Combining forecasts: a fuzzy approach

Antonio Fiordaliso
Faculté Polytechnique de Mons, MathRO
9, rue de Houdain, 7000 Mons, Belgium

Abstract. In this paper, [1] we investigate the use of first order Takagi-Sugeno fuzzy systems (TS1) [19] to combine a set of individual forecasts. Such systems can be interpreted as local linear approximation models and have been used mainly as such in this study. The inference produced by these models can be seen as a new kind of piecewise linear regression with softened transitions between the pieces. We compare TS1 with traditional linear combining models and we show the advantage of this nonlinear approach as well as the flexibility of our system.

Key-words: combination of forecasts, Takagi-Sugeno fuzzy systems, parametric and structural tuning.

Introduction

The combination of forecasts is a technique of great interest when several alternative forecasts of a time series are available. Many theoretical studies and empirical tests have shown the advantage of aggregating forecasts [5]. The motivation comes from the idea that each forecasting model is based on a specific information set and is thus sensitive to particular characteristics of the series to be modelled.

The annotated bibliography found in [5] reports more than two hundred publications in this area and shows the importance of this topic in the forecasting community. In this study, we will adopt the usual combining forecasts policy that consists in taking a vector of forecasts $\mathbf{X}(t) = (X_1(t), \ldots, X_p(t))'$ at time t (' denotes the transpose) and constructing a combined forecast $C(t)$ as a linear mixture of the individual forecasts $X_i(t)$:

$$C(t) = w_0(t) + w_1(t)X_1(t) + \ldots + w_p(t)X_p(t) \tag{1}$$

where $w_i(t)$ are real numbers computed at time t called combination weights. A purely linear combination scheme will result if the combining weights are constants or do not depend on $X_i(t)$. Because we don't know the a priori type of dependence between the individual forecasts and the true values, the purely linear case may be restrictive in general. This motivates the use of more general nonlinear combining schemes such as the one proposed in this study.

Several existing statistical combining methods are closely related to the problem of approximating a p-dimensional mapping $\mathcal{F} : \Re^p \to \Re$ from a set of N points of $\mathcal{F}$ denoted by ($\mathbf{X}$(t), Y(t)) where t is a temporal index taking values in $\{1, \ldots, N, N+1, \ldots, T\}$ and $Y(t)$ is the value to forecast at t. For example, the linear regression combining model [11] can be seen as a (linear) approximation method. In this paper, we use a particular class of universal approximators, namely first order Takagi-Sugeno systems to compute the combination weights. This approach appears to be natural once the combination problem is thought as function approximation problem. Some universal approximators arousing a great interest are regularization networks including as special cases multidimensional splines and radial basis functions [15], a large class of neural networks with one layer of hidden units [12] and various types of fuzzy systems including Takagi-Sugeno fuzzy systems [4]. Fuzzy systems are based on the concept of associative rules and have the advantage over neural networks of allowing more readability. This advantage may be determinant if some prior knowledge is available, or if a minimum of transparency is required.

[1] This paper has been presented at AMSE-ISIS'97 (International Symposium on Intelligent Systems) and published in the related Proceedings.

1 Takagi-Sugeno fuzzy systems

The k^{th} rule of first order Takagi-Sugeno model (TS1) may have the following form: IF $\mathbf{X}$ is A_k THEN $Z = B_k(\mathbf{X})$ $(k = 1, \dots, r)$, where $\mathbf{X} = (X_1 \dots X_p)'$, A_k is a fuzzy set of $\Re^p$ with membership function m_{A_k} and $B_k(\mathbf{X}) = b_k(0) + b_k(1)X_1 + \dots + b_k(p)X_p$. We slightly modify the classical computation of the system output Z corresponding to input $\mathbf{X}$ by introducing a supplementary real parameter ρ_k controlling the importance of rule k in the inference process. The final output Z is now computed as follows:

$$Z = \frac{\sum_{k=1}^{r} g(\rho_k) m_{A_k}(\mathbf{X}) B_k(\mathbf{X})}{\sum_{k=1}^{r} g(\rho_k) m_{A_k}(\mathbf{X})} \tag{2}$$

where function g defined by $g(\rho_k) = 1/(1+\exp(-\rho_k))$ normalizes the intensities ρ_k in the range [0,1]. A small value for g(ρ_k) means that the corresponding rule can be deleted from the system without altering too much the system output value. This sensitivity parameter will be used in the sequel to detect and remove redundant rules. Note that formula 2 leads to the classical TS1 output computation formula when g is constant.

From now, we will suppose that the processed input X_j represents the forecast of model j (at time t). So, the output Z given in equation 2 can be seen as a composite forecast, since it can be written as in equation 1 with $w_0(t) = 0$ and

$$w_j(t) = \frac{\sum_{k=1}^{r} g(\rho_k) m_{A_k}(\mathbf{X}) b_k(j)}{\sum_{k=1}^{r} g(\rho_k) m_{A_k}(\mathbf{X})} \quad j = 1, \dots, p. \tag{3}$$

TS1 models are the only fuzzy systems allowing to achieve this decomposition. This is due to the special structure of TS1 systems, more precisely to the linearity of the local output models. A second advantage of TS1 systems is that, due to the fact that output Z is differentiable versus any of the input and output parameters, gradient based procedures can be used to automate the adjustment phase.

The antecedents A_i of each rule partition the input space into a number of regions for which a specific behaviour (local output model) is encoded in each rule consequent. Usually, input gaussian type membership functions are used to localize the input regions (also called clusters). A very important feature of such systems is that two local regions can overlap (because the antecedent part of a rule is not either true or false, but can be partially true). This produces a kind of piecewise linear regression with softened transitions between the pieces. Other closely related approximators can be found in statistics, such as Kernel Regression Models [20], or in the connectionist literature, such as Mixtures of Experts [13] or Receptive Field Weighted Regression Models [16]. When the local models are constants, TS1 systems can be related to other neural networks such as Radial Basis functions [15] or General Regression Neural Networks [18].

We use the p-dimensional generalized gaussian-type membership functions for the coding of the input linguistic terms:

$$m_{A_k}(\mathbf{X}) = G(\ \|\mathbf{X} - \boldsymbol{\mu}_k\|^2_{\mathbf{S}_k}\ ) \tag{4}$$

where $\boldsymbol{\mu}_k$ is the center of the gaussian, $G(x) = \exp(-x)$ and $\|\mathbf{X}\|_{\mathbf{S}_k}$ is the weighted norm of $\mathbf{X}$ defined by $\|\mathbf{X}\|^2_{\mathbf{S}_k} = \mathbf{X}'\ \mathbf{S}'_k\ \mathbf{S}_k\ \mathbf{X}$ where $\mathbf{S}_k$ is a square matrix. $\mathbf{S}_k$ operates a linear transformation of the inputs vectors that achieves a scaling preprocessing operation. Adjusting the entries of $\mathbf{S}_k$ means changing the metric on the input space. In the diagonal case, the metric induced by $\mathbf{S}_k$ is orthogonal (euclidian, if $\mathbf{S}_k$ is the identity matrix). If we allow $\mathbf{S}_k$ to be any square matrix, then several structures of input clusters can be captured since the resulting gaussians can rotate around their centers and can be very elongated in some directions related to the largest eigenvectors of $\mathbf{S}_k$.

2 Tuning of Takagi-Sugeno fuzzy systems

Due to space limitation, we give a brief description of the gradient-based algorithm used to adjust the free system parameters along with the decremental method aiming at discovering the adequate number of inference rules to use. More details can be found in [9].

Parametric tuning is achieved by minimizing the usual global quadratic error function $E = \sum_{t=1}^{N}(Y(t) - Z(t))^2$ on the learning set. Since E is a derivable (nonlinear) function of the parameters to adjust, a gradient descent technique can be applied for optimization purpose. We use the technique of Silva and Almeida [17] in which an independent learning rate is used and adapted for every parameter. The gradient descent procedure stops when stabilization is observed, practically, when the absolute value of the relative variation of the error E is below a predefined threshold set to $2\ 10^{-4}$.

The basic idea of the decremental algorithm used for structural tuning is to start with an oversized bank of rules and then, gradually detect and remove redundant rules from the system. Beginning with 10 rules in our simulations ensures overparametrization, since the number of initial free parameters is greater than the number of available patterns for learning. The final structure to reach must be as small as possible in order to avoid the well-known overfitting effect, but rich enough to extract important features of the learning set. We use Schwarz's BIC information criteria during learning to evaluate the performance of the model relatively to its complexity.

The outline of the structural algorithm is as follows. We split the learning set in two parts of equal size; one set (training set) is for learning, the other (validation set) is for cross-validation. The final structure is the one producing the lowest BICV (BIC computed on the validation set). Cross-validation helps to prevent the system to model details of the training set because its evaluation is made on a different set. Each time the gradient descent procedure stabilizes, we find the set of rules which sensitivity parameter $g(w_k)$ are below $1/r$, where r is the number of active rules. These rules are then successively selected by increasing values of $g(w_k)$. If the BICV criteria decreases when a particular rule is suppressed, then this rule is removed from the bank, else, it is replaced in the bank and another rule is considered for pruning. The algorithm stops when no rule is suppressed during one pruning pass. This decremental approach has been compared to the incremental one of Gorrini and al. [10] and gives interesting results on one dimensional mapping approximation problems [8]. More tests on multidimensional mappings are in progress.

3 Experimental results

The initial values of the rules weights ρ_k are set to 0 and the learning rates for w_k are set to a large value (1.0) in order to counterbalance the saturation effect caused by the sigmoïd function g and to achieve a sufficient discrimination between the rules. A K-means clustering algorithm [1] is used to locate the initial gaussians centers. The initial matrixes $\mathbf{S}_l$ are set proportional to the identity matrix and the coefficients of the linear local outputs are 0. The initial learning rates for input (resp. output) parameters are set to 0.005 (resp. 0.05).

The data used in this study are univariate time series. Time series A (T=200 values) [2] is related to daily temperatures at noon on Ben Nevis. Series B (T=312) [14] reports monthly FTA All Share price index. Series C (T=240) [2] is related to mean monthly air temperatures at Nottingham Castle. These time series are preprocessed for stationarity purpose. A first-order differencing is used for time series A, a logarithmic transformation followed by a first-order differencing is appropriate for time series B, while a seasonal differencing of period 12 is required for time series C in order to remove the seasonal component. The values Y(t) to forecast are these computed preprocessed values. All the forecasts used in this study for combining purposes are one-step-ahead out-of-sample forecasts. Each time series has been divided in two parts of equal size; the first one is dedicated to the estimation of the forecasting models parameters, while the second one is used to generate out-of-sample forecasts by applying the fitted models. For each series, the same two forecasting models have been used. The first one is the well-known ARIMA(p,d,q) model estimated according to the Box-Jenkins methodology. The fitted models ARIMA(p,d,q) for series A, B, and C are respectively (1,0,1), (3,0,0) and (12,0,0). The second one is a K-nearest neighbours model (KNN) frequently used to forecast chaotic maps [7]. In this case, there may be an advantage to using a combination technique since the two forecasting models process the past information in a completely different way; the ARIMA approach assumes that a stationary *linear stochastic* process generates the data points in the time series, while the KNN method is based on a *nonlinear deterministic* approach.

Figure 1 shows the ten initial input gaussians placed in the case of time series A along with the final input structure obtained and tuned as explained previously. The number of iterations needed to obtain the final structures and their parameters for series A, B and C are respectively 250, 350 and 750. The number of remaining rules are respectively 2, 2 and 1. Note that when only one rule remains, the output of TS1 is purely linear. In this case, the gradient descent procedure tends to move the linear output model of the remaining rule towards the ordinary least squares (OLS) hyperplane. But remember that due to the cross-validation technique used in our algorithm, the TS1 hyperplane may differ slightly from the one computed by OLS on the whole available data set.

We have compared the aggregated forecasts generated by our combining model (TS1) with those produced by other combining methods. The first alternative combining scheme is the popular simple average of individual forecasts (SAV) often used as benchmark in the combining forecasts literature because of its simplicity. The others methods are variants of the classical linear regression model (REG); REG0 is a regression model with no constant term, REG+ is a regression model with non negative parameters, REG01 is a linear regression without independent term and with coefficients adding to one, REG01+ adds to REG01 the additional constraint that the coefficients must be non negative. The motivation for choosing the preceding regression combining schemes is because no consensus actually exist in the general case concerning the issues of whether the combining weights should be constrained to be non-negative, to sum to one or should include an independent term (see [5] and the many references cited therein).

Table 1 reports the mean square error (MSE), the normalized error (NER), the mean absolute error (MAE) and the Theil's U statistic (U) of the forecasting and combining models for the three time series considered. All these error measures are computed on the data sets devoted to the testing of the combining models (the last quarters of the initial time series). So, the error measures concern out-of-sample forecasts and out-of-sample composite forecasts. The NER criteria is simply the root mean square error divided by the standard deviation of the data. A NER value inferior to 1 indicates that the forecasts are better than the constant mean predictor. Improvements over the random walk forecasts are indicated by values $U < 1$. A complete description of these error measures can be found in [7] and [3].

Table 1: A comparison of the different out-of-sample error measures of the forecasting and combining models for time series A, B and C.

	A				B				C			
	MSE	NER	MAE	U	MSE	NER	MAE	U	MSE	NER	MAE	U
ARIMA	21.00	1.26	3.72	0.97	0.40	1.03	4.49	0.75	6.24	0.76	1.85	0.53
KNN	17.55	1.15	3.44	0.88	0.61	1.27	5.51	0.95	8.02	0.86	2.93	0.78
SAV	15.92	1.10	3.29	0.84	0.49	1.13	4.63	0.82	7.74	0.84	2.15	0.60
REG	15.03	1.07	3.33	0.82	0.37	0.99	4.29	0.72	6.61	0.78	1.98	0.56
REG0	14.07	1.03	3.09	0.71	0.39	1.01	4.49	0.73	6.17	0.75	1.83	0.54
REG+	14.07	1.03	3.09	0.79	0.39	1.01	4.48	0.73	6.17	0.75	1.83	0.53
REG01	15.95	1.10	3.29	0.84	0.42	1.06	4.68	0.77	6.23	0.76	1.84	0.54
REG01+	15.93	1.10	3.29	0.84	0.43	1.07	4.51	0.77	6.96	0.80	1.99	0.56
TS1	13.19	0.99	3.01	0.76	0.37	0.99	4.18	0.72	6.13	0.75	1.86	0.53

For series A, each combining method gives smaller forecast errors than those of the two individual models. For the two other series, four of the seven combining methods give better results than those obtained by the individual models. In no case, the composite forecast is worse than the two individuals. This shows once again the advantage of the combining technique. For the three time series, TS1 improves the results compared with the individual forecasting models, especially on time series A (TS1 yields a MSE reduction of 24.84% over the KNN model).

Compared to the other models (individual or aggregated), TS1 realizes the best performance on time series A, and TS1 is on a par with the best regression models on time series B and C. The reason for these interesting results can be seen on figures 2, 3, and 4 which plot the out-of-sample TS1 combining functions (grid plots) for the three time series (the black filled circles represent the observed data points). These surfaces show the general nature of TS1 systems; the combining function is nonlinear for time series A, almost linear for time series B and linear for time series C. In some sense, this shows that the "degree of nonlinearity" of the TS1 output can be adjusted according to the information available in the learning and validation sets. This essential feature allows the approximation system to account for different types of potentially complex nonlinear relationships.

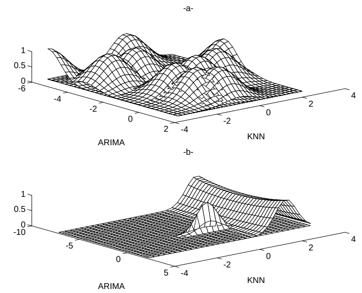

Figure 1: (a) The 10 initial input gaussians. (b) The final input structure (series A)

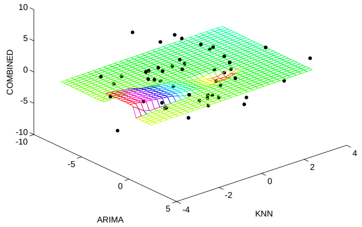

Figure 2: TS1 combining surface (series A)

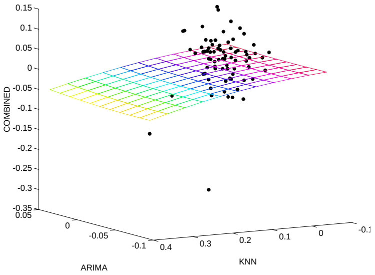

Figure 3: TS1 combining surface (series B)

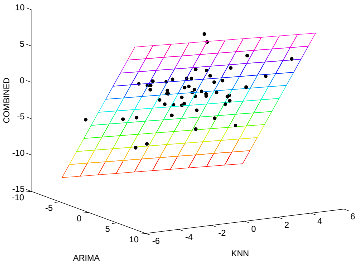

Figure 4: TS1 combining surface (series C)

Conclusions

The results of this study point out the advantage of using Takagi-Sugeno models as nonlinear combining methods compared with purely linear approaches. Our method operates a soft linear mixture of the forecasting models and is in this sense simpler and more readable that the neuronal approach developed in [6]. Practically, the graphs of the combining functions have shown the flexibility of our combining system since the generated surfaces can be nonlinear, almost linear, or purely linear according to the type of information available in the data sets.

References

[1] M.R. Anderberg, Cluster analysis for applications, Academic Press, New York, 1973.

[2] O.D. Anderson, Time Series Analysis and Forecasting, Butterworths, London, 1976.

[3] J.S. Armstrong and F. Collopy, Error measures for generalizing about forecasting methods: Empirical comparisons, *International Journal of Forecasting*, 8, 69-80, 1992.

[4] J.J. Buckley, Sugeno type controllers are universal controllers, *Fuzzy Sets and Systems*, 53, 299-304, 1993.

[5] R.T. Clemen, Combining forecasts: A review and annotated bibliography, *International Journal of Forecasting*, 5, 559-583, 1989.

[6] G. Donaldson and M. Kamstra, Forecasts Combining with Neural Networks, *Journal of Forecasting*, 15, 49-61, 1996.

[7] J.D. Farmer and J.J. Sidorowich, Predicting chaotic time series, *Physical Review Letters*, 59, 845-848, 1987.

[8] A. Fiordaliso, A pruning method for the self-structuring of fuzzy systems applied to function approximation, *Proceedings of the EUFIT96 conference*, 1, 581-586, 1996.

[9] A. Fiordaliso, A nonlinear forecasts combination method based on Takagi-Sugeno fuzzy systems, *IMAGE Technical Report 96.20*, 1996.

[10] V. Gorrini, T. Salomé and H. Bersini, Self-Structuring systems for function approximation, *Proceedings of the FUZZ-IEEE/IFES*, 1, 131-139, 1995.

[11] Granger, C.W.J. and R. Ramanathan, Improved methods of combining forecasts, *Journal of Forecasting*, 3, 197-204, 1984.

[12] K. Hornik, M. Stinchcombe and H. White, Multilayer feedforward networks are universal approximators, *Neural Networks*, 2, 359-366, 1989.

[13] R.A. Jacobs, M.I. Jordan, S.J. Nowlan and G.E. Hinton, Adaptive Mixture of Local Experts, *Neural Computation*, 3, 79-87, 1991.

[14] T.C. Mills, The econometric modelling of financial time series, Cambridge University Press, Cambridge, 1993.

[15] T. Poggio and F. Girosi, Networks for approximation and learning, *Proceedings of the IEEE*, 78, 1481-1497, 1990.

[16] S. Schaal and C.G. Atkeson, From Isolation to Cooperation: An Alternative View of a System of Experts. In D.S. Touretzky, M.C. Mozer and M.E. Hasselmo (eds.): Advances in Neural Information Processing Systems 8, MIT Press, Cambridge, 1996.

[17] F.M. Silva and L.B. Almeida, Acceleration techniques for the backpropagation algorithm. In Lecture Notes in Computer Science 412: Neural Networks, Springer-Verlag, New-York, 1990.

[18] D.E. Specht, A General Regression Neural Network, *IEEE Transactions on Neural Networks*, 2, 568-576, 1991.

[19] T. Takagi and M. Sugeno, Fuzzy identification of systems and its application to modelling and control, *IEEE Transactions on Systems, Man, and Cybernetics*, 15, 116-132, 1985.

[20] L. Xu, A. Krzyzak and A. Yuille, On Radial Basis Function Nets and Kernel Regression: Statistical Consistency, Convergence Rates and Receptive Fields, *Neural Networks*, 7, 609-628, 1994.