Dynamic Financial Forecasting with Automatically Induced Fuzzy Associations

Yazann Romahi and Qiang Shen
{yazannr, qiangs}@dai.ed.ac.uk

School of Artificial Intelligence, Division of Informatics
University of Edinburgh
80 South Bridge, Edinburgh EH1 1HN, UK

Abstract— **The past decade has witnessed significant growth in developing intelligent tools for financial forecasting. Expert Systems were quickly shown to be inadequate for the tasks required in financial forecasting due to their static nature. As a result, interest started to move towards soft computing despite the fact that *comprehensibility* is often of paramount concern in financial forecasting. Merging the domains of fuzzy logic and rule induction paved the way for the emergence of successful generalisation techniques with high comprehensibility. In this paper, we present a novel technique for financial forecasting derived from a fuzzy association induction algorithm, allowing the development of an evolving rule based expert system. In such a way, changing market dynamics are continuously taken into account as time progresses and thus the rulebase does not become outdated. Simulations carried out show promising results for this approach.**

I. Introduction

A typical market tries to establish an equilibrium between the buying and selling forces. This dynamic mechanism of trading has inspired traders to want to outperform the market by predicting what the future trend of the market would be. The financial time series forecasting that results has posed some very interesting problems for researchers.

Artificial Neural Networks (ANN) have drawn a large amount of attention in this area. As well as in financial forecasting [6], [10], they have been applied to a wide range of problems in the financial domain from options pricing [7], [9] to portfolio management [13]. Several disadvantages to ANN approaches are noted, including the fact that there is no single configuration that is adequate for all domains forcing topology determination to proceed by trial and error in a fairly *ad hoc* manner. Another issue where care must be taken is their susceptibility to over-fitting. Most importantly however, from the perspective of using them as a tool in the trading domain, they lack an important attribute: *comprehensibility*. Their *user-opaque* nature forces us to look elsewhere for an approach that could prove potentially useful in financial forecasting and that can also provide a human-readable explanation of its reasoning.

Although rule extraction methods from neural networks do exist [3], considerable work remains before they can be considered mature for real applications. This leads us to look towards fuzzy rule induction as a potential tool for financial time series forecasting. The fuzzification of time series has profound implications for the design of trading models. Based on fuzzified inputs and outputs, trading rules can be identified that embed imprecise ideas. Thus, changes in market indicators and relationships between changing indicators can be captured in general terms.

This paper presents a novel technique for financial forecasting using the fuzzy automatic pattern analysis and classification system (FAPACS) [4], [5]. FAPACS infers fuzzy association rules via an approach based on the "adjusted difference" between observed and expected frequency counts. As a fuzzy logic based algorithm, FAPACS allows the use of linguistic terms to represent the revealed regularities and exceptions, which makes the rules straightforward for a human-user to understand. This is particularly so because the approach used here is concerned with the direct use of predefined fuzzy descriptive sets [8] which may be obtained and augmented by a domain expert.

We introduce the notion of an evolving ruleset to fuzzy rule induction, by training the system on the last n day data (that is, using a moving window of a size n to screen training data) in order to make the prediction for day $n+1$. In such a way, the ruleset continuously evolves as the oldest data is dropped from the training set while the latest is added to it.

The rest of this paper is arranged as follows. Section II provides an overview of the FAPACS approach, showing the association induction process and how inferred associations may be applied. Section III presents a brief description of the application domain. Section IV gives experimental results, demonstrating the potential of the present work. Finally, the paper is concluded in section V.

II. FAPACS

An association rule in FAPACS associates two attribute variables x and y such that

$$\mathcal{L}_{pq} \Rightarrow \mathcal{L}_{jk}$$

where $\mathcal{L}_{pq}$ and $\mathcal{L}_{jk}$ are *linguistic terms* respectively describing the two attributes with their underlying semantics being represented by fuzzy sets.

Traditional algorithms decide on whether or not an association is interesting using two user-defined thresholds [2], [11], [12] namely, the support and the confidence. Given two attributes x and y, the support is defined as the percentage of records having both attributes and the confidence is defined as the percentage of records having y given that they also have x. Whether an association is deemed significant depends on whether both support and confidence are greater than the predefined thresholds. This contains an inherent weakness as it is difficult to ascertain where these thresholds should be.

FAPACS uses *adjusted difference analysis* [4] to identify associations amongst attributes. As a result, it does not require any user-defined thresholds and has a further advantage in that it is possible to infer both *positive* and *negative* associations. Here, a positive association means that an attribute having a certain characteristic defined by $\mathcal{L}_{pq}$ implies that another attribute will have the characteristic $\mathcal{L}_{jk}$. In other words, the existence of an antecedent implies the existence of a consequent. Similarly, a negative result means that the presence of the former implies the absence of the latter.

A. Mining Interesting Associations

The process of association induction begins with the calculation of the sum of degrees to which records in a given training dataset are characterised by the linguistic terms. Each possible combination between a pair of attributes is considered in turn. Given a record $d \in D$, where D refers to the dataset, and linguistic terms $\mathcal{L}_{pq}, \mathcal{L}_{jk} \in \mathcal{L}, p \neq j$, where $\mathcal{L}$ is the collection of linguistic terms which are represented by fuzzy sets (e.g. L_{pq} and L_{jk} with membership functions $\mu_{L_{pq}}(u)$ and $\mu_{L_{jk}}(v)$), the degree of association is accumulated across all records and is used to infer significance. This is done via algorithm 1 (as given in [4]), where $\mathcal{R}$ stands for the collection of generated associations.

Note that the *T-norm* which reflects the degree of association between two membership functions is interpreted using the *min* operator. However, as for the *S-norm* across all the datasets, rather than using the *max*, the addition operator is adopted. This is because it lends itself well for statistical analysis on which the definition of the *interestingness* function is based.

The *interestingness* is defined using *adjusted difference* [4] where

Algorithm 1 Generation of Interesting Associations

```
for all L_pq, L_jk ∈ L, p ≠ j do
  deg_{L_pq L_jk} += min(μ_{L_pq}(u), μ_{L_jk}(v))
end for
for all L_pq, L_jk ∈ L, p ≠ j do
  if interesting(L_pq, L_jk) then
    R ⇐ rulegen(L_pq, L_jk)
  end if
end for
```

$$d_{\mathcal{L}_{pq}\mathcal{L}_{jk}} = \frac{z_{\mathcal{L}_{pq}\mathcal{L}_{jk}}}{\sqrt{\gamma_{\mathcal{L}_{pq}\mathcal{L}_{jk}}}}$$

with $z_{\mathcal{L}_{pq}\mathcal{L}_{jk}}$ being the *standardised difference* given by:

$$z_{\mathcal{L}_{pq}\mathcal{L}_{jk}} = \frac{deg_{\mathcal{L}_{pq}\mathcal{L}_{jk}} - e_{\mathcal{L}_{pq}\mathcal{L}_{jk}}}{\sqrt{e_{\mathcal{L}_{pq}\mathcal{L}_{jk}}}}$$

where $e_{\mathcal{L}_{pq}\mathcal{L}_{jk}}$ is the sum of degrees to which records are expected to be characterised by $\mathcal{L}_{pq}$ and $\mathcal{L}_{jk}$, and is calculated by:

$$e_{\mathcal{L}_{pq}\mathcal{L}_{jk}} = \frac{\sum_{i=1}^{s_j} deg_{\mathcal{L}_{pq}\mathcal{L}_{ji}} \sum_{i=1}^{s_p} deg_{\mathcal{L}_{pi}\mathcal{L}_{jk}}}{\sum_{u=1}^{s_p} \sum_{i=1}^{s_j} deg_{\mathcal{L}_{pu}\mathcal{L}_{ji}}}$$

and $\gamma_{\mathcal{L}_{pq}\mathcal{L}_{jk}}$ is the *maximum likelihood* of the variance of $z_{\mathcal{L}_{pq}\mathcal{L}_{jk}}$, given by:

$$\gamma_{\mathcal{L}_{pq}\mathcal{L}_{jk}} = \left(1 - \frac{\sum_{i=1}^{s_j} deg_{\mathcal{L}_{pq}\mathcal{L}_{ji}}}{\sum_{u=1}^{s_p} \sum_{i=1}^{s_j} deg_{\mathcal{L}_{pu}\mathcal{L}_{ji}}}\right) \times \left(1 - \frac{\sum_{i=1}^{s_p} deg_{\mathcal{L}_{pi}\mathcal{L}_{jk}}}{\sum_{u=1}^{s_p} \sum_{i=1}^{s_j} deg_{\mathcal{L}_{pu}\mathcal{L}_{ji}}}\right)$$

According to [4], if $|d_{\mathcal{L}_{pu}\mathcal{L}_{ji}}| > 1.96$ (i.e. the 95th percentile of the normal distribution), the association is deemed significant or interesting. A positive value of d signals a positive association, while a negative value for d means that the existence of the antecedent implies the absence of the consequent.

If an association is deemed interesting, the *rulegen* function then generates a fuzzy association rule such that it is in the form $\mathcal{L}_{pq} \Rightarrow \mathcal{L}_{jk}[w_{\mathcal{L}_{pq}\mathcal{L}_{jk}}]$, where $w_{\mathcal{L}_{pq}\mathcal{L}_{jk}}$ denotes the *weight of evidence* and means that the existence of the antecedent implies the existence of the consequent to the degree of this weight. The weight of evidence is defined by:

$$w_{\mathcal{L}_{pq}\mathcal{L}_{jk}} = log\frac{Pr(\mathcal{L}_{jk}|\mathcal{L}_{pq})}{Pr(\mathcal{L}_{jk}|\bigcup_{i\neq q}\mathcal{L}_{pi})}$$

Thus, the weight is determined by an estimate of the gain in information when a record is characterised by a particular fuzzy set of a given attribute, L, vs. all fuzzy sets of the same attribute except L.

B. Inference Process

Using the inferred associations, FAPACS can predict or classify previously unseen data. Unlike most classification techniques which categorise records into distinct classes, FAPACS generates a certainty measure for each result of rule-firing.

Consider a record which is characterised by values of n attributes, $\alpha_1, \alpha_2, ..., \alpha_n$. Without losing generality, suppose that the value of $\alpha_p, p \in \{1, 2, ..., n\}$, is to be predicted, and that the domain of α_p is defined as the following set of linguistic terms $\mathcal{L}_p = \{\mathcal{L}_{p1}, \mathcal{L}_{p2}, ..., \mathcal{L}_{ps_p}\}$, with their underlying fuzzy sets being $L_{pq}, q \in \{1, 2, ..., s_p\}$.

The value of α_p should, of course, be assigned according to its underlying domain. To predict this value, FAPACS searches the association rules with $\mathcal{L}_{pq}$ as consequents. If a certain attribute is characterised by a linguistic term in the antecedent of a rule which implies $\mathcal{L}_{pq}$, it can be considered as providing evidence (whether positive or negative), for the value of α_p being assigned to $\mathcal{L}_{pq}$.

This is repeated and the total evidence measure is calculated for each potential value of α_p. The linguistic term with the highest total evidence is then assigned to α_p. More formally, the method is presented in algorithm 2.

Algorithm 2 FAPACS Inference

for all $\alpha_j, j = 1, 2, ..., n,$ **do**
  for all $\mathcal{L}_{jk} \Rightarrow \mathcal{L}_{pq}$ **do**
    for all $\mathcal{L}_{jk} \in \mathcal{L}_j$ **do**
      $w_q += w_{\mathcal{L}_{pq}\mathcal{L}_{jk}} . \mu_{L_{jk}}(\alpha_j)$
    end for
  end for
end for
$\alpha_p \Leftarrow \mathcal{L}_{p(max(w_q))}$

III. Financial Time Series

Financial time series are difficult to analyse. This is because they are typically non-stationary and thus require open or modifiable definitions of memberships and universes of discourse. However, there exists a large body of literature concerned with the conscious and deliberate study of market price history, with a view to predicting the future price change and hence enhance trading profitability. *Technical Analysis* involves in essence a collection of indicators that indicate to a trader a change in the market trend. The method adopted herein for preprocessing the data borrows heavily from this area as the raw data is preprocessed into technical indicators to be fed into the rule induction algorithm. The indicators used in the experiments are outlined below with each using a set of predefined descriptive fuzzy sets.

Price Change: The first, and obviously useful, attribute considered for financial forecasting is the percentage price change over the day.

Comparative Moving Average: This attribute reflects differences between shorter term moving averages and longer term ones.

Volatility: This attribute gives a measure of the intensity of market activity; high volatility is typically an indication of an impending major move in the market.

Volume Moving Average: This is important because when the current volume rises above a long term volume moving average, it is potentially a signal that there is increased market activity that will be affecting the price.

Price Channel Breakout: This is the simplest attribute, indicating that a long trade is placed when the market closes above a previous high and vice versa.

The Stochastic: This attribute aims to identify trends, based on the principle that, as market price rises, the market closes near the market high for the past few periods and vice versa.

Relative Strength Index (RSI): This attribute helps reveal the case of overbought/oversold, which is very useful for predicting price reversal points.

For all the above indicators, the data fed merely provide their values. The decision concerning what action should be taken is left to the rule-firing process of the FAPACS to infer.

Real inputs are fuzzified. As an example, the fuzzy partitioning for the attribute RSI is given in figure 1.

Concerning the training data however, the classification of each record is done by looking "into the future". If the index price of day $k+1$ was 4% above the price on day k, this was considered a StrongBuy; over 2% was considered a Buy. Similarly, if the index price of day $k+1$ was 4% less than the price on day k, a StrongSell would be considered, less than 2% a Sell and otherwise Hold.

IV. Experimental Results

A large number of experimental simulations were carried out using real historical data, typical results are presented below.

A. Explanation Generation

As mentioned previously, one of the main attributes of this approach lay in its superior *comprehensibility* over previous approaches. An example run by the system which implements the associations induced resulted in the following explanation:

```
Today's "Volume-MovingAve:10:Day" is
     definitely (1) "Higher".
This implies that "Conclusion" is "Buy"
     with certainty 0.567332
```

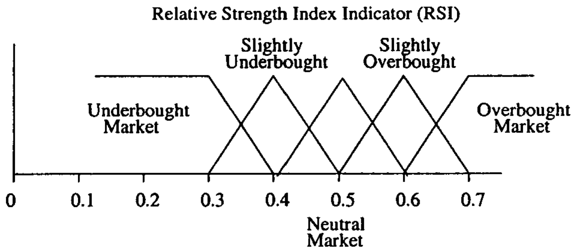


Fig. 1. Fuzzification for Relative Strength Index (RSI)

```
Today's "20:day-Channel-Breakout:UpperBound" is
     slightly (0.425373) "Definite-Breakout".
This implies that "Conclusion" is "Buy"
     with certainty 0.488933

Today's "14:Day-RSI" is
     definitely (1) "Underbought-market".
This implies that "Conclusion" is "StrongBuy"
     with certainty 0.479263

Considering the above, my overall decision for
     today is Buy.
```

A closer examination of these results in fact indicates that the rules inferred by FAPACS are in line with the general ideas of technical analysis. This is very encouraging, given that the system was only lead in the correct direction by allowing the expert to define the fuzzification procedures based on technical analysis but the expert did not provide any of the rules themselves.

B. *Trading on the Equity Financial Markets*

The present application of FAPACS is to train a forecasting system using subsets of data over a certain window size, say n days long. Assuming that today is given by k, the automated trading system is trained from $k-n$ to k in order to predict what will happen on day $k+1$. The window then moves one step and FAPACS is retrained.

The trained system will give one of 5 signals upon classification of a newly presented record: *StrongBuy*, *Buy*, *Hold*, *Sell* and *StrongSell*. A StrongBuy signal would suggest an action to buy the stock with half the cash currently held, while Buy would cause one quarter of cash held to be used. Similarly, StrongSell and Sell would cause an action to liquidate half and a quarter respectively of one's portfolio. Using this approach, the experiment was run during various market eras.

The era between 1973-1988 was first considered. This era was rather volatile due to the existence of two crashes within that time period: the oil crisis (1973) and Black Monday (1987). The results obtained from this run are shown in table I.

Window Size	Total Profit	% Change
50	$2,739,653	↑ 173.4%
100	$1,948,397	↑ 94.8%
150	$1,865,373	↑ 86.5%
200	$1,869,520	↑ 86.9%
250	$1,680,440	↑ 68.0%
300	$1,553,327	↑ 55.3%
Tracker (Buy/Hold)	$2,951,363	↑ 295.1%

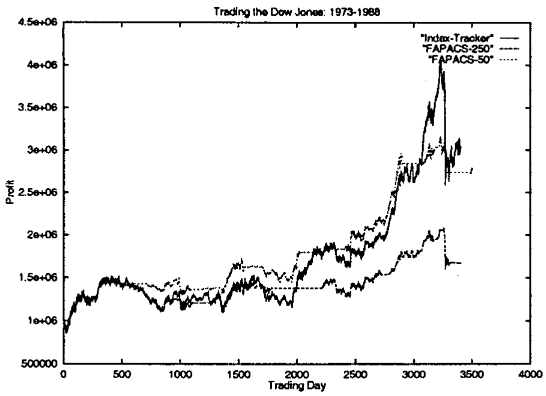
TABLE I

TRADING THE DOW JONES INDUSTRIAL AVERAGE: 7 NOVEMBER 1974 - 12 FEBRUARY 1988

As is evident from the results, the lower the moving window size, the better the results. Figure 2a demonstrates the system's performance during this time period with a window size of 50 and one of 250. It is clear that the use of the 50 day window offers quite a successful run in that throughout the time period, it leads the index tracker. Missing the last market run however caused it to fall under the index tracker.

At first, it may feel counter intuitive that shorter window sizes performed better. One would assume that longer window sizes should be desirable as it makes it possible to differentiate between noise and true system response. Thus, the accuracy of the system should improve as the training window is increased. However, this is only true if the current state of the system remains constant.

During state changes, or in the case of volatile markets as we saw above, long training windows will result in slow system response. Since much of the opportunity for outperforming the market occurs early in the state change, the overall effectiveness of a system with a long training window will be low. The results obtained showed that smaller window sizes are more desirable during turbulent streaks in the market as the association induction algorithm is allowed to react quickly to market changes.



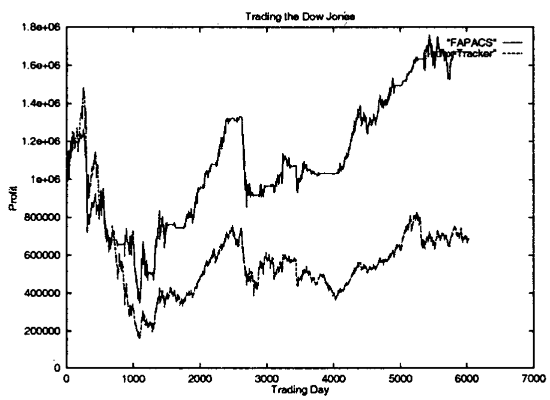


Fig. 2. FAPACS based strategies vs. an Index Tracker Buy/Hold strategy: (a) 1973-1988 (top) and (b) 1928-1949 (bottom)

If we then consider figure 2b which covers data from 1928-1949, we notice that this time, the FAPACS derived rules performed significantly better than the market average. The results obtained for this era are shown in table II. Although in all cases, FAPACS outperformed the market, the best performer was indeed due to the use of, again, the 50 training day window. This reflects an important feature of the present work in that it may suit tasks that require real-time prediction because smaller windows involve less computation, while producing better results.

Attention is then moved to consider an era known for its fairly stable bull market. The graph is shown in figure 3. In this case, the performance of the system unfortunately did not do as well as was hoped. However, considering the graph, it can be seen that during the market dips, FAPACS managed to have caused actions to sell the holdings in time not to be affected by market downturns. Nevertheless, it was unable to give advice to buy quickly enough to make full use of market upturns. The data for this run is shown in table

Window Size	Total Profit	% Change
50	$1,820,402	↑ 82%
100	$782,228	↓ 22%
150	$708,070	↓ 29%
200	$1,508,099	↑ 51%
250	$1,670,836	↑ 67%
300	$1,676,842	↑ 68%
Tracker (Buy/Hold)	$702,193	↓ 30%

TABLE II

Trading the Dow Jones Industrial Average: 25 October 1928 - 17 January 1949

III. As is expected, we notice this time that (except for 50) the performance improved with increasing window sizes for the reasons outlined earlier.

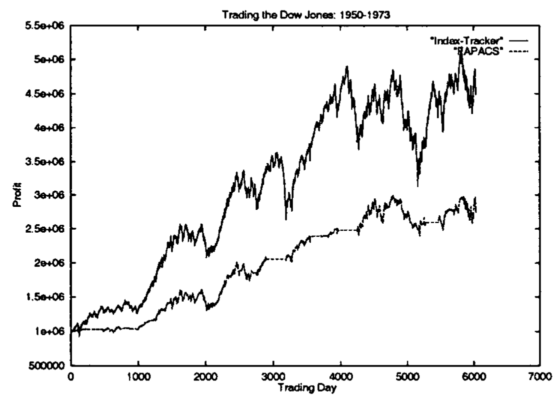


Fig. 3. Trading the Dow Jones Industrial Average: 1950-1973

Window Size	Total Profit	% Change
50	$1,734,906	↑ 73.4%
100	$2,336,453	↑ 133.6%
150	$2,537,919	↑ 153.8%
200	$2,605,027	↑ 160.5%
250	$2,660,607	↑ 166%
300	$2,325,245	↑ 132.5%
Tracker (Buy/Hold)	$4,353,453	↑ 335.3%

TABLE III

Trading the Dow Jones Industrial Average: 8 February 1950 - 30 August 1973

V. Conclusion

This paper has successfully demonstrated the potential for the use of fuzzy association induction in financial forecasting. The approach adopted, using the fixed size training windows sliding through time showed some promise. Continued and automatic adaptation to changes in market conditions is as a result implicitly taken into account by the fact that as time progresses, the newest data is added to the training set while the oldest is discarded. This thus marks a departure from the traditional static Expert System approach and yet managing to solve one of the biggest problems of typical soft computing approaches in financial forecasting (e.g. using a neural network): comprehensibility.

The preliminary results developed here open the door for the analysis and consideration of a whole host of fuzzy rule induction based algorithms against financial forecasting benchmarks. Other extensions may involve using different pre-processing techniques. For example, it would be interesting to use clustering methods on the data without human intervention in order to compare the performance of automatically generated fuzzy sets against that of descriptive sets. Of course, to derive the most out of the potential for explanation generation, descriptive fuzzy sets are desirable.

Acknowledgements

This work was partly supported by a UK-EPSRC grant. The authors are grateful to Ilias Boussios, Alexios Chouchoulas, Bavo Christiaens, Chris Jones, Jeroen Keppens, Javier Gomez Marin-Blazquez and Doris Young for useful discussions.

References

[1] M Adya and F Collopy. "How Effective are Neural Networks at Forecasting and Prediction? A Review and Evaluation", *Journal of Forecasting*, 17(5/6):481–495, 1998.

[2] R Agrawal, T Imierlinkski and A Swami, "Mining Association Rules Between Sets of Items in Large Databases", *Proceedings of 1993 ACM SIGMOD International Conference on Management of Data*, 1993.

[3] R Andrews, J Diedrich and A Tickle, "A Survey and Critique of Techniques for Extracting Rules from Trained Neural Networks", Technical Report, NeuroComputing Research Centre, 1995.

[4] W Au, K Chan. "An Effective Algorithm for Discovering Fuzzy Rules in Relational Databases", *Proceedings of the 7th IEEE International Conference on Fuzzy Systems*, pp.1314–1319, 1998.

[5] K Chan and A Wong. "APACS: A System for Automatic Analysis and Classification of Conceptual Patterns", *Computational Intelligence*, 6:119–131, 1990.

[6] B Flower, T Cripps, M Jabri and A White. "An Artificial Neural Network Based Trade Forecasting System for Capital Markets", *Proceedings of the 4th International Conference on Neural Networks in the Capital Markets*, 1995.

[7] R Gencay and R Garcia. "Pricing and Hedging Derivative Securities with Neural Networks and a Homogeneity Hint", *Proceedings of the 5th International Conference on Neural Networks in the Capital Markets*, 1997.

[8] J Gomez-Marin-Blazquez, Q Shen and A Gomez-Skarmeta. "From Approximative to Descriptive Models", *Proceedings of the 9th IEEE International Conference on Fuzzy Systems*, 2000.

[9] M Qi and G Maddala. "Option Pricing Using ANNs: The Case of S&P 500 Index Call Options", *Proceedings of the 4th International Conference on Neural Networks in the Capital Markets*, 1995.

[10] N Roehl and C Pedreira. "Online Learning for Multi-Layered Neural Network: Applications to Time Series Prediction", *Proceedings of the 4th International Conference on Neural Networks in the Capital Markets*, 1995.

[11] R Srikant and R Agrawal. "Mining Generalised Association Rules", *Proceedings of the 21st VLDB Conference*, 1995.

[12] R Srikant and R Agrawal. "Mining Quantitative Association Rules in Large Relational Tables", *Proceedings of 1996 ACM SIGMOD International Conference in Management of Data*, 1–12, 1996.

[13] H Zimmermann and R Neuneier. "Active Portfolio Management with Neural Networks", *Proceedings of the 6th International Conference on Computational Finance*, 1999.