewise constant regression models. Using these constants $k_i's$ at the leaves of the trees has shown its limitations in terms of accuracy and smoothness of the function approximation [12]. Partial linear trees are one of the possible ways of overcoming these difficulties. By using partial linear models at the leaves, we are able to obtain smoother and more accurate tree-based models.

In our experiments we have used partial linear trees as implemented in system RT. RT is freely available for download at http://www.liacc.up.pt/~ltorgo. Using the data from years 1992-1998 we have obtained a partial linear tree. This model was then used to predict the 1-day return for the stocks with a 1-day rank prediction higher than 0.49, for the years 1999-2001. If this predicted return $\hat{R}_1(t+1)$ was higher than a certain threshold we have a "trustable" buy signal, otherwise we ignore the buy signal of the rank predictor.

4.2 Trading Performance

The annual trading profit for the complete trading system with classification, is presented in rows 4-6 in Table 1. As can be seen, the performance is very good. The trading strategy outperforms the benchmark consistently and significantly every year and the mean annual profit is 89.6%. The benchmark index is a computed uniform stock index, based on all stocks that have data for a given day. Trading this index would give a mean annual profit of 16.4%.

The Sharpe ratio is another entity of importance when evaluating trading systems. The Sharpe ratio is the annualized profit divided by the annualized standard deviation of the trader's wealth over time. This value should obviously be as high as possible for a system with a combined high profit and low risk. In the example, the average Sharpe ratio for the simulated trader is 1.5, while it is 0.9 for the benchmark index. Furthermore, the number of trades every year has been reduced to 267 per year in average, i.e. roughly one buy or sell order per day. The same trading results are also displayed graphically in Figure 4. The upper diagram shows the equity curves for the trading strategy and for the benchmark index. The lower diagram shows the annual profits as histograms. The equity curves illustrate how the capital, initially scaled to 1, changes over time. While being a popular way of presenting trading results, the equity curve is most often less informative than the yearly histograms.

5 Conclusions

We have successfully implemented a trading system where trading rules based on a model for prediction of ranks are combined with a classifier module that reduces the number of trades, while increasing both the hitrate and the profit for the individual trades. The results show how dynamic computation of trading signals in combination with a post-processing step can improve profit without reducing the Sharpe ratio below that of the benchmark. The mean annual profit for the trading system with a classifier is 89.6% compared to 16.4% for the benchmark portfolio, over the investigated 3-year period. The risk-adjusted return, as measured by the Sharpe ratio, is 1.5 for the trading system, while trading the benchmark portfolio gives only 0.9.

Of course, the general idea of adding a post-processing layer that classifies buy signals can be applied to other kinds of technical indicators. It would also be interesting to apply the classification to not only the buy signals, but also the sell signals.

References

[1] W. D. Bondt and R. Taler. Does the stock market overreact? *Journal of Finance*, 40(3):793–805, 1985.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and*

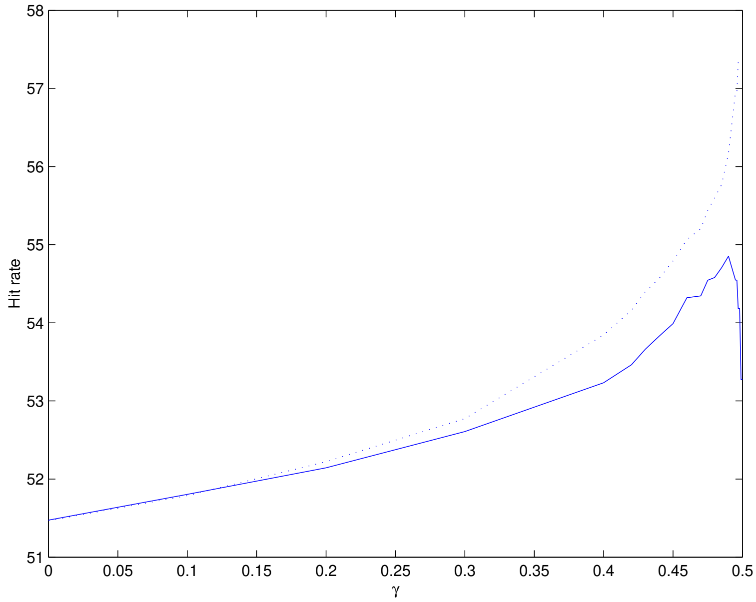


Figure 2: Hitrate for selected rank predictions.

Regression Trees (CART). Wadsworth, Pacific Grove, CA, 1984.

[3] W. Hardle. *Applied Nonparametric Regression*. Cambridge University Press, 1990.

[4] T. R. Hastie, T. *Generalized Additive Models*. Chapman & Hall, 1990.

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

[6] T. Hellström. Optimizing trading rules with a penalty term for increased risk-adjusted performance. *Advanced Modeling and Optimization*, 2(3):135–149, 2000.

[7] T. Hellström. Predicting a Rank Measure for Portfolio Selection. *Theory of Stochastic Processes*, 6(22)(3-4):64–83, 2000.

[8] T. Hellström. *Techniques and Software for Development and Evaluation of Trading Strategies*. PhD thesis, Umeå University, Umeå Sweden, 2001.

[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] E. Nadaraya. On estimating regression. *Theory of probability and its applications*, 9:141–142, 1964.

[11] C. Spiegelman. *Two techniques for estimating treatment effects in the presence of hidden variables: adaptive regression and a solution to Reiersol's*

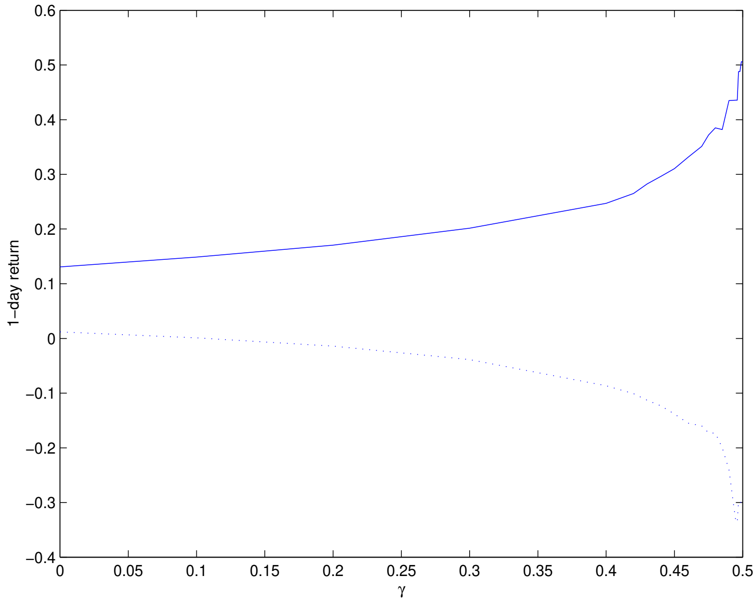


Figure 3: Average 1-day return for selected positive (solid) and negative (dotted) rank predictions.

problem. PhD thesis, Dept. of Mathematics, Northwestern University, 1976.

[12] L. Torgo. Functional models for regression tree leaves. In D. Fisher, editor, *Proceedings of the 14th International Conference on Machine Learning*. Morgan Kaufmann Publishers, 1997.

[13] L. Torgo. *Inductive Learning of Tree-based Regression Models*. PhD thesis, Faculty of Sciences, University of Porto, 1999.

[14] L. Torgo. Partial linear trees. In P. Langley, editor, *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pages 1007–1014. Morgan Kaufmann Publishers, 2000.

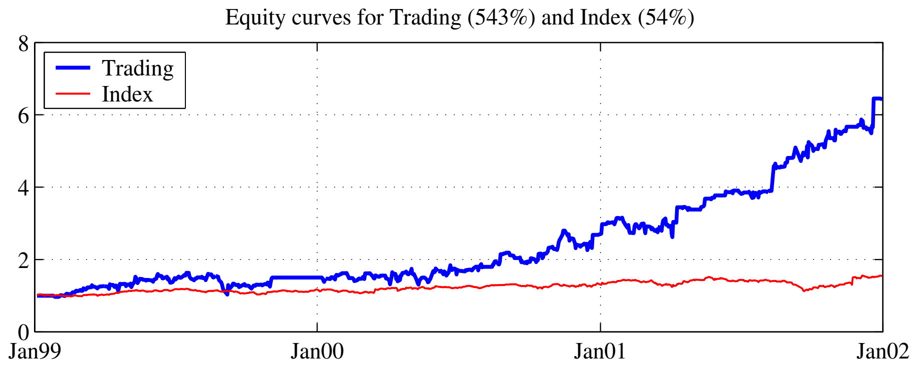
[15] G. Watson. Smooth regression analysis. *Sankhya: The Indian Journal of Statistics*, Series A(26):359–372, 1964.

6 Acknowledgments

The authors are grateful to Vasant Dhar for fruitful discussions and for giving them access to the stock data used for the simulated trading. Luis Torgo is partially supported by project Sol-Eu-Net IST-1999-11495, by PRAXIS XXI plurianual support to LIACC, and by ESPRIT LTR Project METAL 26.357.

Table 1: Performance for trading systems without$_1$ and with$_2$ post-processing for classification of buy signals. A stock index is used as a benchmark.

	1999	2000	2001	Mean
Annual profit$_1$	-76.5	-97.4	0.0	-58.0
Sharpe ratio$_1$	-3.3	-5.8		-4.5
# trades$_1$	2075	455	0	843
Annual profit$_2$	50.1	78.5	140.1	89.6
Sharpe ratio$_2$	1.0	1.3	2.1	1.5
# trades$_2$	168	370	263	267
Annual profit$_{index}$	18.8	14.5	16.0	16.4
Sharpe ratio$_{index}$	1.2	0.7	0.8	0.9



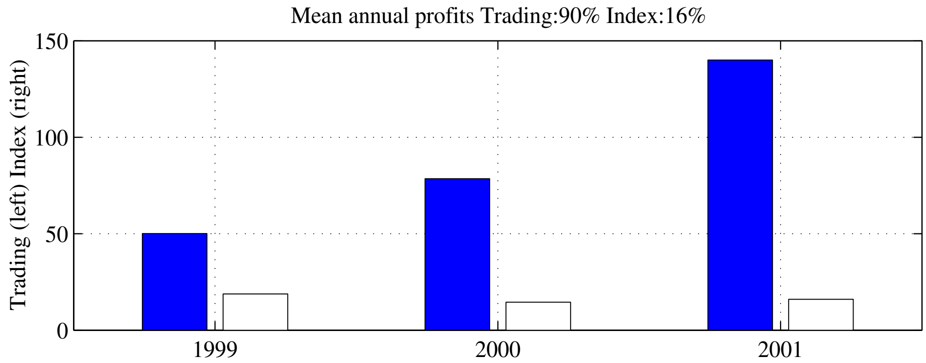


Figure 4: Trading results for trading with classified rank predictions as buy signals. The trader outperforms the benchmark index significantly every year.