ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 433-438

High Frequency Forecasting with Associative Memories

Andy Pasley, Jim Austin

Advanced Computer Architecture Group, University of York,
Heslington, York, YO10 5DD, UK
email: arp@cs.york.ac.uk, austin@cs.york.ac.uk

Abstract. The availability of high frequency data sets in finance has allowed the use of very data intensive techniques using large data sets in forecasting. With many algorithms, speed becomes an issue on these larger data sets. An algorithm requiring fast k-NN type search has been implemented using AURA, a binary neural network based upon Correlation Matrix Memories. This work has also constructed distribution forecasts, the volume of data allowing this to be done in a nonparametric manner, making no assumptions of the distribution of the generating process. Currently these forecasts are collapsed to points for evaluation but future evaluation of the distributions and their utility is reported here.

1 Introduction

Many techniques for forecasting nonlinear time series exist. Traditionally in finance forecasters have looked at daily or monthly prices. Recently the availability of large high frequency data sets has encouraged much more research into intra-day and even tick price forecasting. It has also made possible the use of very data-intensive forecasting techniques and the adjustment of traditional techniques.

The Farmer-Sidorowich forecasting algorithm [2] is well understood and many forecasting methods implement a variant of it. Delay coordinates are used to construct representations of the current and all previous states. It is assumed a functional relationship between the current state and the future state exists. The aim is to construct a predictor approximating this function. A forecast of the next price is constructed from one or more 'local' states considered similar by some distance metric to the current state. The evolutions of the local states are used to forecast the evolution for the current state, often by fitting a linear polynomial. Delay coordinates require a 'window' size, the number of most recent historical values that will be used to construct a state. There are no clear rules on how this should be determined, but several researchers have

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 433-438

produced guidelines. Other issues the technique introduces are which distance metric to use, how to implement it and how many neighbour states should be used to construct a forecast?

2 AURA for Farmer-Sidorowich forecasting

The Farmer-Sidorowich algorithm relies on the implementation of a k-NN(or similar) search to decide on the local states. Also, with large data sets of high frequency data it is important that this search can be done quickly, producing forecasts in time for them to actually be used. This work uses the Advanced Uncertainty Reasoning Architecture (AURA) developed at the University of York. A full explanation of AURA is not included here but details can be found in [3]. AURA is an implementation of associative memories using Correlation Matrix Memories(CMMs). An investigation into the use of AURA for k-NN type problems has already been carried out [6] with promising results. The binary nature of AURA makes it fast and efficient and easy to implement into hardware for further improved performance [5].

AURA maps binary input vectors to binary output vectors which can be treated as classifiers or as the centre of a cluster. It allows partial matching from similar input vectors to the same output vector through the use of threshold logic. Adjusting a threshold allows easy control over the cluster size. One problem specified previously is that of how many states should be used to construct a forecast? It is easy using AURA to select states within a certain distance metric, allowing this value to vary depending upon the proximity of the neighbours.

All prices in a given data set must be converted to a binary code, these codes then being used to construct the state(input) vectors. Details are not presented here but a particular quantisation process was implemented that makes no assumptions of the distribution of the input series. A set of boundaries are defined and each value falls into exactly one 'bin' defined by these boundaries. The bins are required to be of variable width (see [6]), widths defined after examination of the training set. The number of bins that the input is quantised into is fixed for a particular model, but varies as a parameter of the system.

The Farmer-Sidorowich algorithm was modified so as to be able to construct frequency histograms. A CMM is used to map input states to features, combining many similar states to one feature. Features overlap and an input state could map to several features, which can be encouraged further by introducing a lower threshold during forecasting than training. During training the evolution of every state is used to construct a frequency histogram for each feature. This histogram can be normalised and treated as a discrete probability distribution. The forecasting process is visualised in figure 1. Stage A in the diagram is the construction of the current state. The neighbouring features to this state are found in stage B. In stage C the distributions for each neighbour are recovered and they are combined by a weighted summation in stage D to produce the forecast. They can be evaluated and used as distribution forecasts

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 433-438

or collapsed to point forecasts. This conversion to a point forecast can be very simple, taking the mean or the highest point of the distribution.

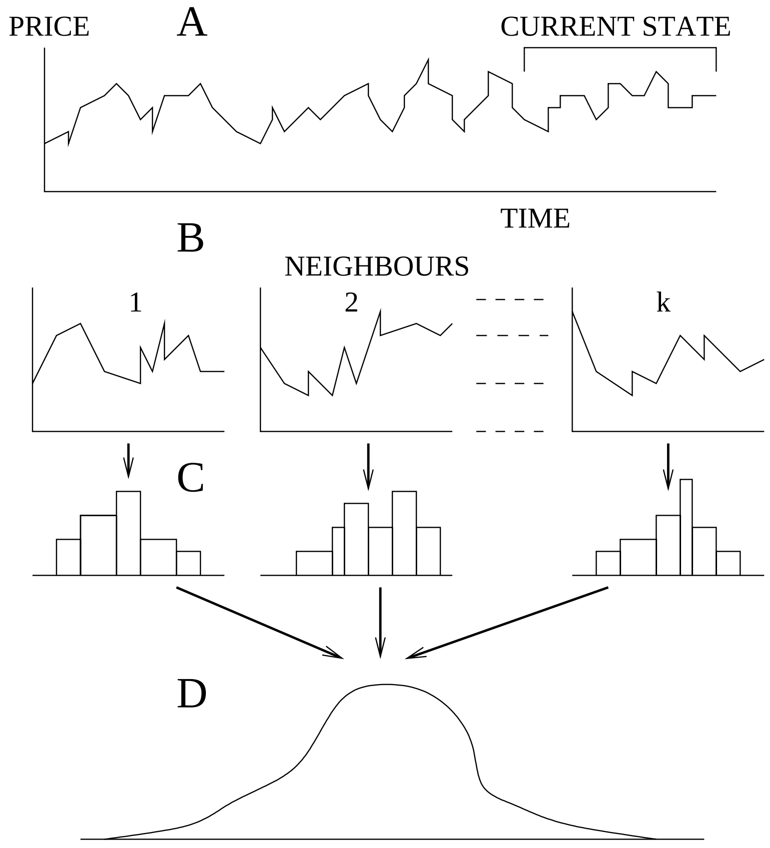

Figure 1: How forecasts are constructed from neighbouring features

This leaves several parameters of the forecasting system which must be decided.

- Window Size
- Number of bins
- Training threshold
- Recall threshold

As discussed before, techniques exist for determining the window parameter, but they are no more than guidelines. The other parameters are particular to CMMs and haven't been investigated elsewhere. Despite the large data sets being investigated, a full search of the parameter space was possible.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 433-438

3 Simulations

The algorithm was tested on a data set of exchange rates between Japanese Yen and US Dollars supplied by Olsen and Associates[1]. From this data, sets of size 25000, 50000 and 100000 were used. They are high frequency sets, with the set of 100000 prices covering only from 1st October 1992 to 8th December 1992.

The first 75% of each data set was taken for training and the rest held back as an out of sample test set. For each data set, many experiments were run, covering all the possible values for the key parameters of window size, number of bins and threshold percentage. Error measures computed were Theil's U-statistic, Normalised Mean Square Error (NMSE) and Directional accuracy. The results for NMSE were exceptionally low, and unfortunately the reason for this is clear and not necessarily the result of good forecasts. NMSE is a measure of how the forecasts compare to the average value of the series, but on large data sets the series can vary from the average for long periods of time, making low NMSE results easy to obtain. Evaluation has therefore concentrated on Theil's U-statistic and Directional Accuracy, and the results reported are for those error measures. Using a simple technique, the forecasts were converted to BUY/HOLD/SELL signals. Based upon these signals a simple trading strategy could be implemented and its return calculated. Two results are reported for the simulation, one where no transaction costs are included and one where trading conditions as in a demonstration on the Olsen website are taken into account.

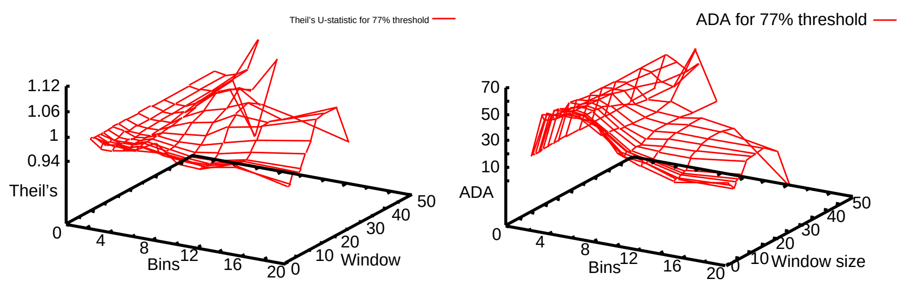

Figure 2: Results using Theil's U-statistic and ADA

Figure 2 shows the results for ADA and Theil's U-statistic over a range of values for the window size and number of bins. Where no result is given for a set of parameters, this is because no forecasts were made. This is due to no state in the test set matching any state encountered during training. This is a symptom of insufficient data, and shows how there is reason to investigate even larger data sets. The Theils U-statistic shows how varying the model makes little impact to the error, and for no model is great forecasting power

[1] Olsen and Associates, Research Institute for Applied Economics, Zurich, Switzerland

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 433-438

achieved. The error is consistently below one however, demonstrating modest forecasting power of a time series considered very efficient. The best results returned are a Theil's of .956 and an ADA of 56.5%, showing promise. There are clear areas where the error is lower and these regions become much more noticeable and interesting when the Average Directional Accuracy (ADA) is investigated. Here results over 50% are achieved only in certain regions of the parameter space, particularly where a low number of bins are used. With Theils below one and ADA over 50%, it has been claimed that forecasting power has been achieved. The simulations in figure 3 show that this can be transferred into theoretical profit, but this disappears when costs are taken into account. Results are reported as the resultant value of trading with an initial investment of 1000. The current method of converting the forecasts to trades is very simple and produces a large volume of trading. Improvements in this regard could yield much improved results.

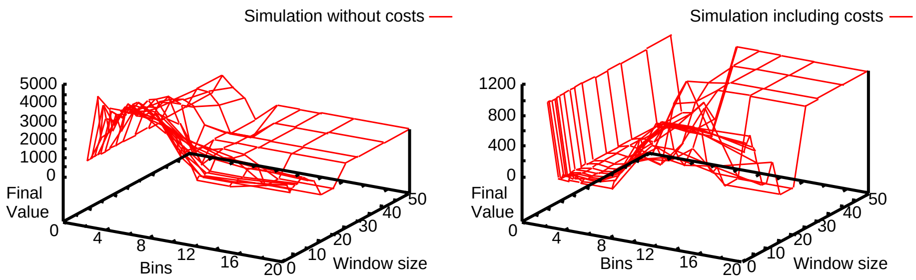

Figure 3: Simulation results with and without transaction costs

It has been stated that the point forecasts being evaluated are generated from probability distributions. It is now fairly well accepted that financial returns do not follow a normal distribution and this has encouraged the measurement of skewness and other properties of financial returns. The market is considered a chaotic system where the generating process varies over time. The actual distribution at any given time is not observed and cannot be known, making evaluating the distributions a difficult task. The only techniques known to the author of evaluating distribution forecasts using the observed values are [1, 4].

An alternative study of the distributions could give further evidence that Farmer-Sidorowich type algorithms are suitable for financial forecasting. The input space is clustered into 'features' by a CMM and these clusters used to build the distributions. If fresh distributions are constructed from the test set, then for each feature the distribution should be similar to that constructed in training. Preliminary search in this area is promising, but no particular similarity metric has been decided upon as ideal for this task.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 433-438

4 Conclusion

Financial time series are considered very efficient and therefore difficult to forecast. Our work provides further evidence that this is so, with the best forecasts showing little improvement on a 'no change' model. The errors vary little as key parameters such as the window size are changed.

There are however reasons to be optimistic about the obtained results. Firstly, the ADA error measure varied more over the different parameters and the best achieved was over 50%. In conjunction with a Theils below 1, this shows forecasting power. The point forecasts are constructed from full probability distribution forecasts that offer additional information which can be incorporated into a trading strategy.

Simulations which assume no barriers to market entry and no transaction costs have proved the excess profit that can be returned. Taking into account minimal costs wipes out this profit. This is due to the trading intense strategy currently constructed by the algorithm. Currently forecasts are next-step forecasts. Methods to extend the forecasts further ahead in time need to be investigated. It is expected that along with an improved trading strategy better simulation results can be achieved.

References

[1] Francis X. Diebold, Todd A. Gunther, and Anthony S. Tay. Evaluating density forecasts with applications to financial risk management. In *International Economic Review*, volume 39, pages 863–883, 1998.

[2] J. D. Farmer and J. J. Sidorowich. Predicting chaotic time series. In *Physical Review Letters*, volume 59, 1987.

[3] D. Kustrin. *Forecasting Financial Time Series with Correlation Matrix Memories for Tactical Asset Allocation*. PhD thesis, University of York, Heslington, York, YO10 5DD, England, July 1998.

[4] A. S. Weigend and S. Shi. Predicting daily probability distributions of s&p500 returns. Unpublished, available from author's web pages, 1999.

[5] P. Zhou and J. Austin. A binary correlation matrix memory k-nn classifier with hardware implementation. In *British Machine Vision Conference*, 1998.

[6] P. Zhou, J. Austin, and J. Kennedy. A high performance k-nn classifier using a binary correlation matrix memory. In *Neural Information Processing Systems Natural and Synthetic*, volume 11, 1998.

The research reported here has been supported by an EPSRC studentship.