Time Series Prediction Using Decomposition onto a System of Random Sequence Basis Functions and a Temperature-Dependent SOFTMAX Combiner

Neep Hazarika

Abstract—**We describe a novel method of combining multiple random sequences using a "temperature dependent" combiner. Each random sequence can be regarded as a prediction model. The goal is to develop an efficient combining methodology that takes into account the performance of each forecasting sequence in an "optimal" way when calculating combiner weightings. The method essentially involves the decomposition of the target time series onto a random sequence basis. A FORTRAN computer program CRANSEQ (*C*ombination of *RAN*dom *SEQ*uences) has been developed to implement the algorithm.**

I. Introduction

There is a large body of literature describing how best to combine models of various types of processes [1]-[7]. Two such groups of methods are ensemble methods [8, 9] and mixture of experts methods [10, 11]. Combining forecasts effectively is a non-trivial process in the case where high levels of noise exist, as can occur in many representative time series structures for industry. The current trend is towards combining predictive models, rather than employing large monolithic predictors. The advantage of the former is that such a methodology would be more efficient in terms of training time [12]. Further, it is also possible to achieve a lower generalization error from the combiner [13], as well as to prevent overfitting. The models considered in this paper are random sequences that are used as prediction models for the time series provided as data for the NN3 2006/2007 Forecasting Competition. .How can an effective solution be achieved for such a task, an important component in the overall prediction process for time series with important industrial implications? We consider that topic in this paper.

A novel feature of the methodology developed in this paper is the use of random sequences as prediction models. These random sequences can be considered as basis functions for the decomposition of the target series. The problem one is faced with is how to combine these sequences in order to effectively leverage their intrinsic randomness to obtain accurate forecasts of the target series. We should note here that the chosen random input sequences chosen for the various "models" are all different, so that one can assume that the different models attempt to extract different information from the target series. So again the essential question remains: How can an effective combination of random sequences be achieved?

A simple approach is to take a linear combination of the sequences which are regarded as prediction models of the target time series. Thus, we take a set of M weights $w_1, w_2, \ldots, w_M$ (where M is the number of random models), and form the weighted sum

$$w_1 x_1{}^t + w_2 x_2{}^t + \cdots + w_M x_M{}^t \ ,$$

where $x_i{}^t$ is the prediction of the i^{th} random sequence model at time t. We are then faced with the question of how one should choose the numerical values of the weights? It is the answer to that question that has been a central focus in prediction over the few previous years, as the references already cited indicate.

Originally, the combining process involved simple averaging over predictions from all random sequence models for a given target series. However, this method did not penalize bad models sufficiently. As an alternative, a linear combining method was developed which minimized the prediction error over a specified test set. This, too, did not sufficiently penalize underperforming models, as determined by careful and extended testing. The method described in this paper is a modification of the technique described in [14]. It incorporates both averaging and "winner takes all" methods. This is achieved by linearly combining all predictions such that, given the data, the "best" prediction can be determined when measured in an appropriate metric. A novel aspect is that the current combiner weights are obtained from a *SOFTMAX* distribution [10, 15]. The weights can therefore be interpreted as probabilities, as they lie in [0, 1] and sum to unity.

In the next section we describe the methodology employed in the development of the combiner and in the following one, the corresponding algorithm. In section IV, we will present the results for the NN3 competition. The paper concludes with a discussion of the results, and scope for further work.

II. Method

We want to derive a linear combination of all the random sequence models such that the "best" possible prediction can

Manuscript received January 20, 2007

Neep Hazarika is with DeepSea Engineering & Management Ltd., 39A East Street, Epsom, Surrey KT17 1BL, UK (phone: +44-(0)780-370-8444; fax: +44-(0)137-274-1497; e-mail: neep_hazarika@hotmail.com).

be determined given the data, when measured in an appropriate metric. Such a metric can be provided by the *Symmetric Mean Absolute Percentage Error (SMAPE*) [16] of the individual models. The *SMAPE* is defined as follows:

Given a time series $x_1, x_2, \ldots\ldots, x_N$, the *SMAPE* for the random prediction sequence $\hat{x}_1, \hat{x}_2, \ldots\ldots, \hat{x}_N$ is given by

$$SMAPE = \frac{100}{N} * \sum_{i=1}^{N} \frac{|x_i - \hat{x}_i|}{(x_i + \hat{x}_i)/2},$$

where x_i is the true value of the i^{th} point of the time series of length N, and $\hat{x}_i$ is the predicted value.

We also want to normalize the weights w_j given to each prediction model j such that they sum to unity. The weights can thus be interpreted as *probabilities, i.e*., they must be in the range [0,1] and they must sum to one. Such a normalization can be achieved using the so-called Potts or SOFTMAX activation function [15] which takes the form

$$w_j = w(SMAPE_1, SMAPE_2 \ldots, SMAPE_M, k, T)$$

$$= \frac{\exp(\alpha_j)}{\sum_{i=1}^{M} \exp(\alpha_i)}$$

where $\alpha_j = -\frac{(SMAPE_j)^k}{T}$, $SMAPE_j$ is the SMAPE of the j^{th} prediction model sequence, and M is the number of models. The term “SOFTMAX” is used because this activation function represents a smoothed version of the *winner-takes-all* model, in which the prediction sequence with the largest “probability” is given a weight of +1 while all other weights have output zero. We incorporate two additional parameters k and T in this model. These parameters are determined by optimization in order to minimize the SMAPE of the combined prediction over a specified training set. When T takes on large values, the average of all predictions is the best that the combiner can obtain. Such a case arises, for example, when there are a large number of equally poor models, so that the combiner cannot distinguish between them. On the other hand, if some of the models outperform most other models, these are then singled out since T tends to have a low value (as expected from the winner-takes-all strategy). Also, the winner-takes-all model is recovered in the limit $k \longrightarrow \infty$ or $T \longrightarrow 0$. For $T \longrightarrow \infty$, we regain the average, with no *a-priori* knowledge. The parameters k and T can be determined by using optimization techniques, as described in the next section.

III. THE COMBINER ALGORITHM

We now describe the prediction methodology, and the combiner cost function to be optimized. The algorithm is as follows:

1. Preprocessing of the target series is performed by first robustly detrending the data. This is accomplished by fitting a minimum absolute deviation line to the data and calculating the detrended series as

   $$x'_t = x_t - (a + bt) + \overline{x},$$

   where x_t is the value of the time series at time t, a and b are the intercept and slope respectively of the trend line, and $\overline{x}$ is a value chosen to be large enough so that the detrended value x'_t is always positive. This is necessary to prevent computational errors while calculating the exponential in the combiner cost function. In most cases, $\overline{x}$ can be chosen to be the mean, *i.e*.,

   $$\overline{x} = \frac{1}{N} \sum_{t=1}^{N} x_t$$

   where N is the total number of points in the target series. If a visual inspection of the detrended data displays a large variance, the data may be further transformed using the logarithmic function. Further, the data is subdivided into a *training* set of length N_{train} and a *test* set of length N_{test}, such that $N_{train} + N_{test} = N.$ In this paper, N_{test} was set equal to $N/3$.
2. Let $x_{\max}$ and $x_{\min}$ represent the maximum and minimum values respectively of the detrended (and, if necessary, transformed) series x'_t ($t = 1,2,\ldots,N$).
3. Let the number of random prediction models be M, and let p be the number of points to be forecast. For the purposes of this paper, a value of M =100 models was sufficient.
4. We now generate M random sequences drawn from an uniform distribution, each of length ($N + p$), using a random number generator, and transforming the resulting values $\hat{x}_{it}$ $(i = 1,2,\ldots M; \quad t = 1,2,\ldots,N + p)$ such that they all lie between $x_{\min}$ and $x_{\max}$. Each of these sequences can be regarded as a prediction model for the target series.

5. Form a sequence of M model SMAPEs for each random model ($SMAPE_i, i = 1, M$) over the first N_{train} points:

$$SMAPE_i = \frac{100}{N_{train}} * \sum_{t=1}^{N_{train}} \frac{|x'_t - \hat{x}_{it}|}{(x'_t + \hat{x}_{it})/2}.$$

6. Set up a counter n for the number of prediction models used at each step, up to a maximum value of M. Initially, set $n = 1$.

7. Compute the sequence of N_{train} combined predictions for the n models on the training set:

$$\hat{y}^{(n)}{}_t = \sum_{j=1}^{n} w^{(n)}{}_j \hat{x}_{jt},\ t = 1,2,\ldots N_{train},$$

where the weights $w^{(n)}{}_j$, ($j = 1,2,\ldots,n$) at level n are given by

$$w^{(n)}{}_j = \frac{\exp(\alpha_j)}{\sum_{j=1}^{n} \exp(\alpha_j)} \qquad (1)$$

and $\alpha_j = -\frac{(SMAPE_j)^{k^{(n)}}}{T^{(n)}}$. Note that the weights $w^{(n)}{}_j$ are functions of the parameters $k^{(n)}$ and $T.^{(n)}$ For the trivial case $n = 1$, $w^{(1)}{}_1 = 1.0$. Further, the weights $w^{(n)}{}_j$ as well as the parameters $k^{(n)}$ and $T.^{(n)}$ will have different values at each level n as the number of random prediction models is increased.

8. Compute the combined SMAPE on the training set at level n as follows:

$$SMAPE^{(n)}{}_{train} = \frac{100}{N_{train}} * \sum_{t=1}^{N_{train}} \frac{|x'_t - \hat{y}^{(n)}{}_t|}{(x'_t + \hat{y}^{(n)}{}_t)/2}$$

Note that $SMAPE^{(n)}{}_{train}$ is also a function of the unknown parameters $k^{(n)}$ and $T.^{(n)}$

9. Determine the value of $k^{(n)}$ and $T^{(n)}$ at the global minimum of $SMAPE^{(n)}{}_{train}$. The global minimization is performed in this case via a *simulated annealing* technique [17].

10. Compute the weights $w^{(n)}{}_j$, ($j = 1,2,\ldots,n$) in equation (1) using the values of $k^{(n)}$ and $T^{(n)}$ obtained at the global minimum.

11. Compute the combined predictions for the out-of-sample test set as follows:

$$\hat{y}^{(n)}{}_t = \sum_{j=1}^{n} w^{(n)}{}_j \hat{x}_{jt},\ t = N_{train} + 1,\ldots,N.$$

Also calculate the combined SMAPE on the test set:
First, obtain the combined predictions for the original test set. This can be obtained by adding in the original trend line for the target series as follows:

$$y^{(n)}{}_t = \hat{y}^{(n)}{}_t + (a + bt) - \bar{x}. \qquad (2)$$

If the detrended series was transformed via the logarithmic function, then

$$y^{(n)}{}_t = \exp(\hat{y}^{(n)}{}_t) + (a + bt) - \bar{x}, \qquad (3)$$

and

$$SMAPE^{(n)}{}_{test} = \frac{100}{N_{test}} \sum_{t=N_{train}+1}^{N} \frac{|x_t - y^{(n)}{}_t|}{(x_t + y^{(n)}{}_t)/2}$$

where x_t is the value at time t of the original target series.

12. At each level, record the minimum SMAPE achieved on the test set, as well as the optimal number of models $nopt$, the weights $wopt^{(nopt)}{}_j$ and random sequence values $\hat{x}opt_{jt}$ $(j = 1,\ldots,n;\quad t = 1, N + p)$ that produced this SMAPE.

13. Increase the value of n by one and repeat steps 7 to 12 until $n = M$. In this study, a value of $M = 100$ was found to be sufficient.

14. In order to simulate a global search as closely as possible, steps 4 to 13 were repeated several times and the relevant quantities described in step 12 were stored in order to determine the optimal combination of random prediction models that minimized the SMAPE on the test set. We used 200 simulations in this research.

15. Finally, the required p predicted values of the detrended target series are computed as

$$\hat{y}^{(nopt)}{}_t = \sum_{j=1}^{nopt} wopt^{(nopt)}{}_j \hat{x}opt_{jt},$$

$$t = N + 1, N + 2, \ldots N + p.$$

The trend line is then added in to obtain the predictions for the original target series in a manner analogous to the formulae in equations (2) and (3).

IV. RESULTS

The general effectiveness of this method is determined as a function of the SMAPE on the training set. The effective generalization capability of the combiner remains to be verified after a detailed comparison of the results obtained from the NN3 2006/2007 competition.

Various modifications of the combiner have been studied, *e.g.*, fixing the values of k or T, or only allowing them to lie within a specified restricted range. None of these modifications appear to be beneficial, as measured by the generalization error of the resulting combiner on the training set.

V. CONCLUSIONS

The decomposition of the target series onto a system of random sequence basis functions, used in tandem with the temperature-dependent SOFTMAX combiner developed in this paper leads to a highly effective prediction methodology. This technique allocates the highest weights to those sequences that best model the target series. However, a more efficient search algorithm for the optimal random sequences could improve the performance of the method. Also, the global search could be greatly enhanced with increased computational storage capacity, as the search could then be accomplished with just one iteration sweep over a much larger ensemble of realizations. In reality, only a few random sequences will be chosen as basis functions for a given target series, with the majority being allocated weights that are zero or negligible.

Further work needs to be undertaken in order to relate the current methodology to others, such as those based on *automatic relevance detection* which employs Bayes' theorem. Some progress has already been achieved in constructing a Bayesian optimal linear combiner, using a linear *relevance vector machine (RVM)* predictor [18, 19]. We will report elsewhere the relationship between these approaches.

REFERENCES

[1] D. W. Bunn, “A Bayesian Approach to the Linear Combination of Forecasts,” *Opinions on Research Quarterly*,, pp. 325-329, vol. 26, 1975

[2] D. W. Bunn and E. Kappos, “Synthesis of Selection of Forecasting Models,” *European Journal of Operational Research*, . pp. 173-180, vol. 9, 1982

[3] D. W. Bunn, “Statistical Efficiency in the Linear Combination of Forecasts,” *International Journal of Forecasting*, pp. 151-163, vol. 1, 1985

[4] R. A. Jacobs, “Methods of Combining Experts' Probability Assessments,” *Neural Computation*, pp. 867-888, vol. 7, 1995

[5] P. G. Harrald and M. Kamstra, “Evolving Neural Networks to Combine Financial Forecasts,” *IEEE Transactions on Evolutionary Computation*, pp. 40-51, vol. 1, 1997

[6] L. M. de Menezes, D. W. Bunn and J. W. Taylor, “Review of Guidelines for the Use of Combined Forecasts,” *London Business School Preprint*, April 1998

[7] A. Sharkey, Combining Models, London: Springer Verlag, 1998

[8] Y. Freund and R. E. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” *Proceedings of the Second European Conference on Computational Learning Theory*, pp. 23-37, Springer=Verlag, March 1995

[9] L. Breiman, “Bagging Preictors,” *Machine Learning*, pp. 123-140, vol. 24, April 1996

[10] R. A. Jacobs, M. I. Jordan, S. J. Nowlan and G. E. Hinton, “Adaptive mixtures of Local Experts”, *Neural Computation 3 (1)* , pp. 79-87, 1991

[11] M. Jordan and R. Jacobs, “Hierarchical Mixtures of Experts and the EM Algorithm,” *Neural Computation 6 (2)* , pp. 181-214, 1994

[12] B. L. Lu and M. Ito, “Task Decomposition and Module Combination Based on Class Relations: a Modular Neural Network for Pattern Classification,” *Technical Report*, Nio-Mimetic Control Research Center, The Institute of Physical and Chemical Research (RIKEN), 2271-130 Anagahora, Shimosi-dami, Moriyama-ku, Nagoya 463-003, Japan, March 1998

[13] A. Krogh and J. Vedelsby, “Neural Network Ensembles, Cross Validation, and Active Learning,” in G. Tesauro, D. Touretzky and T. Leen (Eds.), *Advances in Neural Information Processing Systems*, vol. 7, MIT Press, pp. 231-238, 1995

[14] N. Hazarika and J. G. Taylor., ``A Temperature-Dependent SOFTMAX Combiner," *Proceedings of IJCNN 2001, the INNS-IEEE International Joint Conference on Neural Networks*, Washington, DC, pp. 1847-1851, 14-19 July 2001

[15] J. S. Bridle, “Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition,” in F. Fogelman Soulié and J. Hérault (Eds.), *Neurocomputing: Algorithms, Architectures and Applications*, New York: Springer-Verlag, pp. 227-236, 1990

[16] S. Makridakis, “Accuracy measures: theoretical and practical concerns,” *International Journal of .Forecasting*, **9**, pp. 527–529. 1993

[17] W. L. Goffe, G. D. Ferrier and J. Roberts, “Global optimization of Statistical Functions with Simulated Annealing,” *Journal of Econometrics*, vol. 60, pp. 65-99, 1994

[18] N. Hazarika and J. G. Taylor, ``Predicting Bonds Using the Relevance Vector Machine," *Proceedings of the Eighth International Conference on Forecasting Financial Markets: Advances for Exchange Rates, Interest Rates and Asset Management*, London, 30 May-1 June, 2001

[19] N. Hazarika and J. G. Taylor, ``A New Bayesian Combiner for Financial Time Series prediction," *Proceedings of the Ninth International Conference on Forecasting Financial Markets: Advances for Exchange Rates, Interest Rates and Asset Management*, London, 29-31 May, 2002