Analysis Improvement of Takagi-Sugeno Fuzzy Rules Using Convexity Constraints

Antonio Fiordaliso
Faculté Polytechnique de Mons - MathRO
9, rue de Houdain, 7000 Mons, Belgium
E-mail: fiordal@mathro.fpms.ac.be

Abstract

In this paper, we present a modification of the GTS (Generalized Takagi-Sugeno) model [6]. The key idea is to constraint the conclusions of each rule to perform a convex combination of the input patterns. This constraint allows to interpret each rule as an input patterns mixer and therefore contributes to a better understanding of the system inference.

Introduction

Takagi-Sugeno (TS) fuzzy rules differ mainly from conventional fuzzy rules in that their conclusions are not fuzzy sets but (crisp) polynomial functions [10]. This feature should allow to express complicated behaviours with a small number of rules. Unfortunately, the major drawback of TS rules is that they are less intuitive than conventional fuzzy rules such as Mamdani ones.

One part of our research is motivated by the construction of Takagi-Sugeno-based inference systems allowing for better interpretation and analysis. In view of this, we have proposed a modified version of TS systems called GTS (Generalized TS systems, see section 1) that should allow for possible intuition and interpretation [6].

This paper proposes a second modification of TS systems that improves the readability of the resulting model when it is applied for combining tasks. The key idea that will be developed here consists in constraining the conclusions of each rule to perform a convex combination of the inputs. In this paper, we will apply this system to the case of individual forecasts combination.

1 GTS fuzzy systems

A typical single antecedent fuzzy rule R_k in a first order Generalized Takagi-Sugeno model (GTS, in the sequel) has the following form:

$$R_k: \quad (\gamma_k) \qquad \textbf{If } \mathbf{x}_t \text{ is } A_k \textbf{ then } \hat{y}_{t,k} = \mathbf{x}_t'\boldsymbol{\beta}_k, \quad k = 1, 2, \ldots, c$$

where $\mathbf{x}_t = (x_{t1}, \ldots, x_{tn})'$ is the input variable ($\mathbf{x}_t \in \mathbb{R}^n$), A_k is a fuzzy set of $\mathbb{R}^n$ with membership function μ_{A_k}, $\boldsymbol{\beta}_k = (\beta_1, \ldots, \beta_n)'$ and $\gamma_k = g(\rho_k) = 1/1 + e^{-\rho_k}$ where ρ_k is a new real parameter associated to each rule R_k.

Output $\hat{y}_t$ relative to input $\mathbf{x}_t$ obtained after aggregating a set of c GTS-rules can be written as a weighted sum of the individual conclusions: $\hat{y}_t = \sum_{k=1}^{c} \pi_k(\mathbf{x}_t)\hat{y}_{t,k}$ with $\pi_k(\mathbf{x}_t) = \gamma_k \mu_{A_k}(\mathbf{x}_t) / \sum_{j=1}^{c} \gamma_j \mu_{A_j}(\mathbf{x}_t)$.

γ_k can be interpreted as a sensitivity parameter: a small value for γ_k means that the corresponding rule can be deleted from the system without altering too much the system output value. The γ_k parameter can be seen as the *expertise level* of R_k, while μ_{A_k} informs about its *expertise domain*. Remark that if one chooses for g any constant ($g(\rho_k) = \gamma, \quad k = 1, \ldots, c$), then the GTS system results in a conventional Takagi-Sugeno system. System output $\hat{y}_t$ is a nonlinear function of the new parameter ρ_k and one can use any nonlinear optimization procedure for the ρ_k adjustment.

From a parameters identification point of view, GTS systems can be trained essentially in two ways: on the one hand, global learning strategies are based on the minimization of a global quadratic cost function and can be achieved using nonlinear optimization tools in conjunction with pseudo-inverse methods or orthogonal decompositions techniques for the identification of the linear parameters $\boldsymbol{\beta}_k$ (see [7]). On the other hand, local learning strategies minimize a sum of c local costs [1]. In this framework, nonlinear and linear parameters can be identified respectively by projection of product space fuzzy clusters [1, 8] and by weighted least squares [7].

It appears that the behaviour of the experts (rules) is completely different according to the kind of learning method used. When local learning is considered, these experts compete for the determination of their expertise domains, while they cooperate in the other case. As a result, more interpretable fuzzy rules are obtain by local learning since the competition removes the coupling (or cooperating) effects between the system parameters [1, 7].

In [6, 7], we have focus our attention on the problem of knowing how many rules to consider in a GTS fuzzy model and we have proposed a decremental (or pruning) algorithm (DEC) that progressively detects and removes redundant rules from the system till stabilization occcurs. DEC pruning operations rely on the observation of the γ_k.

2 Combining forecasts

Many studies and empirical tests have shown the advantage of aggregating forecasts in practice [3]. The usual approach to combining forecasts consists of taking a vector $\mathbf{F}_t(h) = (F_{t1}(h), F_{t2}(h), \ldots, F_{tp}(h))'$ of forecasts at time t with horizon h (in the sequel, we will omit h) and constructing a new (combined) forecast C_t as

$$C_t = w_{t0} + \mathbf{F}_t'\mathbf{w}_t \quad (1)$$

where $\mathbf{w}_t = (w_{t1}, \ldots, w_{tp})' \in \mathbb{R}^p$ is the combining vector computed at time t.

It is clear that output $\hat{y}_t$ of a GTS fuzzy system performs the combination given in relation (1) when $\mathbf{x}_t = (1, F_{t1}, F_{t2}, \ldots, F_{tp})'$. Rewriting vector $\boldsymbol{\beta}_k$ like this: $\boldsymbol{\beta}_k = (\beta_0, \beta_1, \ldots, \beta_p)'$, the combination weights are then computed as follows: $w_{tj} = \sum_{k=1}^{c} \pi_k(\mathbf{x}_t)\beta_{kj}, \quad j = 0, \ldots, p$. This decomposition property results from the linearity of the local output models. Hence, combination (1) could not be achieved with other fuzzy systems because they actually realize nonlinear mixture operations. The idea of exploiting the forecasts vectors localization $(\mu_{A_k}(\mathbf{x}_t))$ is new in the field of combining forecasts and has been applied with success in [5].

3 Constrained GTS systems

In this section, we explain how it is possible to allow for a better interpretation of the linear combinations $\hat{y}_{t,k}$ operated by the local experts. The approach developed here is to constraint the local experts in order to achieve convex combinations of the forecasts: $\beta_{k0} = 0; \quad \sum_{j=1}^{p} \beta_{kj} = 1; \quad \beta_{kj} \geq 0 \quad (j = 1, \ldots, p)$. This has the immediate advantage of interpreting β_{kj} as the confidence level of expert k relatively to forecast F_{tj}. In the sequel, we will denote by 01+ this set of constraints. A GTS-01+ rule R_k can thus be interpreted as a mixer whose action can be modulated by γ_k. This particular interpretation of a fuzzy rule is new and may help in giving more intuition about TS fuzzy systems processing operations. The β_{kj} identification can be achieved using the LSIE method (Least Squares with Inequality and Equality constraints) [9].

4 Empirical results

The time series considered here is related to the fluctuations in a far-infrared NH_3 Laser (see [7]). This time series exhibits a low-dimensional chaotic behaviour. In [7], we've compared the forecasting ability of 3 models at horizon $h = 6$; KNN-FS is a K Nearest Neighbours method introduced by Farmer and Sidorowich [4], DRNN (Dynamic Recurrent Neural Network) is a totally interconnected recurrent neural network [11] and GTS-DEC is the fuzzy system described in section 1.

The first 400 data points (out-of-sample forecasts) are used for learning purpose and the last 100 patterns are dedicated to the test. We've trained a GTS-DEC system with $c_{init} = 20$ initial rules with local (l), constrained local (l01+) and global (g) strategies. Local learning is achieved by method FMLE+WLS: we use the Fuzzy Maximum Likelihood Estimates clustering method (FMLE) [8] for the identification of μ_{A_k} and the Weighted Least Squares technique (WLS) for the $\boldsymbol{\beta}_k$ parameters. Method (l01+) is FMLE+WLS where the local linear models are adjusted with the LSIE algorithm which ensures the convexity constraints (01+) to be checked. The global learning strategy uses the Levenberg-Marquardt (LM) nonlinear least squares method and the Singular Values Decomposition (SVD) method for the $\boldsymbol{\beta}_k$ optimization.

The results are reported on table 1. The error criteria NER and MAE are respectively the normalized error and the mean absolute error (see [7]). We compare our forecasts combination system with other traditional combining methods including the popular simple average of individual forecasts (SAV), variant of regression based methods and the well-known bayesian method OUTPERF (outperformance method) proposed by Bunn [2]. The classical linear regression model is denoted by 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 (see [3] for the motivation for choosing the preceding regression combining schemes). Note that the combining weights generated by REG01+ and OUTPERF are constrained in the 01+ sense and thus, a comparison with GTS-DEC-3-l01+ is possible.

Models	NER	MAE
DRNN-5	0.2683	11.1687
KNN-FS	0.2174	6.7038
GTS-DEC-19	0.0866	2.7328
SAV	0.1473	5.5069
REG	0.2168	7.1174
REG0	0.2070	6.4590
REG+	0.2168	7.1174
REG01	0.1975	6.4388
REG01+	0.1975	6.4388
OUTPERF	0.1149	4.0743
GTS-DEC-5-l	0.1104	4.0397
GTS-DEC-10-g	0.0989	3.8086
GTS-DEC-3-l01+	0.0884	3.1936

Table 1: **Results on test set (Laser NH_3 time series).**

Respectively 5, 10 and 3 final rules are obtained by DEC for the cases (l), (g) and (l01+). We remark the fuzzy combinations are better (in terms of NER and MAE) than those produced by the 7 other combining systems. Notice however that only GTS-DEC-3-l01+ competes with the best forecasting model (GTS-DEC-19). The fact that one forecasting model is more accurate than the others leads us to suspect that the best combining strategy is in fact the models *selection*. It appears that the selection operated by GTS-3-l01+ is better than the one produced by the 01+ constrained methods REG-01+ and OUTPERF.

The mixtures achieved by GTS-DEC-3-l01+ are reported in table 2. It appears that the GTS-3-l01+ mixing operations consist in considering 3 combinations of 2 forecasting models; rule 1 deals with the 2 best models with a preference for model GTS-DEC-19, while rule 2 and rule 3 combine DRNN and KNN-FS giving more weight to the latter. These 3 strategies are next mixed via the defuzzification process relative to system GTS.

	$\boldsymbol{\beta}_1$	$\boldsymbol{\beta}_2$	$\boldsymbol{\beta}_3$
DRNN	0.0000	0.2212	0.3474
KNN-FS	0.1725	0.7788	0.6526
TSG-19	0.8275	0.0000	0.0000
	Rule 1	Rule 2	Rule 3

Table 2: **Mixtures processed by system GTS-DEC-3-l01+ (Laser NH_3 time series).**

Let us consider the following quantities Δ_t defined as follows: $\Delta_t = \max_{i \neq j} |F_{ti} - F_{tj}|, \quad 1 \leq i, j \leq p$. The evolution of Δ_t over time enables to detect zones where the forecasts values are similar, or as a contrary, dissimilar.

Observation of figures 1 and 2 gives more details about the combining process. Subplots b, c and d report the time evolution of $\pi_k(\mathbf{x}_t)$. The graph of fig. 1e shows 3 distinct zones relative to the time evolution of Δ_t (zones A, B and C). Actually, when the 3 forecasts values are quite similar (zones B), rule 2 is dominant and the GTS-DEC-3-l01+ selects model KNN-FS (see subplots b, c, d and e). When the dissimilarities Δ_t are more important (zones A), confidence goes to model GTS-DEC-19 (rule 1). One can see rule 3 is never activated, except when large dissensions exist between the forecasting models (zones C). These zones are relative to sudden transitions observed in the time series between high and low amplitude oscillations (see fig. 1a). These structural changes are difficult to forecast for the 3 models. In such cases, the fuzzy combining system activates rules 1 and 3 simultaneously and proceeds to a weighted average of the 3 forecasting models.

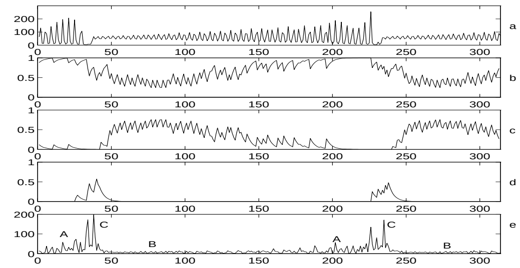

Figure 1: **(a) Laser NH_3 time series (learning part). (b) $\boldsymbol{\pi}_1(\mathbf{x}_t)$. (c) $\boldsymbol{\pi}_2(\mathbf{x}_t)$. (d) $\boldsymbol{\pi}_3(\mathbf{x}_t)$. (e) Δ_t.**

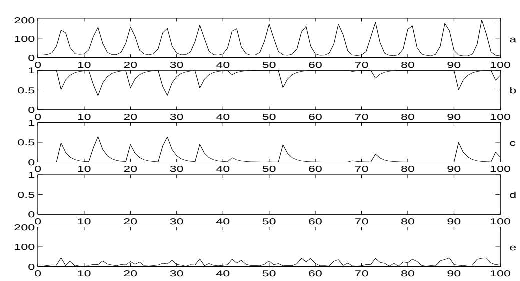

Figure 2: **(a) Laser NH_3 time series (testing phase). (b) $\boldsymbol{\pi}_1(\mathbf{x}_t)$. (c) $\boldsymbol{\pi}_2(\mathbf{x}_t)$. (d) $\boldsymbol{\pi}_3(\mathbf{x}_t)$. (e) Δ_t.**

In order to show the importance of the input patterns localization for the combining weight assignment procedure, we have represented the $\mathbf{F}_t$ scatter plot on figures 3 and 4 and we have placed a bar of length $100 \times \pi_k(\mathbf{x}_t)$ on each data point $\mathbf{F}_t$. Observation of fig. 3 reveals that rule 2 is mainly dominant in the central region of the cloud - where the forecasts are similar - while rule 1 is influent on a greater region spreading from north to south. Activation of rule 3 is linked to the presence of points far from the cloud (outliers). These could be seen as anomalies, or atypic points describing brutal transitions localized in zones C (cf. fig. 1e). Such zones do not appear in test phase (see fig. 2e) and so rule 3 does not cooperate in the forecasts consensus (see fig. 2d).

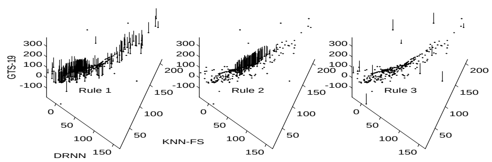

Figure 3: **Learning phase. The scatter plot shows the input data points $\mathbf{F}_t$ (the length of the bars is $100\times\pi_k(\mathbf{x}_t)$).**

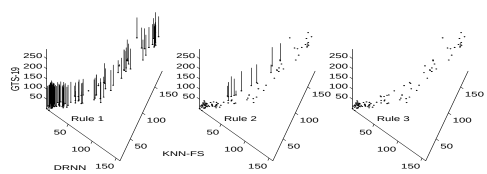

Figure 4: **Testing points $\mathbf{F}_t$.**

The main objective of this experiment was to show that a GTS-DEC-l01+ model is fully transparent to interpretation and analysis. The action of the 3 final rules - automatically obtained by DEC - are well understood. This interpretation is possible because of the 01+ constraints and the local learning that tends to dissociate the rules contribution.

5 Conclusions

This paper presents a modification of the GTS (Generalized Takagi-Sugeno) model [6] where each rule performs a convex combination of the input patterns. Since the parameters of the linear conclusions reflect the confidence levels each rule (expert) puts in each input pattern component, the proposed system may be useful for a wide range of applications where a consensus has to be performed, including controllers and patterns classifiers aggregation.

Focusing our attention on forecasts combination, it appears that the constrained rules perform soft selections of the individual models guided by the localization of the forecasts vector in the input space. This feature has been exploited in a practical case from which intuitive combining strategies have emerged.

References

[1] R. Babuška, Fuzzy Modeling and Identification, Thesis, Delft University of Technology, 1996.

[2] D.W. Bunn, A Bayesian approach to the linear combination of forecasts, Operational Research Quarterly 26 (1975) 325-329.

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

[4] J.D. Farmer, J.J. Sidorowich, Predicting chaotic time series, Physical Review Letters 59 (1987) 845-848.

[5] A. Fiordaliso, A nonlinear forecasts combination method based on Takagi-Sugeno fuzzy systems, to appear in International Journal of Forecasting 14 (1998).

[6] A. Fiordaliso, A pruning method for the self-structuring of fuzzy systems applied to function approximation, in: Proc. of the EUFIT96 conference, vol. 1, 1996, pp. 581-586.

[7] A. Fiordaliso, Contribution des systèmes neuronaux et flous à la prévision des séries temporelles, Thesis, University of Mons-Hainaut, 1998.

[8] I. Gath, A.B. Geva, Unsupervised optimal fuzzy clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (1989) 773-781.

[9] C.L. Lawson, R.J. Hanson, Solving Least Squares Problems, Prentice Hall, New York, 1974.

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

[11] R. Teran, J.P. Draye, D. Pavisic, G. Calderon, G. Libert, Predicting a Chaotic Time Series Using a Dynamical Recurrent Neural Network, in: B.G. Mertzios and P. Liatsis (Eds.), Proc. IWISP '96, Elsevier Science, 1996, pp. 115-118.