s c c h
software competence center
hagenberg

FUZZY.LOGIC.LAB.LINZ

Abstracts of the FLLL/SCCH Master and PhD Seminar

Softwarepark Hagenberg

SCCH, Room 0/10

April 28, 2006

Software Competence Center Hagenberg
Hauptstrasse 99
A-4232 Hagenberg
Tel. +43 7236 3343 800
Fax +43 7236 3343 888
www.scch.at

Fuzzy Logic Laboratorium Linz-Hagenberg
Hauptstrasse 99
A-4232 Hagenberg
Tel. +43 7236 3343 431
Fax +43 7236 3343 434
www.flll.jku.at

Program

Session 1 (Chair: Ulrich Bodenhofer) 9:00–11:00

9:00 Rainer Geschray:
Fuzzy Sliding Mode Control

9:30 Werner Groißböck:
HDFormGen: A Fast Nonlinear Approximation Formula Generator for Very High Dimensional Data Based on Variable Selection and Genetic Programming

10:00 Leila Muresan:
Introduction to Markov Random Fields with Applications to Microscopy Image Processing

10:30 Martina Daňková:
Integral Based Aggregation Operators in the Theory of Fuzzy Approximation

11:00 Coffee Break

Session 2 (Chair: Werner Groißböck) 11:30–13:00

11:30 Martin Štěpnička, Edwin Lughofer, Viktor Pavliska:
Comparison of Data-Driven Fuzzy Modelling Methods tested on NOx Data

12:00 Thomas Vetterlein:
Po-group Representations of MTL-Algebras — How Far Can We Get?

12:30 Martina Daňková, Radek Valášek:
Image Fusion Using Fuzzy Transform

13:00 Lunch

Fuzzy Sliding Mode Control FSMC

FLLL/SCCH Master and PhD Seminar, 2006-04-28

Rainer Geschray,
Johannes Kepler University, Linz, Austria,
k0030408@students.jku.at

April 24, 2006

Abstract

In the first paragraph the well know principle of sliding mode control is presented. There exist many different approaches using the sliding mMode controller as foundation for implementation with fuzzy formulations. In [3] a critical review of Fuzzy Sliding Mode Control FSMC approaches is presented. This knowledge forms a kind of marker for the continuative work.

1 Introduction

In literature there are two different classes for the term *Fuzzy Sliding Mode Control FSMC*. Firstly, there are algorithms based on the traditional sliding mode control, but the signum function is replaced by a fuzzy map. Secondly, there are algorithms which try to approximate the input/output map of the traditional sliding mode control. The next section deals with a short review to the traditional sliding mode control. In section 3 different approaches for FSMC are summarized.

2 Sliding Mode Control

With the state space vector

$$\mathbf{x} = \begin{bmatrix} x & \dot{x} & \cdots & x^{(n-1)} \end{bmatrix}^T \tag{1}$$

the plant model has the form

$$x^{(n)} = f(\mathbf{x}(t)) + u(t) + d(t), \tag{2}$$

where $u(t)$ is the control signal and $d(t)$ is an unknown disturbance. The control objective is to set the control signal $u(t)$, such that the control error $\mathbf{e} = \mathbf{x}_d - \mathbf{x} = \begin{bmatrix} e & \dot{e} & \cdots & e^{(n-1)} \end{bmatrix}^T$ becomes zero $\mathbf{e} = \mathbf{0}$ with $\mathbf{x}_d$ as the desired time variing state. This is indirectly implemented with the differential equation

$$\begin{aligned} 0 &= q(\mathbf{e}) = \left(\frac{\partial}{\partial t} + \lambda\right)^{n-1} e \\ &= \binom{n-1}{0} e^{(n-1)} + \binom{n-1}{1} \lambda e^{(n-2)} + \cdots + \lambda^{n-1} e \\ &= e^{(n-1)} + g_\lambda(\mathbf{e}), \end{aligned} \tag{3}$$

where $\lambda > 0$. If the control error fulfilles (3) he is going to get zero for an arbitrary initial state $\mathbf{x}_0$, i.e. if $q(\mathbf{e}) = 0$ the original control objective is reached! The function $q^2(\mathbf{e})$ is definitely always positive except for the control objective $q(\mathbf{e}) = 0$. Geometrical, $q(\mathbf{e}) = 0$ defines a hyperplane in the n-dimensional space of $\mathbf{e}$, the so called *sliding surface*. Therefore, if

$$\frac{\partial}{\partial t}\left(q^2(\mathbf{e})\right) < -2\eta\,|q(\mathbf{e})| \quad \eta \geq 0, \tag{4}$$

holds, $q^2(\mathbf{e})$ becomes zero and so $q(\mathbf{e})$, too. Why was the original control objective replaced with this overhead? (4) is equivalent to

$$q\dot{q} < -\eta\,|q|\,, \tag{5}$$

i.e.

$$\dot{q}\mathrm{sgn}(q) < -\eta, \tag{6}$$

which is of order one (instead of order n such the original objective)! With (6) the plant (2) is forced to reach the hyperplane and stay there for all time

(in the time invariant case)! Differentiation of (3) results with (2) and (6) in

$$\dot{q} = e^{(n)} + g_\lambda(\dot{\mathbf{e}}) = x_d^{(n)} - x^{(n)} + g_\lambda(\dot{\mathbf{e}})$$
$$\dot{q} = g_\lambda(\dot{\mathbf{e}}) - f(\mathbf{x}) - u - d + x_d^{(n)}$$
$$\left(g_\lambda(\dot{\mathbf{e}}) - f(\mathbf{x}) - u - d + x_d^{(n)}\right)\operatorname{sgn}(q) < -\eta. \quad (7)$$

If the plant model is split up in $f = f_0 + \Delta f$, where f_0 is the (correct) nominal model and

$$|\Delta f| < F$$

is the model uncertainty, the control signal (constant U not specified yet) can set to

$$u = -f_0(\mathbf{x}) + g_\lambda(\dot{\mathbf{e}}) + x_d^{(n)} + U\operatorname{sgn}(q) \quad (8)$$

and the inequality (7) becomes

$$(-\Delta f(\mathbf{x}) - d)\operatorname{sgn}(q) - U < -\eta.$$

With the upper boundary for the disturbance $|d| < D$, the constant in (8) becomes $U \geq F+D+\eta$, thus the sliding mode controller becomes with the equals sign

$$u = -f_0(\mathbf{x}) + g_\lambda(\dot{\mathbf{e}}) + x_d^{(n)} + (F + D + \eta)\operatorname{sgn}(q). \quad (9)$$

Remarks:

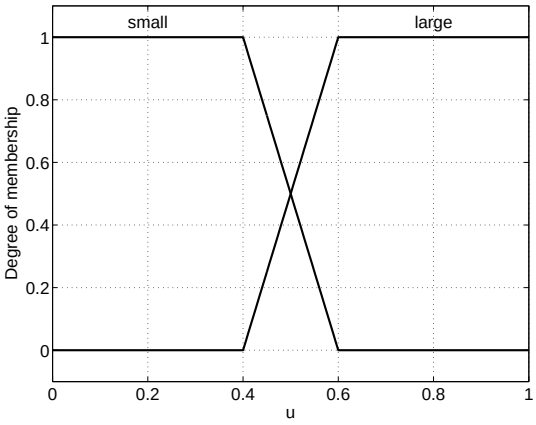
- The calculation of the error derivatives ist problematic in the presence of noise (needs a nonlinear observer for state estimations with an accurate plant model).
- Parameters λ and F influence the robustness of the overall system (expansion to $n-1$ parameters with $q(e) = \sum_{i=0}^{n-2} c_i e^{(i)} + e^{(n-1)}$, where the polynom $c(s) = s^{n-1} + c_{n-2}s^{n-2} + \cdots + c_1 s + c_0$ has zeros with negative real part).
- With the signum function in (9) is the control variable *not continual*, i.e. *chattering* along the sliding surface occures (substitution with a saturation function)!

3 Fuzzy Sliding Mode Control

3.1 Why a Fuzzy Approach?

A main focus of this thesis lies on the control of complex industrial processes, where only sparse analytical knowlegde is available. Instead, there exists a data based model, e.g. Takagi-Sugeno model, to describe the input/output behaviour of the plant. In [5] is an illustrative example (10,11) of the interpolation characteristic of a first-order TS-model presented. Figure 1 shows the membership functions and figure 2 illustrates the interpolation of the two local linear models.

Figure 1: membership functions of the example in [5]



$$\text{If} \quad u = small \quad \text{Then} \quad y = u \quad (10)$$
$$\text{If} \quad u = large \quad \text{Then} \quad y = 5u - 2.5 \quad (11)$$

Conclusion:

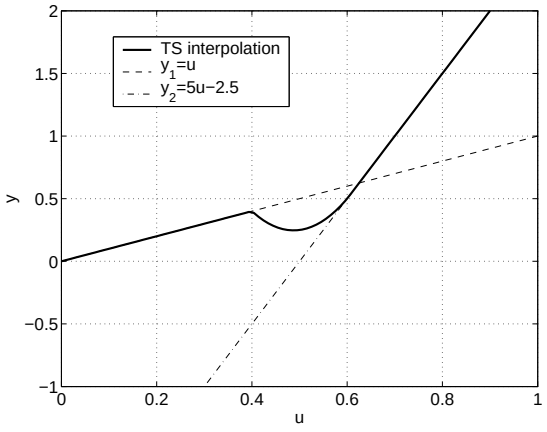
- Without handling the interpolation (e.g. with local linearization suggested in [5]) a traditional controller design is problematic (there are negative derivatives during interpolation, whereas in the local linear models derivatives are all positive)!
- Fuzzy control structures are examinded to handle the disadvantageous interpolation characteristic in TS-models (actual topic of thesis)!

Therefore, fuzzy approaches for the sliding mode control are analyzed in a first step.

3.2 Adaptive Sliding Mode Control

In [1, 4] there is an adaptive sliding mode controller on basis of (9) presented. There, the constant U is

Figure 2: interpolation of the two local linear models



replaced through

$$U^*(\mathbf{x}) = |\Delta f(\mathbf{x})| + D + \eta, \tag{12}$$

where asymptotic stability (Lyapunov) is guaranteed with an adaptation of the fuzzy characteristic $u = \sum_i k_i(\mathbf{x}) u_i = \mathbf{u}^T \mathbf{k}(\mathbf{x})$ as approximation of (12) with

$$\dot{\mathbf{u}} = \gamma \, |q| \, \mathbf{k}(x) \quad \gamma > 0, \tag{13}$$

such that $U(\mathbf{x}) \approx U^*(\mathbf{x})$. This could be an interesting approach, especially in the case of time varing processes in combination with an online model refinement!

3.3 Fuzzy Boundary Layer

This section is geared to [3] and summarizes the results in a short manner. Fuzzy boundary layer involves the replacement of the signum function in (9) with a fuzzy map. The objective to use "fuzzy" is to get a smoother transition and therefore a reducing of chattering is obtained.

Methods:

1. De Neyer and Gorez [6]: Fuzzy maps only a saturation function. No significant improvement over more traditional designs was shown. The set of rules and membership functions could possible be approximated by an explicit equation.
2. Palm [7]: comparable results as in [8] but with more overhead.
3. Ghalia and Alouani [8]: Use FSMC with two fuzzy components, one to fuzzify the signum function and one to fuzzify the gain $F + D + \eta$. Analysis in [3] has shown that this procedure duplicates a combination of the saturation function and a simple gain scheduling approach, neither requires a fuzzy approach.

3.4 Fuzzy Lyapunov Function

In this approach a fuzzy controller is defined such that the Lyapunov function

$$V(q) = \frac{1}{2} q^2 \tag{14}$$
$$\dot{V}(q) = q\dot{q} < -\eta \, |q|$$

holds like (5).

Methods:

1. Shih and Lu [9]: Useful for some types of problems, but with limitations (estimation of the state and its derivatives as mentioned yet).
2. Wang [10]: Works with the objective to speed up convergence to the sliding surface (3) and works essentially with a saturation function. Therefore, fuzzy approach could be replaced by an excplicit expression as in the methods before.

4 Conclusion

Sliding mode control was presented in his traditional form. It uses a nonlinear state model with an arbitrary unknown disturbance signal. The controller (9) uses a nominal model and the control error with his derivatives. So, one focus in further work lies on the analysis of how to use a TS-model for $f_0(\mathbf{x})$ and opportunities to estimate the control error $\mathbf{e} = \begin{bmatrix} e & \dot{e} & \cdots & e^{(n-1)} \end{bmatrix}^T$. Especially the influence of model uncertainties and the interpolation characteristic of TS-model will be studied. In [5] an approach for local linearization is presented, which avoids unfavorable transitions between the local linear equations in a TS-model. This could be an important aspect in controller design and is part of further work.

Different approaches for FSMC was summerized as in the work of [3] presented. It seems that there are less improvements for using fuzzy in sliding mode controls, but there will be more analysis to proof this assertion. Therefore, categorized approaches for FSMC are going be analysed due to there capabilities for robustness, steady state behaviour (integral characteristic) and the handling of time variant plants in combination with an online TS-model adaption. Another future aspect could be the extension of the sliding mode control to multiple input multiple output MIMO plants and a conversion to discrete time systems, but this has less priority at that time.

References

[1] K. Michels, F. Klawonn, R. Kruse, A. Nürnberger, *Fuzzy Regelung - Grundlagen, Entwurf, Analyse*, Springer, 2002

[2] J. Abonyi, *Fuzzy Model Identification for Control*, Birkhäuser, 2003

[3] B. O'Dell, *Fuzzy Sliding Mode Control - A Critical Review*, Oklahoma State University - Advanced Controls Laboratory, 1997

[4] C. S. Ting, T. H. S. Li, F. C. Kung, *An approach to systematic design of the fuzzy control system.* Fuzzy Sets and Systems, 77:151-166, 1996

[5] O. Nelles, M. Fischer, *Lokale Linearisierung von Fuzzy-Modellen*, Automatisierungstechnik, 47:217-223, 1999

[6] M. De Neyer, R. Gorez, *Use of fuzzy concepts in adaptive sliding mode control*, Proceedings of the IEEE Conference on Systems, Man and Cybernetics, Vol 1, 443-447, 1994

[7] R. Palm, *Robust control by fuzzy sliding mode*, Automatica, 30(9):1429-1437, 1994

[8] M. B. Ghalia, A. T. Alouani, Sliding mode control synthesis using fuzzy logic, Proceedings of the American Control Conference, Vol 2, 1528-1532, 1995

[9] M. C. Shih, C. S. Lu, Pneumatic servomotor drive a ball-screw with fuzzy-sliding mode position control, Proceedings of the IEEE Conference on Systems, Man and Cybernetics, Vol 3, 50-54, 1993

[10] S. Y. Wang, C. M. Hong, C. C. Liu, W. T. Yand, Design of a static reactive power compensator using fuzzy sliding mode control, International Journal of Control, 63(2):393-413, 1996

HDFormGen: A Fast Nonlinear Approximation Formula Generator for Very High Dimensional Data Based on Variable Selection and Genetic Programming

Werner Groißböck

Department of Knowledge-Based Mathematical Systems
Fuzzy Logic Laboratorium Linz-Hagenberg
Johannes Kepler University Linz, A-4040 Linz, Austria
werner.groissboeck@jku.at

Summary. A new approach for finding nonlinear approximation formulas for very high-dimensional data is presented. This method has been developed for static data analysis, but it can be used for dynamic data analysis as well. The method is based on linear regression, but instead of the original variables we use nonlinear terms with these variables. Such a formula is still linear in the parameters, so ordinary least squares methods can be applied to find the globally optimal parameters. We use an accelerated version of genetic programming to find the optimal nonlinear terms, and we use variable selection methods to select those terms leading to an approximation formula which shows an optimal balance of accuracy and simplicity. In general, evolutionary methods like genetic programming tend to produce many individuals with low fitness. To save computation time, an early stopping strategy in case of low fitness is used. The method was tested with three benchmark data sets (the auto-mpg data set and the CPU data set in the UCI repository http://www.ics.uci.edu/ mlearn/MLRepository.html and the friedman data set in the KEEL repository http://sci2s.ugr.es/keel/). Although these data sets are only low dimensional and thus not in the core application area of our method, for the auto-mpg data set, an approximation formula has been determined, whose accuracy is comparable to the benchmark papers, for the CPU data set, an approximation formula has been achieved which is more exact than most of the benchmark papers, and for the friedman data set, an approximation formula has been determined which is more exact than all of the benchmark papers found so far.

1 Introduction

In the car industry, an engine test bench system is used which can measure up to 1500 variables. From time to time, some parts of the measurement system are in an invalid state, maybe because one of the sensors is overheated. To

safe time and money, such an invalid state has to be detected as soon as possible, and the experiment has to be aborted as soon as possible. So a system is needed, which can detect faults.
For most of the variables measured useful expert knowledge is not available. For this reason, only data driven methods can be used. Different methods are available. The major challenge is that the methods have to deal with a very high dimensionality.

The method HDFormGen (A fast nonlinear **For**mula **Gen**erator for **H**igh **D**imensional Data) can be used to find a nonlinear approximation formula for very high dimensional data. To demonstrate the strength of our approach, the following artificial data set with 201 variables and 800 entries has been constructed: The variables $x1, x2, ..., x200$ are filled with independent standard normally distributed numbers. The remaining variable (which we call y) is determined with the following formula:

$$\begin{aligned} y =& x1 \cdot (0.3 \cdot x5 - 0.6 \cdot (x3 \cdot x5 + x2 \cdot x6) \\ &+ 0.2 \cdot (x2 \cdot x4 \cdot x6 + x2 \cdot x3 \cdot x7 + x3 \cdot x4 \cdot x5 - x5 \cdot x6 \cdot x7)) \end{aligned} \tag{1}$$

We want to find an approximation formula for the variable y. [1] So we want to see if our only data driven method can find any reasonable results. After running our algorithm for half an hour (all our results have been processed on a 1600MHz pentium laptop) the following formula has been achieved:

$$\begin{aligned} y =& 9.4589e-008 \\ &- 0.6 \cdot (x6 \cdot (x2 \cdot x1)) \\ &- 0.6 \cdot ((x3 \cdot x5) \cdot x1) \\ &+ 0.3 \cdot (x1 \cdot x5) \\ &+ 0.2 \cdot ((x7 \cdot x3) \cdot (x2 \cdot x1)) \\ &+ 0.2 \cdot (((x1 \cdot x4) \cdot x5) \cdot x3) \\ &+ 0.2 \cdot ((x6 \cdot x1) \cdot (x4 \cdot x2)) \\ &- 0.2 \cdot ((x1 \cdot (x5 \cdot x6)) \cdot x7) \end{aligned} \tag{2}$$

This formula is nearly identical to a simplified form of the formula in 1. The only difference is the constant $9.4589e-008$, which is caused by the limitations of machine accuracy. The most important question is: Does the algorithm still work, when data sets containing noise have to be analyzed? To answer this question, the data set described above is used again, but now to each variable a certain amount of noise is added, before our algorithm is applied. As noise we use independent standard normally distributed numbers, which are divided by ten.

[1] The estimated standard deviation of y is 0.81222, so it is not zero, which would make the task trivial.

After an average time consumption of about 4.5 hours, the following approximation formula (for the noisy data set) can be achieved:

$$\begin{aligned} y = & 0.0097468 \\ & - 0.56631 \cdot (x1 \cdot (x2 \cdot x6)) \\ & - 0.57815 \cdot ((x1 \cdot x3) \cdot x5) \\ & + 0.28516 \cdot (x1 \cdot x5) \\ & + 0.18876 \cdot (x2 \cdot ((x1 \cdot x7) \cdot x3)) \\ & + 0.18276 \cdot (x1 \cdot (x5 \cdot (x3 \cdot x4))) \\ & - 0.17787 \cdot (x1 \cdot (x5 \cdot (x6 \cdot x7))) \\ & + 0.18482 \cdot (((x4 \cdot x6) \cdot x1) \cdot x2) \end{aligned} \tag{3}$$

This formula is not identical to the formulas above. But if the subterms are compared, then we can see that all the subterms in formula 3 can also be found in formula 2 and vice versa. So the only real differences are the exact values of the parameters before each nonlinear subterm in the formulas. For example for the subterm with $x1, x3$ and $x5$, we get the parameter -0.57815 instead of the parameter -0.6. This slight modification of the parameters is a consequence of the noise that has been added to the data variables. If a data based method is used, and if you have to deal with noisy data, then a certain amount of error in the models achieved can never be avoided.
Conclusion: We have been able to find a formula that is 'nearly' equivalent to the formulas in 1 and 2. The only relevant differences are the real parameters in the formulas. For finding the correct parameters, we use the least squares algorithm, which finds the globally optimal parameter setting. Finally, the correct structure of the formula is found, and the globally optimal parameter setting!

2 The approximation formula generator HDFormGen

In this paper, the new algorithm *HDFormGen* (A **For**mula **Gen**erator for **H**igh **D**imensional Data) is introduced which is able to find an approximation formula with nonlinear terms for a high dimensional regression data set. With this algorithm, formulas similar to the following can been achieved:

$$y = \beta_0 + \beta_1 \cdot x1 \cdot x100 + \beta_2 \cdot \sin(x77) + \beta_3 \cdot \exp(x5/x6)$$

The basic idea of the algorithm is the following:

- The structure of each of the nonlinear terms in the whole formula is found and optimized with the use of genetic programming (see [5]).
- The parameters of the formula can be optimized easily with a least squares algorithm. This can only be done, if the formula is linear in the parameters, so the genetic programming tool must not generate terms which contain additional parameters.

There is another aspect that has to be considered:
The terms that are used in the approximation formula finally shall be as uncorrelated to each other as possible. We want an approximation formula which is on the one hand as simple as possible, and on the other hand as exact as possible. So we have to find the most important nonlinear terms, such that the regression formula based on these terms is as good as possible. Variable selection methods like *forward selection* have been designed to fulfill this task. In *HDFormGen* a variant of forward selection is used. For this reason, the basic concept of *forward selection* will be explained in the following rows:

- At first, the most important variable (or nonlinear term) is selected. This is that variable (or term) which is correlated strongest to the actual dependent y.
- Then the effects of the variables/terms selected so far are subtracted from the original dependent y. This is necessary to avoid that variables that are highly correlated to the first choice will be chosen again and again.
- Then, again the most important variable/term is selected.
- And again, the dependent is modified, such that the effects of the variables/terms chosen already are eliminated.
- Continue in this manner, until enough variables/terms are selected.

3 The new algorithm HDFormGen

3.1 The core of the new algorithm

In the following, the original dependent is called y. At the beginning, the actual dependent is the original dependent $y_{actual} = y$. Later y_{actual} will be modified. The constant term $c = (1, \ldots, 1)^T$ is always the first variable that is chosen. But this variable is not counted as real variable. The algorithm performs the following steps:

1. An accelerated version of genetic programming (including a population of individuals and a crossover operator) is used to generate millions of very simple formulas. We select that formula x_A which is best correlated with the actual dependent y_{actual}. We look only at the absolute value of the correlation coefficient.
2. Then we modify y_{actual} such that all the parts of y that can be approximated with the regressors already chosen are subtracted, setting y_{actual} to $y - \hat{y}(c, x_A)$. Here $\hat{y}(c, x_A)$ is the linear best approximation of y with the use of the regressors c and x_A. We can say, y_{actual} is y made orthogonal to the regressors already chosen.
3. Once again the accelerated version of genetic programming is used to generate millions of very simple formulas. And now we select that formula x_B which is correlated strongest with the actual dependent y_{actual}. We look only at absolute values again.

4. Then once again, y_{actual} is made orthogonal to the regressors already chosen, so we set y_{actual} to $y - \hat{y}(c, x_A, x_B)$.
5. Continue in this manner, until a given number of regressor terms is selected or some other termination criterion is fulfilled.

3.2 The accelerated version of genetic programming - an overview

Stopping the calculation of the correlation coefficient as early as possible, when it can be seen that the checked individual is not worth spending additional time, accelerates the algorithm enormously. But how can this be carried out, if we have a population of individuals and not a single individual? In the following lines the major steps of the accelerated genetic programming algorithm are described.

1. Generate an initial population with *popsize* individuals.
2. Evaluate each individual for $n1$ points of the training data set and estimate the correlation coefficient with the actual dependent by using only these $n1$ points.
3. Determine the $popsize_{small}$ best correlated individuals out of *popsize*, based on the estimated correlation coefficient. We look only at the absolute value of the correlation coefficient.
4. For these $popsize_{small}$ chosen individuals the *e*xact value of the fitness function (i.e. the absolute value of the correlation coefficient) using *a*ll the points of the training data set has to be calculated.
5. Produce a new generation of *popsize* out of the $popsize_{small}$ chosen individuals:
   - Repeat the following, until we have enough new individuals. Choose randomly two of the $popsize_{small}$ individuals and compare their fitness. The better one is called the winner, and the other one is called the loser. Let the winner produce two offsprings, one is an exact copy of the winner, and the other offspring is made via crossover (as crossover partner, one of the $popsize_{small}$ individuals is chosen, which is neither the winner nor the loser).
   - The individual which is the best so far is always copied into the next generation ('elitism').
   - A small part of the new generation is produced in the same way as the initial population. This is one way of avoiding the problem with local optima. A mutation is not needed any more.
6. Go to step 2, until a termination criterion is fulfilled.

- As termination criterion, we usually take that a specific number of generations is reached.
- The parameter *popsize* determines, how many individuals are evolved in the genetic programming algorithm. The parameter *popsize* can take any positive integer number. The larger *popsize* is, the more computation time

is needed, and the better the results are. In our experiments, a *popsize* of 5000 has been used successfully.
- The parameter $n1$ tells the algorithm, how many points are used to get a quick estimation of the correlation coefficient. $n1$ can be an arbitrary positive integer, but $n1$ shall not exceed the number of training data points. In our experiments, settings of $n1 = 30$, $n1 = 50$ and $n1 = 100$ have been used successfully.
- The parameter $popsize_{small}$ determines, how many individuals of the total population are selected to be examined in detail. The value of $popsize_{small}$ shall be much smaller than *popsize*, for example $popsize/10$.

4 Variants of the Formula Generator Algorithm Applied To Standard Benchmark Data Sets

4.1 The data set cpu

The data set 'cpu' can be found in the directory 'cpu-performance' of the UCI-repository, which can be found in the following address:

`http://www.ics.uci.edu/~mlearn/MLRepository.html`

Number of instances: 209
Number of attributes: 10
The data set contains the following attributes:

vendor name: string
model name: string
MYCT: machine cycle time in nanoseconds (integer)
MMIN: minimum main memory in kilobytes (integer)
MMAX: maximum main memory in kilobytes (integer)
CACH: cache memory in kilobytes (integer)
CHMIN: minimum channels in units (integer)
CHMAX: maximum channels in units (integer)
PRP: published relative performance (integer); the dependent variable;
ERP: estimated relative performance via linear regression (integer)

At first we deleted the attributes *vendorname* and *modelname* because our algorithm can not handle strings. Furthermore the data set contains the attribute ERP, which is an old estimation for PRP. So we have to delete the attribute ERP, because we do not want to generate an approximation formula by using the results of an old approximation. This would be too easy. So finally we have only 7 attributes remaining. Before the core of our algorithm has been run, we split the data into two parts: `70%` of the 209 instances have been randomly chosen to play the role of the training data. And the other `30%` play the role of the test-data.

Our algorithm has been started 10 times. Roughly 3.7 seconds are necessary per term for performing the evolutionary part of the algorithm. Totally we received ten approximation formulas, with an average MAE of 23.33 determined for the test data set. The worst MAE is only 25.15, and the best MAE is 23.06. The best formula is the following:

$$\begin{aligned} PRP =& 16.344 \\ &+ 0.0032443 \cdot (sqrtabs((MMIN \cdot (MMAX \cdot CHMAX)))) \\ &+ 0.7936 \cdot ((CACH - CHMAX) - \sin(CHMAX)) \end{aligned} \tag{4}$$

The MSE of this formula is 1394.9, and the RMSE is 37.348. In our standard benchmark paper (see [7]), various different methods have been tried out. The best method leads to an MAE of 38.0. So compared to this paper, our method leads to a more exact approximation.

Additionally, newer papers (see [10], [12], [2] and [1]) have been found, where the data set cpu is used.

Conclusion: In these papers, totally 30 variants of standard methods have been tried out. Only in 5 out of 30 cases, our approximation formula is outperformed.

4.2 The data set friedman

The data set 'friedman' can be found in the KEEL repository, in the following location:

http://sci2s.ugr.es/keel/datasets1.php?SID&codeds=36

In the keel repository, benchmark papers can be found. For the friedman data set, a quite actual (2004) benchmark paper is mentioned via the abbreviation 'Lee04' (see [6]).

We try to design our experiments as similar as possible to the benchmark paper, to get comparable results. In the benchmark paper, the following is done:

'This is a synthetic benchmark data set. Each sample consists of five inputs and one output. The formula for the data generation is $y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4^4 + 5x_5^5 + \varepsilon$, where ε is a Gaussian random noise $N(0,1)$, and $x_1, ..., x_5$ are uniformly distributed over the domain $[0,1]$. 1400 samples were created, of which 200 samples were randomly chosen for network training and 200 samples for validation. The remaining 1000 samples were used for network testing.'

In the KEEL repository, the data sets are already available as described in [6]. So we have a 200 sample training data set, and a 200 sample validation data set, and a 1000 sample test data set. Unlike the benchmark paper, we do

not need any validation data. So we only take the 200 sample training data set to find an approximation formula, and we take the 1000 sample test data set to determine the quality. As a quality measure, here the MSE is used, according to the benchmark data.

Our formula detection algorithm has been run 20 times. Here we need 13 seconds for each term, and 30 second for finding the total formula, because the formula consists of two nonlinear terms, and four seconds are needed in the non-evolutionary part of the algorithm. The best formula that we get is the following:

$$\begin{aligned} out =& 4.8843 \\ &+ 10.1761 \cdot (in4 + \sin((in2 \cdot (in1 + (in1 + in1))))) \\ &- 5.3183 \cdot (\sin((in3 + (in3 + in3))) - in5) \end{aligned} \tag{5}$$

The MAE of this formula is 0.889281, the MSE of this formula is 1.23629, and the RMSE is 1.11189. In the benchmark-paper (see [6]), the best method leads to an MSE of 4.502. So our formula is much more exact.

Cross-validation and the data set friedman

For the dataset friedman, a tenfold cross validation experiment has been performed. For this experiment, a 1200-sample version of the friedman data set has been used, which can be downloaded from the KEEL repository, in the following location:

http://sci2s.ugr.es/keel/datasets1.php?SID&codeds=36

After the cross-validation, we have to calculate the average error measures on the test data files. We get an average MAE of 0.8394133, an average MSE of 1.1127881, and an average $RMSE$ of 1.0537323 .

So with cross validation, we finally get ten formulas. The formula, which reaches the best quality on the corresponding test data set, is the following formula:

$$\begin{aligned} out =& 4.9946 \\ &- 10.1215 \cdot (\sin(((in2 \cdot in1) \cdot (1.051813 \cdot -2.92026))) - in4) \\ &+ 20.4701 \cdot ((in3 \cdot in3) - ((-0.2465477 \cdot in5) + in3)) \\ &+ 2.9015 \cdot ((0.3611782 - (in1 \cdot in2)) \cdot (in1 \cdot \sin(in3))) \end{aligned} \tag{6}$$

This formula reached (on the test data set number 10) an MAE of 0.786382, an MSE of 0.963263, and an RMSE of 0.981459627 . The name of the corresponding test data file in the KEEL repository is 'Friedman-10-10tst.dat', so everybody is invited to check the quality of the formula! It has to be mentioned that here the best formula out of ten has been selected (via the

test data), so we can not expected to get such a result in average. The average qualities have been stated above, and are more important.

Conclusion: For the friedman data set, all the benchmark papers that we found so far (see [3], [6] and [11]), have been outperformed by our method.

References

1. M. Birattari, G. Bontempi and H. Bersini, "Lazy Learning Meets the Recursive Least Squares Algorithm", MIT Press, Advances in Neural Information Processing Systems 11, Cambridge, 1999.
2. M. Ceci, A. Appice and D. Malerba, "Comparing Simplification Methods for Model Trees with Regression and Splitting Nodes", Springer Lecture Notes in Computer Science, Volume 2871/2003, ISBN: 3-540-20256-0, 2003.
3. Q. Fu, S. X. Hu, S. Y. Zhao, "Clustering-based selective neural network ensemble", Journal of Zhejiang University SCIENCE 2005 6A(5), ISSN 1009-3095, doi:10.1631/jzus.2005.A0387, 2005.
4. Hastie, T., Tibshirani, R., and Friedman J. , "The Elements of Statistical Learning: Data Mining, Inference, and Prediction.", Springer Berlin, 2001.
5. J. R. Koza, "Genetic Programming", The MIT Press, Cambridge, Massachusetts, 1992.
6. W. M. Lee, C. P. Lim, K. K. Yuen, S. M. Lo, "A Hybrid Neural Network Model for Noisy Data Regression", IEEE Transactions on Systems, Man and Cybernetics, Part B 34:2, Pages 951-960, 2004.
7. Merz, C. J., Pazzani, M. J., "Combining Neural Network Regression Estimates with Regularized Linear Weights", Advances in Neural Information Processing Systems 9, edited by M.C. Mozer, M.I. Jordan, and T. Petsche, 1997.
8. A. Miller, "Subset Selection in Regression - Second Edition", ISBN 1-58488-171-2 Chapman & Hall/CRC Boca Raton London New York Washington, D.C. 2002.
9. O. Nelles, "Nonlinear System Identification - From Classical Approaches to Neural Networks and Fuzzy Models", ISBN 3-540-67369-5 Springer-Verlag Berlin Heidelberg New York, 2001.
10. N. Rooney, D. W. Patterson, S. S. Anand and A. Tsymbal, "Random subspacing for regression ensembles", The Florida AI Research Society Conference, 2004.
11. D. P. Solomatine, M. B. L. A. Siek, "Flexible and Optimal M5 Model Trees with Applications to Flow Predictions", 6^{th} International Conference on Hydroinformatics, Liong, Phoon & Babovic (eds), ISBN 981-238-787-0 , World Scientific Publishing Company, 2004.
12. R. Setiono, W. K. Leow and J. Y. L. Thong, "Opening the neural network black box: an algorithm for extracting rules from function approximating artificial neural networks", Proceedings of the twenty first international conference on Information systems, Australia, 2000.

INTRODUCTION TO MARKOV RANDOM FIELDS WITH APPLICATIONS TO MICROSCOPY IMAGE PROCESSING

LEILA MURESAN

Introduction

Inspired by statistical physics, Markov random field models in image processing are built by specifying local interactions, which are leading to global models. This work is an introduction to the theory of Markov random fields (MRF), based on the seminal paper [2]. Many successful applications are known in the literature, some of them are summarized in [3], [1].

1. Images and degradation model

The problem considered in [2] is the computation of a maximum *a posteriori* estimate (MAP) of the original image $\mathbf{f}$ given the degraded image $\mathbf{g}$.

The original image is seen as a pair $\mathbf{X} = (\mathbf{F}, \mathbf{L})$, where $\mathbf{F}$ is the matrix of observable pixel intensities and $\mathbf{L}$ is the matrix of (unobservable) edge elements. $\mathbf{F}$ is called *intensity process*, while $\mathbf{L}$ is the *line process*.

Let $\mathbb{Z}_m = \{(i,j) : 1 \leq i,j \leq m\}$ denote the $m \times m$ integer lattice.
$F = \{F_{ij}, (i,j) \in \mathbb{Z}_m\}$ represent the gray levels of the original image.

The suggested relaxation algorithm maximizes the conditional probability distribution of (F, L) given the data $G = g$, i.e., finds the mode of the posterior distribution $P(X = x|G = g)$. The Bayes estimation formulation (also known as maximum *a posteriori* estimation, or penalized maximum likelihood) is: maximize $\log P(G = g|X = x) + \log P(X = x)$ as a function of x, where the second term is a penalty term.

The model of image degradation consists of noise (N), blurring (H) and nonlinearities (Φ), well suited to describe the typical degradations in microscopy images:

$$G = \Phi\left(H(F)\right) \odot N$$

where $\odot$ denotes any invertible operation, such as addition or multiplication. At pixel level, for each $(i,j) \in \mathbb{Z}_m$

$$G = \Phi\left(\sum_{k,l} H(i-k, j-l)F_{kl}\right) \odot \eta_{ij}. \tag{1.0.1}$$

Furthermore F and N are supposed independent stochastic processes (as well as L and N).

For computational reasons, the degradation model 1.0.1 should preserve locality. This is achieved when H is a simple convolution over a small window.

2. Graphs and Neighborhoods

Definition 2.0.1. Let $S = \{s_1, s_2, \ldots, s_N\}$ be a set of sites. A neighborhood system for S is a collection of subsets of S, $\mathcal{G} = \mathcal{G}_s, s \in S$ for which

- $s \notin \mathcal{G}_s$
- $s \in \mathcal{G}_r \iff r \in \mathcal{G}_s$

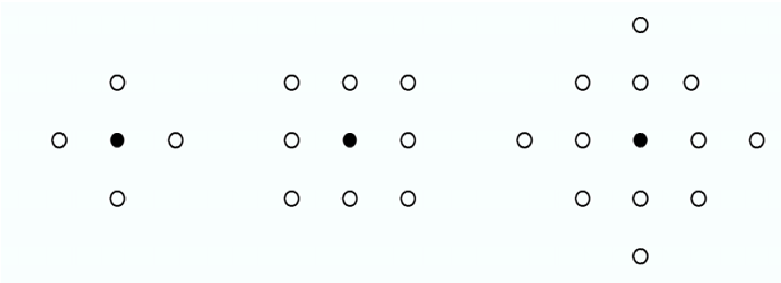


FIGURE 1. Neighborhoods of order 1, 2 and 4

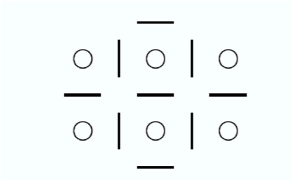


FIGURE 2. Pixel and line neighbors of a line element

Definition 2.0.2. A subset $C \subseteq S$ is a clique if every pair of distinct sites in C are neighbors. The set of cliques is denoted by $\mathcal{C}$.

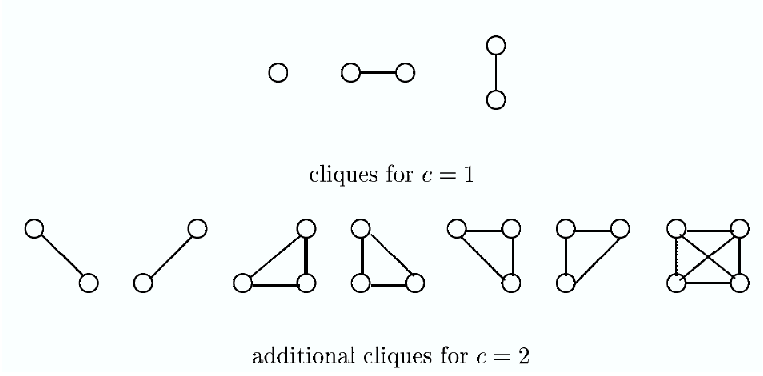


FIGURE 3. Cliques for neighborhoods of order 1 and 2

3. The Hammersley-Clifford theorem

Let (S, G) be an arbitrary graph, $X = \{X_s, s \in S\}$ a family of random variables indexed by S. For simplicity, $X_s \in \Lambda = \{0, 1, 2, \dots L-1\}$

Let Ω be the set of all possible configurations:

$$\Omega = \{\omega = (x_{s_1}, \dots x_{s_N}) : x_{s_i} \in \Lambda, 1 \leq i \leq N\}$$

Definition 3.0.3. X is a *Markov random field* (MRF) over $(S, \mathcal{G})$ if

$$P(X = \omega) > 0, \forall \omega \in \Omega$$

$$P(X_s = x_s | X_r = x_r, r \neq s) = P(X_s = x_s | X_r = x_r, r \in \mathcal{G})$$

The $P(X_s = x_s | X_r = x_r, r \neq s)$ are called the local characteristics of the MRF.

Definition 3.0.4. A Gibbs distribution relative to $(S, \mathcal{G})$ is a probability measure π on Ω having the following representation:

$$\pi(\omega) = \frac{1}{Z} e^{\frac{-U(\omega)}{T}},$$

where

- $U(\omega) = \sum_{C \in \mathcal{C}} V_C(\omega)$, is called the *energy function*
- The family of functions $\{V_C, C \in \mathcal{C}\}$ where $V_C(\omega)$ depends only on the variables $\{x_s : s \in C\}$ is called a *potential*
- $Z = \sum_\omega e^{\frac{-U(\omega)}{T}}$ is a normalizing constant and the corresponding function is called *partition function*
- T is a constant which stands for *temperature*

T controls the degree of "peaking" of π. When T is big, coupling between pixels is loose, the distribution when $T \to \infty$ is uniform. When T is small, the coupling between pixels becomes stronger, the modes are more accentuated, easier to find by sampling. This is the pronciple of annealing applied to the posterior distribution

$$\pi(f, l) = P(F = f, L = l | G = g)$$

in order to find the MAP estimate.

Theorem 3.0.5. *(Hammersley-Clifford) Let $\mathcal{G}$ be a neighborhood system. Then X is an MRF with respect to $\mathcal{G}$ if and only if $\pi(\omega) = P(X = \omega)$ is a Gibbs distribution with respect to $\mathcal{G}$.*

The theorem above provides a practical way of specifying MRF's by specifying appropriate potentials (instead of specifying local characteristics, which would be extremely difficult). A proof of the theorem can be found in [4].

For the problem 1.0.1, in order to find ω which maximizes the posterior distribution for a given g the following expression has to be minimized:

$$U(f, l) + \frac{\|\mu - \Phi(g, \phi(H(F)))\|^2}{2\sigma^2}$$

The identification even of a near-optimal solution is very difficult. The approach suggested in [2] to solve this problem is stochastic relaxation.

4. Stochastic relaxation and the Gibbs sampler

The general computational problems are:

- Sample from the distribution π
- Minimize U over Ω
- Compute expected values

At time t, the the total configuration is $X(t) = (X_{s_1}(t), X_{s_2}(t), \ldots, X_{s_N}(t))$. The starting configuration $X(0)$ is arbitrary.
$X(t)$ can differ from $X(t-1)$ at most in one coordinate : given an ordering sequence of $S : n_1, n_2, \ldots$, then $X_{s_i}(t) = X_{s_i}(t-1), i \neq n_t$.
At time t, for $s = n_t$ a sample for π is generated , meaning that X_{n_t} is updated from the conditional distribution, based on $X_r(t-1), r \in \mathcal{G}_{n_t}$.

The computations are local, and if π is homogeneous, also identical in nature.

The following three theorems proove the correctness of the approach.

Theorem 4.0.6. *(Relaxation) If for each $s \in S$ the sequence n_t, $t \geq 1$ contains s infinetly often, then for every starting configuration $\eta \in \Omega$ and every $\omega \in \Omega$*

$$\lim_{t\to\infty} P(X(t) = \omega | X(0) = \eta) = \pi(\omega) \tag{4.0.2}$$

The Markov chain $X(t), t = 0, 1, 2, \ldots$ has the equilibrium distribution π.

Theorem 4.0.7. *(Annealing) Assume that there exists an integer $\tau \geq N$ such that for every $t = 0, 1, 2 \ldots$ $S \subseteq \{n_{t+1}, n_{t+2}, \ldots n_{t+\tau}\}$. Let $T(t)$ be any decreasing sequence of temperature for which:*

- *$T(t) \to 0$ as $t \to \infty$*
- *$T(t) \geq N\Delta / \log t$ for all $t \geq t_0$ for some integer $t_0 \geq 2$*

Then for any starting configuration $\eta \in \Omega$ and for every $\omega \in \Omega$

$$\lim_{t\to\infty} P(X(t) = \omega | X(0) = \eta) = \pi_0(\omega) \tag{4.0.3}$$

Theorem 4.0.8. *(Ergodicity) Assume that there exists an integer τ such that $S \subseteq \{n_{t+1}, n_{t+2}, \ldots n_{t+\tau}\}$ for all t. Then for every function Y on Ω and for every starting configuration $\eta \in \Omega$*

$$\lim_{t\to\infty} \frac{1}{n} \sum_{t=1}^{n} Y(X(t) = \int_{\Omega} Y(\omega) d\pi(\omega) \tag{4.0.4}$$

References

1. P.B. Chapple, D.C. Bertilone, R.S. Caprari, and G.N. Newsam, *Stochastic model-based processing for detection of small targets in non-gaussian natural imagery*, IEEE Transcations on Image Processing **10** (2001), no. 4.
2. S. Geman and D. Geman, *Stochastic relaxation, gibbs distributions, and the bayesian restoration of images*, IEEE Trans. Pattern Anal. Machine Intell. **6** (1984), no. 6, 721–741.
3. Perez P., *Markov random field and images*, CWI Quarterly **11** (1998), no. 4, 413–437.
4. Won C. S. and Gray R.M., *Stochastic image processing*, Kluwer, 2004.

Integral based aggregation operators in the theory of fuzzy approximation

Martina Daňková

Inst. for research and applications of fuzzy modeling
University of Ostrava

30. dubna 22, 701 03 Ostrava
Czech Republic

Martina.Dankova@osu.cz

1 Introduction

The paper contributes to the field of fuzzy approximation. Fuzzy approximation may be treated as a theory studying approximating functions created using techniques based on the theory of fuzzy sets. It is understood generally in two different ways differing by tool that we have at disposal intended to approximate a given function or a data-set: (1) dependency is expressed by fuzzy function or (2) it is approximated by an ordinary function using techniques based on fuzzy set theory. Both approaches have common feature which lies in transparent interpretability.

Later, we will focus only on the second type of fuzzy approximation techniques. The basic idea behind the approach introduced in this paper is to apply two times aggregation operator (similarly as proposed in [2] or later in [6]):

1'th aggregate data on subsets of input-space, and then

2'nd aggregate this local information into the global one.

The first part of this work may be treated as a generalization of approximation method for continuous functions (discrete data relating to continuous function) introduced by Perfilieva in [7] as the technique called fuzzy transformation. There, the weighted arithmetic mean is used only, while in this approach, approximations using Choquet-like aggregation operators (leading to weighted quasi-arithmetic means) are presented. This aggregation can be seen also as a transformation of the input data with the aim to obtain symmetrical spread around the final central characteristic. Using the Choquet-like aggregation operators, we can capture and eliminate different kinds of noise or an error included in the system etc.

The second part is devoted to approximations based on Sugeno-like integrals [10] and may be treated as an alternative to fuzzy transformation using operations on residual lattice. Unlike in [7], we follow the main idea of applying aggregating operator to local subsets by defining maxitive fuzzy measure on on these subsets and afterwards on the whole domain.

The main goal of this contribution is to investigate whether integral-based aggregation operators (weighted quasi-arithmetic means or weighted maximum etc.) can be used to approximate a given function with an arbitrary precision. Moreover, we hope to find approximating functions which minimize different criterions.

2 Preliminaries

Before turning to the main investigations let us recall some basic notions and concepts.

2.1 Pseudo-operations

Let $[a, b]$ be a closed subinterval of $[0, \infty]$. The operation $\oplus : [a, b] \times [a, b] \to [a, b]$ is called a pseudo-addition if it is commutative, nondecreasing, associative and has a zero element denoted by $\mathbf{0}$.

The operation $\odot : [a, b] \times [a, b] \to [a, b]$ is a pseudo-multiplication if it is commutative, nondecreasing, associative and for which there exists a unit element $\mathbf{1} \in [a, b]$. Let us further suppose that $\odot$ is distributive with respect to $\oplus$ i.e.,

$$x \odot (y \oplus z) = (x \odot y) \oplus (x \odot z).$$

The algebraic structure induced by the above described pseudo-operations, i.e.

$$\mathcal{L} = \langle X = [a, b], \oplus, \odot, \mathbf{0}, \mathbf{1} \rangle, \tag{1}$$

is a semiring. Remark that mathematical analysis over this structure is known as pseudo-analysis.

The following are examples of pseudo-operations:

Case 1. $\mathcal{L}_P = \langle [0, \infty], \vee, \cdot, 0, 1 \rangle$ is the max-product algebra over the extended positive reals.

Case 2. $\mathcal{L}_* = \langle [0, 1], \vee, *, 0, 1 \rangle$, where $*$ is a t-norm.

Case 3. $\mathcal{L}_g = \langle [0, \infty], \oplus, \odot, 0, \mathbf{1} \rangle$ with pseudo-operations defined by a monotone and continuous generator $g : [0, \infty] \to [0, \infty]$ (see e.g. [5]). Moreover, from this class we consider only strict pseudo-additions, that is operations with $\oplus$ strictly increasing on $(0, \infty) \times (0, \infty)$. In this case by Aczél's representation theorem there exists a monotone function $g : [0, \infty] \to [0, \infty]$ such that $g(0) = 0$ and

$$x \oplus y = g^{-1}(g(x) + g(y)).$$

In this case the unique pseudo-multiplication associated to $\oplus$ is given by

$$x \odot y = g^{-1}(g(x) \cdot g(y)).$$

Relating g, we can define also power, pseudo-division and pseudo-subtraction

$$x^{\mathbf{n}} = \underbrace{x \odot \ldots \odot x}_{n-times}, \quad \frac{x}{y}\odot = g^{-1}\left(\frac{g(x)}{g(y)}\right), \quad x \ominus y = g^{-1}(g(x) - g(y)),$$

respectively.

Case 4. $\mathcal{L}_g^{\vee} = \langle [0,\infty], \vee, \odot, 0, \mathbf{1}\rangle$, where $\odot$ and $\mathbf{1}$ are from the previous case.

2.2 Fuzzy measure and special pseudo-integrals

Having on mind the above structure $\mathcal{L}$, we will consider a function $f : X \to X$ and denote $X \times \ldots \times X$ by X^n. Moreover, we consider $\mathcal{A}$ to be the σ-algebra of subsets of X^n, $(X^n, \mathcal{A})$ which is a measurable space. We say that a $\mathbf{m} : \mathcal{A} \to [\mathbf{0}, \mathbf{1}]$ is a fuzzy measure on X^n, if

(1) $\mathbf{m}(\emptyset) = \mathbf{0}$ and $\mathbf{m}(X^n) = \mathbf{1}$,

(2) $\mathbf{m}(I) \leq \mathbf{m}(J)$, whenever $I \subset J \subseteq X^n$.

It is moreover additive, if

(3) $\mathbf{m}(J) = \sum_{x \in J} \mathbf{m}(\{x\})$, $J \subseteq X^n$.

For the sake of brevity, we will write $gf(x)$ instead of $(g \circ f)(x)$ or $g(f(x))$ for arbitrary g, f. Moreover, let us denote a non-decreasing permutation of the input n-tuple $\mathbf{x} \in X^n$ by $\mathbf{x}'$ and $x'_0 = 0$, $x'_{n+1} = \infty$ by convention. Further, we specify a system of sets for $\mathbf{x}'$ associated to $\mathbf{x} \in X^n$ as follows $I'_i = \{x'_i, \ldots, x'_n\}$, i=1,. . . ,n, and $I'_{n+1} = \emptyset$.

Assuming $\mathcal{L}_g$ as in Case 3, Choquet-like integral (introduced in [4]) for f w.r.t. $\mathcal{L}_g$ is related to an additive measure $\mathbf{m}$ and is of the following form:

$$f_g^C(\mathbf{x}) = \int_{X_n}^{\oplus} f \, d\mathbf{m} = \bigoplus_{i=1}^{n} [f(x'_i) \odot (\mathbf{m}(I'_i) \ominus \mathbf{m}(I'_{i+1}))], \tag{2}$$

As stated in [1], f_g^C is an idempotent continuous n-ary aggregation operator which is pseudo-linear and comonotone pseudo-additive with respect to the pseudo-addition $\oplus$ and pseudo-multiplication $\odot$.

Note that if we fix the order of the input n-tuple $\mathbf{x}$ then (2) becomes a weighted quasi-arithmetic mean that is idempotent and bisymmetric aggregation operator (see also [1]).

Sugeno-like integral for f w.r.t. $\mathcal{L}_g^{\vee}$ and a fuzzy measure $\mathbf{m}$:

$$f_g^S(\mathbf{x}) = \int_{X_n}^{\vee} f \, d\mathbf{m} = \bigvee_{i=1}^{n} f(x'_i) \odot \mathbf{m}(I'_i), \tag{3}$$

The integral is named after Sugeno ([10], 1974), where he introduced (3) with $\odot = \wedge$ and even before by Shilkret ([9], 1971) with $\odot = \cdot$. Note that instead of the pseudo-multiplication, we may use an arbitrary t-norm with no zero divisors as published by Weber ([11], 1986) such that (3) is refereed to as Sugeno-Weber integral. In general, we may assume arbitrary t-norm but in this case, (3) may appear to be zero for positive-constant function on set with positive measure. Remind that an arbitrary t-norm $*$ is distributive w.r.t. $\vee$ (Proposition 1.4.6. in [3]), i.e. $(x_1 \vee x_2) * y = (x_1 * y) \vee (x_2 * y)$, therefore, we still keep the important property of max-homogeneity for all $a \in [0, 1]$, i.e.

$$a \vee f_*^S(\mathbf{x}) = f_*^S(a \vee \mathbf{x}),$$

and also comonotone maxitivity, i.e.

$$f_*^S(\mathbf{x}) \vee f_*^S(\mathbf{y}) = f_*^S(\mathbf{x} \vee \mathbf{y}).$$

Note that all aggregation operators based on Sugeno-like integrals are continuous, idempotent aggregation operators which are comonotone maxitive.

3 Mean square method based fuzzy approximation

In the following text, we will assume that $\mathcal{L}$ is of the form (1) and f is a function on a support of $\mathcal{L}$. If d is a metric on X and $D = \{x_i \in X |\ i \in I = \{1, \ldots, n\}\}$ is a data set then (CFD) the classical problem of curve fitting to empirical data (that represent functional dependence f) is formulated as

- Assume that $f(x_i) = P(x_i) + e_i$, for all $x_i \in D$, $i \in I$, $P : X \to X$ and e_i is an error of measurement.
- The structure of function P is known and it is dependent on $r\,(r < n)$ parameters $p_1, \ldots, p_r$. We can write $P(x) = P(x, p_1, \ldots, p_r)$
- Find $\{\bar{p}_j\}_{j \in R}$, $R = \{1, \ldots, r\}$ such that

$$S = \bigoplus_{i \in I} d^{\mathbf{2}}(f(x_i), P(x_i, \bar{p}_1, \ldots, \bar{p}_r)) \text{ is minimal.}$$

We may generalize this problem by assuming classical problem of curve fitting to empirical data w.r.t. a given fuzzy set A on the universe X.

Convention 3.1 *A fuzzy set A on the universe X is specified by the characteristic function of A that is a mapping $\mu_A : X \to [0, 1]$. We denote the fact that A is a fuzzy set on X by $A \underset{\sim}{\subseteq} X$ and for the simplicity, we will use the same symbol for a fuzzy set as well as for its membership function.*

Then we can reformulate the classical problem into the generalized one (CFD-A) as follows:

- Assume $\mathcal{L}$ of the form (1), a metric d on X and $A \subsetneq X$.
- Moreover, let $d(f(x_i), P(x_i)) = e_i$, for $x_i \in D$, $i \in I$, where $P : X \to X$ and e_i is an error of measurement.
- The structure of function P is known and it is dependent on $r\,(r < n)$ parameters $p_1, \ldots, p_r$.
- Find $\{\bar{p}_j\}_{j\in R}$, such that

$$S_A = \bigoplus_{i\in I} A(x_i) \odot d^{\mathbf{2}}(f(x_i), P(x_i, \bar{p}_1, \ldots, \bar{p}_r)) \text{ is minimal.}$$

  This might be understand as: find minimum of S for $x_i \in A$.

3.1 Fuzzy approximation by means of fuzzy partition

Under this notion, we understand an approximation technique that applies to each fuzzy set of a fuzzy partition of the input domain. We specify a parametric set of functions from which we choose a concrete function due to some criterion. In our case, the criterion and a measure of precision is taken as described above.

Definition 3.2 *Let $J = \{1, \ldots, k\}$ be an index set and $\mathbb{A} = \{A_i\}_{i\in J}$ be a system of fuzzy sets on X, i.e. each $A_i \subsetneq X$. We say that $\mathbb{A}$ is fuzzy partition if $\bigoplus_{j\in J} A_j(x) = \mathbf{1}$ for all $x \in X$.*

We specify the following space of functions w.r.t. $\mathcal{L}$, $\{P_i\}_{i\in J}$ system of functions depending on r parameters and the fixed fuzzy partition $\mathbb{A}$:

$$\mathcal{H}_{\mathbb{A}} = (\{\bigoplus_{i\in J}[A_i(x) \odot P_i(x, \mathbf{p}_i)] |\ \mathbf{p}_i \in X^r\}, \oplus, \odot, \mathbf{0}_f, \mathbf{1}_f),$$

where $\mathbf{0}_f(x) = \mathbf{0}$ and analogously $\mathbf{1}_f(x) = \mathbf{1}$ for each $x \in X$. Notice that $\mathcal{H}_{\mathbb{A}}$ is the subspace of the pseudo-linear space

$$\mathcal{H} = (\{f |\ f : X \to X\}, \oplus, \odot, \mathbf{0}_f, \mathbf{1}_f).$$

The mean square error problem for fuzzy approximation by means of fuzzy partition (CFD-$\mathbb{A}$) is given as follows:

- Let d be a metric on X and $f : X \to X$ be a continuous w.r.t. d. Let us denote space of all such a functions by $\mathcal{H}_d$. Note that if all $\{A_i\}_{i\in J}$ and $\{P_i\}_{i\in J}$ are continuous w.r.t. d then $\mathcal{H}_{\mathbb{A}} \subset \mathcal{H}_d \subset \mathcal{H}$.
- Find $\{\mathbf{p}_i\}_{i\in J}$ such that each S_{A_i} of the problem (CFD-A_i) is minimized.

When having a solution $\{\mathbf{p}_i\}_{i\in J}$ relating to $\{P_i\}_{i\in J}$ of (CFD-$\mathbb{A}$), we know that each function from the system $\{P_i\}_{i\in J}$ minimize the mean square error of

the particular fuzzy set of fuzzy partition $\mathbb{A}$. We would need to employ pseudo-derivatives to solve this problem. Therefore, we restrict ourselves to special cases of the structure $\mathcal{L}$ that will alow us to use the classical notion of derivatives.

This special structures will be those that depend on a continuous generator g of pseudo-operations, i.e. $\mathcal{L}_g$ and $\mathcal{L}_g^\vee$. Moreover, the criterion to be minimized is now different. Let us summarize the problem of minimizing mean square error (CFD-$\mathbb{A}, g$) for this special case:

- Let $\mathcal{L} = \mathcal{L}_g (= \mathcal{L}_g^\vee)$ and the metric d on X is defined by

$$d(x, y) = g^{-1}(|g(x) - g(y)|),$$

  moreover, let $f \in \mathcal{H}_d$.

- Find $\{\mathbf{p}_i\}_{i \in J}$ such that each criterion $g(S_{A_i})$, where S_{A_i} is the criterion of the problem (CFD-A_i), is minimized.

The following lemma shows that an arbitrary function from $\mathcal{H}_\mathbb{A}$ minimizes the variance from $\{P_i\}_{i \in J}$ w.r.t. $\mathbb{A}$.

Lemma 3.3 *Consider $\mathcal{L}$ of the form (1), a fuzzy partition $\mathbb{A}$ and $F(x) = \bigoplus_{i \in J}(A_i(x) \odot P_i(x))$.*

1. *If $\mathcal{L} = \mathcal{L}_g$ then*

$$y = F(x) \text{ minimizes } g[\textstyle\bigoplus_{i \in J} A_i(x) \odot (P_i(x) \ominus y)^{\mathbf{2}}] \text{ for each } x \in X.$$

2. *If $\mathcal{L} = \mathcal{L}_g^\vee$ then*

$$y = F(x) \text{ minimizes } g[\textstyle\bigvee_{i \in J} A_i(x) \odot (P_i(x) \ominus y)^{\mathbf{2}}] \text{ for each } x \in X.$$

3.2 Choquet-like integral based approximation

In the sequel, we will fix fuzzy measures in order to specify aggregation operators which are used for the approximation, i.e., for the computation of the local estimation of a function f by basic functions and finally the approximation of f based on the local estimates. Let us remind that weighted quasi-arithmetic mean is a special case of the Choquet-like integral, and the later specification leads to that specific class of aggregation operators.

Assume $\mathcal{L}_g$ and a fuzzy partition $\mathbb{A}$. The mapping $\mathbf{m}_x$ defined as

$$\mathbf{m}_x(\mathcal{X}) = \bigoplus_{A \in \mathcal{X}} A(x),$$

is fuzzy measure of $\mathcal{X} \subseteq \mathbb{A}$ on $\mathbb{A}$ for each $x \in X$.

Moreover, the mapping

$$\mathbf{m}_i(\mathcal{X}) = \bigoplus_{x \in \mathcal{X}} \frac{A_i(x)}{\bigoplus_{y \in D} A_i(y)} \odot,$$

is fuzzy measure of $\mathcal{X} \subseteq D$ on D for each $i = 1, \ldots, k$.

These fuzzy measures determine Choquet-like integrals that are used to specify function $F(x)$ ($= f_g^C$ w.r.t. $\mathbf{m}_x$) and the system of constant functions $\{P_i\}_{i\in J}$ ($= \{f_g^C$ w.r.t. $\mathbf{m}_i\}_{i\in J}$).

Below, we summarize the received formulas:

$$P_i(x) = f_{i,g} = \frac{\bigoplus_{x\in D} A_i(x) \odot f(x)}{\bigoplus_{y\in D} A_i(y)} \odot, \tag{4}$$

is Choquet-like integral w.r.t. $\mathbf{m}_i$ for each $i \in J$. And

$$F_g(x) = \bigoplus_{i\in J} A_i(x) \odot f_{i,g}, \tag{5}$$

is Choquet-like integral w.r.t. $\mathbf{m}_x$ for each $x \in X$.

Definition 3.4 *We say that (5) is fuzzy transform of f w.r.t. the sample set D and fuzzy partition $\mathbb{A}$ on $\mathcal{L}_g$.*

Remark 3.5 *The notion fuzzy transform due to [7]. Motivation comes from Takagi-Sugeno models.*

Clearly, $(f_{1,g}, \ldots, f_{k,g})$ is a solution of the problem (CFD-$\mathbb{A}, g$).

Lemma 3.6 *Let $f_{i,g}$ and $F_g(x)$ be as above. Then,*

$$y = f_{i,g} \text{ minimizes } g[\textstyle\bigoplus_{x\in D} A_i(x) \odot d^{\mathbf{2}}(f(x), y)],$$

Now we are going to show that using fuzzy transform by means of $\mathcal{L}_g$ we are able to approximate a given sample data set of a continuous function with an arbitrary precision.

Theorem 3.7 *Let $f \in \mathcal{H}_d$. Then for arbitrary $n \in \mathbb{N}$ and $\varepsilon > 0$ there exists $\mathbb{A}$ such that F_g w.r.t. D_n and $\mathbb{A}$ on $\mathcal{L}_g$ ε-approximates $f(D_n)$, i.e.*

$$d(f(x), F_g(x)) \leq \varepsilon, \text{ for each } x \in D_n.$$

3.3 Sugeno-like integral based approximation

We will follow the approach from the previous section. For this subsection, let us assume $\mathcal{L}_g^{\vee}$. The mapping

$$\mathbf{m}_x(\mathcal{X}) = \bigvee_{A\in\mathcal{X}} A(x),$$

defines fuzzy measure of $\mathcal{X} \subseteq \mathbb{A}$ on $\mathbb{A}$ for each $x \in X$.

Moreover, the mapping

$$\mathbf{m}_i(\mathcal{X}) = \frac{\bigvee_{x\in\mathcal{X}} A_i(x)}{\bigvee_{y\in D} A_i(y)} \odot, \tag{6}$$

defines fuzzy measure of $\mathcal{X} \subseteq D$ on D for each $i = 1, \ldots, k$.

Analogously as in the previous case, these fuzzy measures determine Sugeno-like integrals that are used to specify function $F(x)$ of the problem (CFD) ($= f_g^S$ w.r.t. $\mathbf{m}_x$) and the system of constant functions $\{P_i\}_{i\in J}$ ($= \{f_g^S$ w.r.t. $\mathbf{m}_i\}_{i\in J}$).

Below, we summarize the received formulas:

$$P_i(x) = f_{i,g} = \frac{\bigvee_{x\in D} A_i(x) \odot f(x)}{\bigvee_{y\in D} A_i(y)} \odot, \tag{7}$$

is Sugeno-like integral w.r.t. $\mathbf{m}_i$ for each $i \in J$. And

$$F_g(x) = \bigvee_{i\in J} A_i(x) \odot f_{i,g}, \tag{8}$$

is Sugeno-like integral w.r.t. $\mathbf{m}_x$ for each $x \in X$.

Definition 3.8 *We say that (8) is fuzzy transform of f w.r.t. the sample set D and fuzzy partition $\mathbb{A}$ on $\mathcal{L}_g^\vee$.*

Analogously, $(f_{1,g}, \ldots, f_{k,g})$ is the solution of (CFD-$\mathbb{A}, g$).

Lemma 3.9 *Let $f_{i,g}$ and $F_g(x)$ be given by (7) and (8), respectively. Then,*

$$y = f_{i,g} \text{ minimizes } g[\textstyle\bigvee_{x\in D} A_i(x) \odot d^{\mathbf{2}}(f(x), y)],$$

Now we are going to show that using Max-transformation we are able to approximate a given data with an arbitrary precision.

Theorem 3.10 *Let $f \in \mathcal{H}_d$. Then for arbitrary $n \in \mathbb{N}$ and $\varepsilon > 0$ there exists $\mathbb{A}$ such that fuzzy transform of f w.r.t. D_n and $\mathbb{A}$ on $\mathcal{L}_g^\vee$ ε-approximates $f(D_n)$, i.e.*

$$d(f(x), F_g(x)) \leq \varepsilon, \text{ for each } x \in D_n.$$

4 Conclusions

The class of aggregation operators is very broad and hence the number of possibilities how to create an approximating function is unlimited. Important is to create such approximation that fulfills required criterion. In this work, we have minimized the residual sum of squares. The cases of Choquet-like integrals and Sugeno-like integrals has been discussed. There the choice of fuzzy measure that determines particular integral and it is closely related to fuzzy partition of the input domain is crucial and deserves a deeper study.

Acknowledgement I would like to express my gratefulness to Radko Mesiar and Susanne Saminger for the fruitful discussions over the topic in concern.

Moreover, I acknowledge partial support of the grant 1M0572 of the MŠMT ČR.

References

[1] Calvo T., Mayor G., Mesiar R. (2002) Aggregation Operators: New Trend and Applications. *Studies in Fuzziness and Soft Computing. 97*, Physica Verlag, Heidelberg.

[2] F. Klawonn, V. Novák (1996) The relation between inference and interpolation in the framework of fuzzy systems. *Fuzzy Sets and Systems*, Vol. 81, 331–354.

[3] E. P. Klement, R. Mesiar, and E. Pap. (2000) Triangular Norms, In *Trends in Logic*, Vol. 8, Kluwer, Dordrecht.

[4] R. Mesiar (1995) Choquet-like integrals. *J. Math. Anal. Appl.*, Vol. 194, 477–488.

[5] E. Pap, Pseudo-additive measures and their applications, in *Handbook of Measure Theory (E. Pap, ed.)*, Elsevier Science B.V., 2002.

[6] Perfilieva, I.(2001). Normal forms for fuzzy logic functions and their approximation ability. *Fuzzy Sets and Systems*, Vol. 124, 371–384.

[7] Irina Perfilieva. Fuzzy transforms. In (J. F. Peters and A. Skowron Ed.) *Transactions on Rough Sets II. Rough Sets and Fuzzy Sets*, LNCS 3135, 63–81.

[8] H. Ruspini, E. (1969) A new approach to clustering. *Inform. and Control*, Vol. 15, 22–32.

[9] V. Shilkret (1971) Maxitive measures and integration. *Indag. Math.*, Vol. 33, 109–116.

[10] M. Sugeno (1974) *Theory of fuzzy integrals and its applications*. PhD thesis, Tokyo Institute of Technology, Tokyo.

[11] S. Weber (1986) Two integrals and some modified versions: critical remarks. *Fuzzy Sets and Systems*, Vol. 20, 97—105.

Comparison of Data-Driven Fuzzy Modelling Methods tested on NOx Data

Martin Štěpnička[1], Edwin Lughofer[2] and Viktor Pavliska[1]
Institute for Research and Applications of Fuzzy Modeling[1]
University of Ostrava
30. dubna 22, 701 03 Ostrava 1, Czech Republic
martin.stepnicka@osu.cz, viktor.pavliska@osu.cz
Fuzzy Logic Laboratorium Linz-Hagenberg[2]
Johannes Kepler University Linz
Altenbergerstrasse 69, A-4040 Linz, Austria
edwin.lughofer@jku.at

ABSTRACT:

This report is an application-experiment paper based on experimenting with real (**NOx**) data provided by Fuzzy Logic Laboratorium Linz-Hagenberg ($FLLL$) - Johannes Kepler University in Linz. The NOx data which are described below have been studied and functional dependencies between them successfully modelled by several fuzzy models identified by data-driven methods by the $FLLL$ institute; some particular result can be found e.g. in [2]. Bilateral project Aktion 41p19 between $IRAFM$-OU and $FLLL$-JKU made possible to realize a deeper cooperation between both institutes. One of key issues of the proposed cooperation was (for $IRAFM$) to benefit from experiences based on many applications and industrial projects solved by $FLLL$. Second part of the issue was (for $FLLL$) to benefit from theoretical research and techniques developed in $IRAFM$ and implemented in the software package LFLC2000. Based on the cited project, it was possible to obtain a real data and to make lots of experiments which enriched $IRAFM$ by experiences and prompted several improvements in the techniques developed in the institute as well as changes and implementations and the software package. $FLLL$ institute which cooperated on the project will be provided with all methods used in the experiments and experiment results.

1 Data Description

Let us shortly describe a problem yielding the NOx data with a brief description of an on-line approach used in a solution realized by *FLLL*.

At an engine test bench the task was to identify a k step ahead prediction model for the emission channel NOx directly from online measurement data. The task emerged from two purposes: first to be able to have an early detection of accidents and faults in the whole emission cycle, e.g. a broken pipe which can get even dangerous for the test bench operators, and second to save expenses on a measurement sensor for NOx later on. The later aspect is due to the fact that the obtained prediction model can be used for calculating the value of NOx out of some other measured channels. Compared to a fault detection based on static models, with a k step ahead prediction models upcoming events can be earlier recognized and hence faults or even accidents prevented.

Originally, the input data matrix consisting of 6700 samples which were recorded with a certain frequency. This frequency was too high in order to obtain feasible time delays of the original channels, as shifts up to 100 steps had to be carried out producing 1600 additional channels out of the 16 original channels (for each channel 100 different shifts: $k-1, k-2, ..., k-100$) in order to get a good approximation for NOx. Thus, it turned out that a simple down-sampling by taking just each 10th point and throwing away all the others yielded a sufficient resolution. Hence, the input matrix was reduced to 670 sample, where then a time shift up to 10 was sufficient causing a manageable amount of 160 channels and finally 660 samples (due to this shift the first ten samples needed to be cut out). It is obvious that a delay in the new data matrix of $k-l$ belongs to a delay of $k-10l$ in the original

one and vice versa. With the knowledge about the chosen frequency for sampling, namely 10 Hz per second, we can conclude to the real absolute delay for the impact of the input channels on NOx: 4 to 6 seconds. After applying variable selection it turned out, that at least four inputs (some original channels and their time delays) were needed in order to obtain an approximation quality higher than 0.9. The input channels for approximating NOx at time instant k consisted of the following list of channels (in the order they were selected): N = Engine Rotation Speed, P2offset = Pressure in Cylinder number 2, Te = Engine Output Torque, Nd = Speed of the Dynanometer together with their appropriate delays yielding a dynamic model in form of a four-step-ahead prediction of NOx

$$\begin{aligned} NOx(k) =& f(\mathrm{N}(k-4), \mathrm{P2offset}(k-5), \mathrm{Te}(k-5), \\ & \mathrm{Nd}(k-6), \mathrm{N}(k-6)) \end{aligned}$$

where one step back denotes exactly one second. In this sense $\mathrm{Te}(k-5)$ denotes linguistically 'the engine output torque five seconds ago', $\mathrm{P2offset}(k-5)$ denotes 'Pressure in Cylinder number 2 five seconds ago', $\mathrm{N}(k-4)$ denotes 'Engine speed five seconds ago' and so on. In this sense a four seconds ahead prediction model is yielded.

2 Original Results

Let us briefly recall some original results reached by *FLLL* which should serve us as an exemplar we would like to approach. Obviously, one could hardly expect that by general methods we can reach results of the same quality and therefore we say, that the original results serve us as an exemplar. However, from further sections it will be obvious that we do not obtain results of the same quality in all aspects but we obtain results of the same quality in some aspects, for instance we get the same accuracy of the model while we use more fuzzy rules or viceversa.

In Table 1 the model qualities as well as the model complexities are demonstrated when taking the five input variables stated above. The qualities were measured by a normalized average percent error:

$$APE\_norm = \frac{1}{N} \sum_{i=1}^{N} \frac{|\hat{y}_i - y_i|}{\max y - \min y} \qquad (1)$$

From this table it is obvious, that all methods performed similar, except *ANFIS*, which produced a quite high overfitting by generating all fuzzy set combination into rules (32 in this example). It has to be noticed that FLEXFIS-MOD is an incremental variant of fuzzy system modelling and hence applicable for online processes (e.g. online fault detection), where the models should be kept up-to-date as fast as possible (it processes point per point through its algorithm). When reducing the input dimensionality to 4 respectively 3, a drop of the normalized *APE* from 4.90% to 5.04% respectively 5.64% could be observed when applying *genfis2 ext2 (VQ-INC-MOD)* and the same number of rules.

Concluding, based on these fuzzy models (with an expected deviation smaller than 5% on fresh data), it is possible to detect faults with an intensity of approximately 2 times 5% = 10% or more in newly recorded measurements.

3 Newly Used Data-Driven Methods

This section is devoted to a short description of newly used methods implemented in the software package *LFLC2000*, see [1].

First of all, we succinctly introduce inference methods and learning algorithm implemented in the program and used for modelling the NOx prediction since not all the techniques in the package are appropriate for this purpose.

3.1 Inference Techniques and defuzzifications

Fuzzy approximation with conjunctions (FAC) is in fact the well known Mamdani-Assilian [3] approach since the mathematical interpretation of the fuzzy rule base of n rules is given by the following fuzzy relation R

$$R(x,y) = \bigvee_{i=1}^{n} (A_i(x) \wedge B_i(y)),\ x \in X\, y \in Y \qquad (2)$$

where A_i and B_i is the i-th antecedent and consequent fuzzy set, respectively.

Fuzzy approximation with implications (FAI) is also well known but for ceratin reasons

much less used in applications. It is based on the mathematical interpretation of the fuzzy rule base of n rules is given by the following fuzzy relation R

$$R(x,y) = \bigwedge_{i=1}^{n} (A_i(x) \to B_i(y)),\ x \in X\, y \in Y \quad (3)$$

where A_i and B_i is the i-th antecedent and consequent fuzzy set, respectively. In the *LFLC2000*, the *Łukasiewicz* residuation operation is used for an interpretation of the implication between antecedents and consequents represented by $\to$ in (3).

Concerning the inference technique, it is realized by the well known computational rule of inference (CRI) proposed by Lotfi A. Zadeh, see [11]. It is obvious that for a crisp input x' and a singleton fuzzifier the CRI gives an equivalent result to the simple evaluation of the fuzzy rule base interpretation at the node x' i.e. the inference is equal to $R(x', y) \subsetneq Y$ where R is given either by (2) or by (3).

For every single one approach a different defuzzification method has to be used. For the first approach based on conjunctive interpretation of rules, the *center of gravity* (COG) is used and for the second one based on implicative interpretation rules *mean of maxima* (MOM) is used.

3.2 Learning

For this paper, we consider the word *learning* in a wider sense i.e. every single method leading to an automatic generation of a fuzzy rule base will be considered to be a learning so the word is not understood necessarily on a neural point of view.

Expertly based linguistic approaches like *Perception based logical deduction* [4] have the advantage they the fuzzy rule base can be built expertly. Since the problem of prediction NOx values is nothing else but an approximation of the data and there is no expert knowledge of the system some learning method has to be used.

LFLC2000 has been equipped with the so called *linguistic learning* [1] for an automatic generation of a linguistic fuzzy rule base appropriate for the perception based logical deduction. The learning can be described as follows: If new measured input/output pair (x_j, y_j) comes find the most appropriate linguistic fuzzy sets (A_j, B_j) where $A_j \subsetneq X$ and $B_j \subsetneq Y$ and create a rule

$$\text{IF } x \text{ is } \mathcal{A}_j \text{ THEN } y \text{ is } \mathcal{B}_j \quad (4)$$

where $\mathcal{A}_j$ and $\mathcal{B}_j$ are just evaluating linguistic expressions represented by fuzzy sets A_j and B_j, respectively. In this way, huge rule base is created and then duplicate rules erased as well as complexity, inconsistency (conflicts in rules) and redundancy by sophisticated algorithms solved.

For the both fuzzy approximation methods, no learning has been implemented yet and the real NOx project demanded its development and implementation as a part of the Aktion project.

For this stage of investigation, a learning based on the same principle as the linguistic one was implemented. Compared to the linguistic one, it does not need a consistency and redundancy analysis there are no fully overlapping fuzzy sets in the fuzzy rule base (usually uniform triangles). On the other, as well as other usual axis based approaches it suffers from the curse of dimensionality.

3.3 Fuzzy Transform

Another fuzzy modelling method which was employed in this experiment is the fuzzy transform (F-transform) [6]. It is a fuzzy approximation method (approximating a functional dependency i.e. a continuous function $f : X \to Y$) based on two transforms - a direct one and an inverse one. It deals with a fuzzy partition of the domain X given by fuzzy sets called *basis functions* $A_i \subsetneq X\ i = 1, \ldots, n$ fulfilling several conditions including the Ruspini condition [7]

$$\sum_{i=1}^{n} A_i(x) = 1 \quad \forall x \in X. \quad (5)$$

Usually, the technique deals with triangular shaped fuzzy sets or sinusoidal shaped fuzzy sets. For details see [5] or [6].

The *direct F-transform* is a discrete simplified representation of the function f given by a real vector $[F_1, \ldots, F_n]$ where

$$F_i = \frac{\int_X f(x)A_i(x)dx}{\int_X A_i(x)dx} \quad (6)$$

and if the function is given only at (measured) samples $(x_j, f(x_j))\ j = 1, \ldots, m$ where $m >> n$, in

principle, then

$$F_i = \frac{\sum_{j=1}^{m} f(x_j)A_i(x_j)}{\sum_{j=1}^{m} A_i(x_j)dx}. \quad (7)$$

The *inverse F-transform* is again a continuous function on X and it is given by a linear combination of the basis functions and the components F_i of the direct F-transform i.e.

$$f_n^F(x) = \sum_{i=1}^{n} F_i A_i(x). \quad (8)$$

In the terminology used in the previous subsection, we can say that the inverse F-transform is an inference method (belonging to singleton models, close to T-S rules of the 0th order) while the direct F-transform is its learning algorithm.

The recalled approximation method can be easily generalized for functions with more variables, see [9, 10]. In the *LFLC2000* there is such a method implemented for an arbitrary number of variables, in principle.

4 FAI and FAC Results

Based on the NOx data consisting of a file of training samples and a file of $N = 159$ testing samples provided by the *FLLL* we have tested the *LFLC2000* techniques described above. To have a relevant comparison we have measured the accuracy by the correlation coefficient and by the normalized average percent error (1) as well as in the original case. Moreover, since we model a process of prediction NOx during an engine activity a speed of a chosen inference method including its defuzzification method is of a high interest.

In Table 2 there is an overview of some chosen results reached by the methods implemented in the *LFLC2000* software. By methods FAI and FAC we do not mean only fuzzy approximation with implications and conjunctions, respectively as inference methods but the whole data-driven method in the software i.e. fuzzy approximation learning, deleting duplicate rules, respective defuzzifiaction method (COG for FAC and MOM for FAI). Number of fuzzy sets expresses how many uniform triangular fuzzy sets on each axis have been used for generating the rule base. The speed of inference expresses time needed for getting all output values for 159 inputs from testing samples. This information is very rough and imprecise since always depends on used hardware, installed operational system and running software (in our case Intel® Pentium® IV 1.7GHz, 512MB SDRAM, Win XP Professional).

Obviously, it can be stated that the result do not reach such accurate values as in the original case but tend to them. Unfortunately, only if the number of rules increases rapidly.

5 Improvements

Clearly, the biggest problem is hidden in the curse of dimensionality since no fuzzy cluster analysis is used. To decrease the number of rules while keeping the given methods since their interpretability is very high, if we understand it as a possibility to interpret, say linguistically, the model or its unique rules, a new algorithm has to be implemented. Of course, in case of a high number of rules, a possibility of understanding them decreases and moreover, speed of an inference method including its respective defuzzification method decreases as well. The following algorithm has been implemented.

Let on ith axis K triangular fuzzy sets A_k^i $k = 1, \dots K$ are created for $i = 1, \dots, I$. Let a fuzzy approximation rule base is generated by the introduced algorithm. If there are two rules of the form

$$\text{IF } \cdots x_i \text{ is } \mathcal{A}_{k_i}^i \cdots \text{ THEN } y \text{ is } \mathcal{B}, \quad (9)$$

$$\text{IF } \cdots x_i \text{ is } \mathcal{A}_{k_i+1}^i \dots \text{ THEN } y \text{ is } \mathcal{B} \quad (10)$$

i.e. consequent fuzzy sets and antecedent fuzzy sets excepting one ith antecedent fuzzy set are equal and the $i$the antecedent fuzzy sets are neighboring half-overlapping fuzzy sets then the rules are merged to one

$$\text{IF } \cdots x_i \text{ is } \mathcal{T}_{k_i\_k_i+1}^i \cdots \text{ THEN } y \text{ is } \mathcal{B} \quad (11)$$

where the linguistic expression $\mathcal{T}_{k_i\_k_i+1}^i$ is represented by a trapezoidal fuzzy set

$$T_{k_i\_k_i+1}^i(x) = A_{k_i+1}^i(x) \oplus A_{k_i+1}^i(x) \quad (12)$$

and where $\oplus$ is the Łukasiewicz t-conorm.

Similarly, if there is (after several merging) a rule, which on some antecedent axis is totally overlapped by some trapezoidal fuzzy set from another

rule and fuzzy sets on the other axes are equal in both rules, then this rule is erased. In fact, instead of clustering data before an automatic generation of rules, this algorithm clusters already generated rules.

In Table 3, there is a short overview of some chosen results.

6 F-transform Results

Finally, the F-transform method was tested on NOx data as well. This method suffers from the curse of dimensionality a bit less then the previous fuzzy approximation techniques although it simply creates all possible combination of fuzzy subdomains and searches for their discrete representatives. The advantage is that the inverse F-transform is computationally extremely simple and fast. It is just a linear combination of fuzzy sets and there are no time requirements for a defuzzification. The difference can be seen in the speed columns of Tables 2, 3 and 4.

So, the curse of dimensionality does not cause a computational complexity growth and inference speed problems. Interpretability is very natural since the F-transform method in fact claims that

$$\text{IF } x_1 \text{ is } \mathcal{A}^1_{k_1} \text{ and ... and } x_i \text{ is } \mathcal{A}^i_{k_i} \text{ and ...}$$
$$\text{... and} x_I \text{ is } \mathcal{A}^I_{k_I} \text{ THEN } y \text{ is } F_i.$$

Although one can hardly expect that from a huge number of, say "rules", a human can understand relationships between variables, the F-transform with its interpretability can easily provide a user with a local type of information i.e. what happens on some chosen subdomains.

7 Comparison, Conclusions and Further Work

All in all, it can be stated that the *LFLC2000* methods did not bring as good results as the original approach [2] in all monitored features (No. of rules, accuracy, speed etc.) at ones. On the other hand, in some concrete features the results can be at least as good if we allow e.g. higher number of rules. Especially the F-transform method can be significantly useful in getting high accuracy and speed properties.

The fuzzy approximation methods *FAI* and *FAC* has been demonstrated to be useful universal methods which are especially effective in lower number of input variables. Otherwise, they suffer from curse of dimensionality some cluster analysis has to be used to improve their results. Within the Aktion project, some promising improvements based on the rule merging algorithm have been reached.

The F-transform algorithm featured by extremely high inference speed and accuracy below 5% available. Its high number of fuzzy set combinations (curse of dimensionality) did not influence the speed. The interpretability is natural as well but the transparency of such model composed of hundreds or even more combinations is questionable.

Another key issue of the data-driven model is hidden in incremental vs. batch type of learning (also on-line vs. off-line). Every single method in the *LFLC2000* is implemented for an off-line learning applications and therefore all results in the tables in the previous sections are results of batch models. On the other hand, any of them can be in principle modified to an incremental one without a significant influence.

Fuzzy approximation models *FAI* and *FAC* do learn from individual incoming data and step by step generate individual rules. The suggested rule merging algorithm can be used in on-line merging as well without any change.

In case of the F-transform, it is a bit more complicated since formula (7) is a typical batch formula. On the other hand, if the partial summations are kept in a memory of the learning machine, the coefficients F_i can be on-line modified. Moreover, an on-line neural approach to the F-transform based on the gradient descent method was already suggested, see [8]. This approach unfortunately has not been tested on NOx data since it is programmed only for single-input-single-output problems yet. On the other, we consider it to be a promising method which should be tested within the on-going work since it also allows us to adapt the *basis functions* which increases accuracy and decreases number of fuzzy sets.

Acknowledgement

This investigation has been supported by projects Aktion 41p19 and DAR 1M0572 of the MŠMT ČR.

References

[1] Dvořák A., Habiballa H., Novák V. and Pavliska V. (2003), The Software Package LFLF2000 - Its Specificity, Recent and Perspective Applications. *Computers in Industry*, 51, 2003, 269–280.

[2] Lughofer E. (2005), Data-Driven Incremental Learning of Takagi-Sugeno Fuzzy Models, PhD-Thesis, Department of Knowledge-Based Mathematical Systems, University Linz, Austria.

[3] Mamdani E. and Assilian S. (1975), An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, 7, 1975, 1–13.

[4] Novák V. (2005), Perception-Based Logical Deduction as Alternative Approximate Reasoning Method. *Proc. of FUZZ-IEEE '05*, Reno, Nevada, USA, 1032–1037.

[5] Perfilieva I. (2003) Fuzzy Approach to Solution of Differential Equations with Imprecise Data: Application to Reef Growth Problem. In: *Fuzzy Logic in Geology* (R.V. Demicco and G.J. Klir, Eds.), Academic Press, Amsterdam, Chap. 9, 275–300.

[6] Perfilieva I. (2006), Fuzzy Transforms. *Fuzzy Sets and Systems*, in press.

[7] Ruspini E. H. (1969), A New Approach to Clustering. *Inform. and Control*, 15, 22–32.

[8] Štěpnička M. and Polakovič O. (2006), The Fuzzy Transform from the Neural Network Point of View. *Proc. IPMU 2006*, submitted.

[9] Štěpnička M. and Valášek R. (2004), Fuzzy Transforms and Their Applications to Heat Flow Equation. *Abstracts of the 2nd FLLL/SCCH Master and PhD Seminar 2004*, 51–56.

[10] Štěpnicka M. and Valášek R. (2005), Numerical Solution of Partial Differential Equations with Help of Fuzzy Transform. *Proc. of FUZZ-IEEE '05*, Reno, Nevada, USA, 1104–1109.

[11] Zadeh L. A. (1973), Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Trans. Systems, Man and Cybernet.* **SMC-3**, 28–44.

Method	**APE norm 5 features / Av. No. Fuzzy Sets / No. of Rules**
ANFIS	5.25% / 2 / 32
genfis2	4.83% / 5 / 5
genfis2 loc.	4.86% / 5 / 5
genfis2 ext. (VQ-INC)	4.92% / 4 / 4
genfis2 ext2 (VQ-INC-MOD)	4.90% / 4 / 4
FLEXFIS-MOD sam. (inc.)	4.98% / 5 / 5

Table 1: Comparison of data-driven modelling methods for fuzzy systems based on NOx data

Var.	Method	No. of F. Sets	No. of Rules	APE norm	Correlation	Speed
3	*FAI*	5	83	9.25%	0.595	5.56 s
3	*FAC*	5	83	9.33%	0.734	2.88 s
3	*FAI*	7	182	8.59%	0.666	10.60 s
3	*FAC*	7	182	6.57%	0.849	4.16 s
3	*FAI*	10	362	7.49%	0.751	18.50 s
3	*FAC*	10	362	5.96%	0.859	5.61 s
4	*FAI*	7	323	6.01%	0.862	20.00 s
4	*FAC*	7	323	5.72%	0.875	7.77 s
5	*FAI*	5	181	8.18%	0.731	14.60 s
5	*FAC*	5	181	7.74%	0.842	7.56 s
5	*FAI*	7	350	6.10%	0.863	22.50 s
5	*FAC*	7	350	5.81%	0.879	8.66 s

Table 2: Some chosen results by *LFLC2000*.

Var.	Method	No. of Fuzzy Sets	No. of Rules	APE norm	Correlation	Speed
3	*FAI*	5	6	10.85%	0.585	0.33 s
3	*FAC*	5	6	10.85%	0.745	0.30 s
3	*FAI*	7	18	11.27%	0.503	0.41 s
3	*FAC*	7	18	7.87%	0.830	0.48 s
5	*FAI*	5	18	9.76%	0.631	0.97 s
5	*FAC*	5	18	7.47%	0.842	0.89 s
5	*FAI*	7	64	8.17%	0.719	2.34 s
5	*FAC*	7	64	7.55%	0.802	1.75 s

Table 3: Some chosen results by *LFLC2000* after merging rules.

Variables	F. Set Shape	No. of Fuzzy Sets	APE norm	Correlation	Speed
3	*triang.*	2	10.59%	0.824	0.05 s
3	*sinus.*	2	9.65%	0.869	0.05 s
3	*triang.*	5	5.70%	0.891	0.06 s
3	*sinus.*	5	5.55%	0.879	0.03 s
4	*triang.*	3	7.23%	0.875	0.03 s
4	*sinus.*	3	6.88%	0.876	0.05 s
4	*triang.*	5	4.77%	0.925	0.05 s
4	*sinus.*	5	4.91%	0.904	0.05 s
5	*triang.*	2	9.59%	0.824	0.05 s
5	*sinus.*	2	8.45%	0.829	0.06 s
5	*triang.*	4	6.10%	0.863	0.06 s
5	*sinus.*	4	5.23%	0.897	0.06 s
5	*triang.*	6	4.69%	0.908	0.06 s
5	*sinus.*	6	4.62%	0.912	0.06 s

Table 4: Some chosen results by the F-transform method in *LFLC2000*.

Po-group representations of MTL-algebras – how far can we get?

Thomas Vetterlein
Department of Computer Sciences 1, University of Dortmund
44221 Dortmund, Germany
Thomas.Vetterlein@uni-dortmund.de

March 2006

Abstract

We show, step by step, how MTL-algebras and in particular left-continuous t-norm algebras may be represented by means of partially ordered groups. We will see that for the cases typically appearing in applications, the methods works. However, a further example shows that we cannot treat every MTL-algebra in the described manner.

1 Introduction

Fuzzy logics, as understood by us, are many-valued logics whose formulae are modelled by values from the real unit interval $[0,1]$, whose conjunction is interpreted by some t-norm $\odot\colon\ [0,1]^2 \to [0,1]$ and whose implication is interpreted by the operation $\to\colon\ [0,1]^2 \to [0,1]$ given by $a \to b = \max\{x : a \odot x \leq b\}$. Not for all t-norms, $\to$ is definable as indicated; a necessary and sufficient condition is the left-continuity of $\odot$. So the weakest fuzzy logic is based on left-continuous t-norms and the corresponding residua, as it is the case for Esteva and Godo's MTL [EsGo].

We are here concerned with t-norm algebras $([0,1]; \odot, \to, 0, 1)$, where $\odot$ is a left-continuous t-norm and $\to$ is its residuum. Efforts to characterise the structure of these algebras have not been successful for several years.

Recall that an MTL-algebra is a structure $(L; \wedge, \vee, \odot, \to, 0, 1)$ such that (M1) $(L; \wedge, \vee, 0, 1)$ is a bounded lattice, (M2) $(L; \odot, 1)$ is a commutative monoid, (M3) $\odot$ is isotone in both variables, (M4) for any a, b, $a \to b$ is maximal among all elements x such that $a \odot x \leq b$, and (M5) for any a, b, $(a \to b) \vee (b \to a) = 1$. It has been shown in [JeMo] that the variety generated by left-continuous t-norm algebras consists exactly of the MTL-algebras. So left-continuous t-norm algebras are MTL-algebras, and any MTL-algebra is a homomorphic image of a subdirect product of left-continuous t-norm algebras. It follows that to analyse MTL-algebras is a problem as difficult as to analyse left-continuous t-norms. We will deal here with (a variant of) MTL-algebras, but our examples will be t-norm algebras.

2 Basic algebraic notions related to MTL

We shall not deal directly with MTL-algebras; for our convenience, we will slightly change this notion.

First of all, we are interested only in totally ordered MTL-algebras. Moreover, since $a \leq b$ holds in MTL-algebras exactly if $a \to b = 1$, the assumption that the order is total makes the prelinearity condition (M5) superfluous.

Second, without the presence of (M5), the implication $\to$ appears only in (M4) for its definition; (M4) just says that $\to$ is well-defined. So we do not loose information by dropping $\to$ and replacing (M4) by the following axiom: (M4') For any a, b, among all elements x such that $a \odot x \leq b$, there is a maximal one.

Third, to assume the existence of a smallest element turns out to be not helpful, but rather complicates matters. A *basic semihoop* [EGHM] is a structure $(L; \leq, \odot, \to, 1)$ such that (M1') $(L; \wedge, \vee, 1)$ is an upper-bounded lattice, and conditions (M2)–(M5) hold. MTL-algebras are basic semihoops which possess a smallest element. Taking into account the preceeding two paragraphs, we consider here structures $(L; \leq, \odot, 1)$ such that (M1") $(L; \leq, 1)$ is an upper-bounded totally ordered set, and (M2), (M3), (M4') hold.

The forth and last step is of purely technical nature, done in order to support our intuition of what follows. Namely, we shall reverse the order. The multiplication-like operation $\odot$ becomes an addition-like $\oplus$, and the constant 1 will be renamed to 0. We arrive at our main definition.

Definition 2.1 A structure $(L; \leq, \oplus, 0)$ is called a *dbs-chain* if the following holds.

(T1) $(L; \leq, 0)$ is a totally ordered set with the smallest element 0.

(T2) $(L; \oplus, 0)$ is a commutative semigroup with neutral element 0.

(T3) $\oplus$ is isotone in both variables, that is, for any $a, b, c \in L$, $a \leq b$ implies $a \oplus c \leq b \oplus c$.

(T4) For any a and b, there is a smallest element x such that $a \oplus x \geq b$.

To illustrate this notion, consider the product t-norm algebra. Namely, let $\odot$ be the usual multiplication of reals, let $a \to b = \frac{b}{a}$ if $a \leq b$ and else $= 1$. Then $([0,1]; \odot, \to, 0, 1)$ is an MTL-algebra, so in particular $([0,1]; \odot, \to, 0)$ is a basic semihoop. Now, $\leq$ is a obviously a total order, and $\to$ may be dropped since it is definable from $\odot$ and $\leq$; we arrive at $([0,1]; \odot, 1)$. Reversing the order finally leads to the dbs-chain $([0,1]; \oplus, 0)$, where $a \oplus b = a + b - ab$.

By dropping the element 1, we also get an example of a dbs-chain which does not originate from an MTL-algebra; $([0,1); \oplus, 0)$ fulfils all axioms (T1)–(T4). This is a consequence of the special situation that $[0,1)$ is closed under the operation $\oplus$.

3 Partially ordered groups

The representation of MTL-algebras will be based on partially ordered groups.

Definition 3.1 A structure $(L; \leq, +, 0)$ is called a *totally ordered abelian group*, or *to-group* for short, if the following holds.

(G1) $(L; \leq)$ is a totally ordered set.

(G2) $(L; +, 0)$ is an abelian group.

(G3) $+$ is isotone in both variables, that is, for any $a, b, c \in L$, $a \leq b$ implies $a + c \leq b + c$.

The reals provide an easy example of a to-group. Namely, $(\mathbb{R}; \leq, +, 0)$ is a to-group, where $\leq$ is the natural order and $+$ the usual addition.

We shall describe the structure of to-groups, which is known in detail. Let $(I; \leq)$ be a total order, and for every $\iota \in I$, let R_ι be a copy of the reals. Let

$$\Gamma_{\iota \in I} R_\iota = \{(a_\iota)_{\iota \in I} \in \Pi_{\iota \in I} R_\iota \colon \{\iota \in I \colon a_\iota \neq 0\} \text{ is a well-ordered subset of } I\}.$$

Recall that a totally ordered set J is called well-orderd if every subset of J possesses a smallest element.

We define the addition $+$ on $\Gamma_{\iota \in I} R_\iota$ componentwise. Moreover, we endow $\Gamma_{\iota \in I} R_\iota$ with a total order as follows. For a non-zero $(a_\iota)_\iota \in \Gamma_{\iota \in I} R_\iota$, let ι_a be the smallest $\iota \in I$ such that $a_\iota \neq 0$. Then we let $(a_\iota)_\iota > 0$ if $a_{\iota_a} > 0$.

$\Gamma_{\iota \in I} R_\iota$ is called the *lexicographical product* of the R_ι. Now, the Hahn embedding theorem [Fuc] says that any to-group is a subgroup of some lexicographical product of groups $(\mathbb{R}; \leq, +, 0)$.

4 Partial algebras derived from total ones

We next describe how we relate MTL-algebras to partially ordered groups, or, to be more precise, how we relate dbs-chains to to-groups. This idea has been developed in [Vet1, Vet2].

Note that dbs-chains and positive cones of po-groups do not differ fundamentally according to their definition. The basic difference is cancellativity: Whereas in the case of po-groups, $a + c = b + c$ implies $a = b$, this is in general not true for dbs-chains. The question is if we cannot derive from the non-cancellative addition $\oplus$ on which dbs-chains are based, a cancellative one. We can – but it will no longer be a total operation.

Definition 4.1 Let $(L; \leq, \oplus, 0)$ be a dbs-chain. Let $+$ be the partial binary operation specified as follows. For $a, b \in L$, let $a + b = a \oplus b$ if a is the smallest element x such that $x \oplus b = a \oplus b$ and b is the smallest element y such that $a \oplus y = a \oplus b$. Else let $a + b$ be undefined. Then $(L; \leq, +, 0)$ is called the *partial algebra associated to* L.

This definition makes sense because the transition from the total $\oplus$ to the partial $+$ does not mean any loss of information. Namely, from $(L; \leq, +, 0)$

we recover the operation $\oplus$ underlying a dbs-chain by

$$a \oplus b = \max \{a' + b' \colon\ a' \leq a \text{ and } b' \leq b \text{ such that } a' + b' \text{ is defined}\}.$$

Our question is: Can we isomorphically embed the partial algebra $(L; \leq, +, 0)$ associated to a dbs-chain into a to-group $(G; \leq, +, 0)$?

Definition 4.2 Let $(L; \leq, \oplus, 0)$ be a dbs-chain. $(L; \leq, +, 0)$ be the partial algebra associated to L. Then we call L *representable in the to-group* $(G; \leq, +, 0)$ if there exists a mapping $\varphi \colon\ L \to G$ such that for all $a, b, c \in L$, (i) $a \leq b$ iff $\varphi(a) \leq \varphi(b)$, (ii) $\varphi(a + b) = \varphi(a) + \varphi(b)$ if $a + b$ is defined, and (iii) $\varphi(0) = 0$.

We know that in general, dbs-chains are not representable in a to-group. But we also know that in practically all cases of left-continuous t-norms which have been found, such a representation does exist.

5 Po-group representations of known left-continuous t-norm algebras

We shall show that the t-norm algebras based on two well-known examples of left-continuous t-norms, are representable in a to-group.

First, let $\odot_L$ be the Łukasiewicz t-norm, that is,

$$a \odot_L b = (a + b - 1) \vee 0$$

for $a, b \in [0, 1]$. Then the corresponding dbs-chain is $([0, 1]; \leq, \oplus_L, 0)$, where

$$a \oplus_L b = (a + b) \wedge 1.$$

Furthermore, as it is easily verified, the partial algebra associated to L is $([0, 1]; \leq, +_L, 0)$, where

$$a +_L b = \begin{cases} a + b & \text{if } a + b \leq 1, \\ \text{undefined} & \text{else.} \end{cases}$$

The to-group representation is straightforward. Take the to-group $(\mathbb{R}; \leq, +, 0)$, and let $\varphi \colon\ [0, 1] \to \mathbb{R}$ be the identity.

The next example is a non-continuous left-continuous t-norm. Let $\odot_{rP}$ the rotated product t-norm [Jen]; for $a, b \in [0,1]$, let

$$a \odot_{rP} b = \begin{cases} 2ab - a - b + 1 & \text{if } a, b > \frac{1}{2}, \\ \frac{a+b-1}{2a-1} & \text{if } a > \frac{1}{2},\ b \leq \frac{1}{2}, \text{ and } a + b > 1, \\ 0 & \text{if } a + b \leq 1. \end{cases}$$

The corresponding dbs-chain is in this case $([0,1]; \leq, \oplus_{rP}, 0)$, where

$$a \oplus_{rP} b = \begin{cases} a + b - 2ab & \text{if } a, b < \frac{1}{2}, \\ \frac{b-a}{1-2a} & \text{if } a < \frac{1}{2},\ b \geq \frac{1}{2}, \text{ and } a + b < 1, \\ 1 & \text{if } a + b \geq 1; \end{cases}$$

In this case, it is more tedious, but still routine to calculate the associated partial algebra $([0,1]; \leq, +_{rP}, 0)$; we have

$$a +_{rP} b = \begin{cases} a + b - 2ab & \text{if } a, b < \frac{1}{2}, \\ \frac{b-a}{1-2a} & \text{if } a < \frac{1}{2},\ b > \frac{1}{2}, \text{ and } a + b \leq 1, \\ 1 & \text{if } a = b = \frac{1}{2}, \\ \text{undefined} & \text{else.} \end{cases}$$

Furthermore, let now $R_0 = \mathbb{N}$ and $R_1 = \mathbb{R}$; let $R = R_0 \times_{\text{lex}} R_1$ be the lexicographical product of R_0 and R_1; and let

$$L = \{(a, b) \in R^+ :\ a = 0,\ b \geq 0 \text{ or } a = 1,\ b = 0 \text{ or } a = 2,\ b \leq 0\}.$$

Then $([0,1]; \leq, +_{rP}, 0)$ is representable in R, the embedding $\varphi :\ [0,1] \to L$ being given by

$$\varphi(a) = \begin{cases} (0, -\ln(1-2a)) & \text{if } a < \frac{1}{2}, \\ (1, 0) & \text{if } a = \frac{1}{2}, \\ (2, \ln(2a-1)) & \text{if } a > \frac{1}{2}. \end{cases}$$

We note that except for our aim of finding a po-group representation, we also achieved in the present case a very clear picture of the t-norm under consideration. For further examples, see [Vet2].

6 A dbs-chain not representable in a to-group

We conclude with one more example of a dbs-chain, taken from [EKMMW]. Let $L = \{0, 9, 12, 16, 18, 21, 24, 25, 27, 28, 32, 30\}$; endow L with the total order according to the way in which we enumerated the elements of L, which

in particular means that 0 is the zero and 30 the largest element; for $a, b \in L$, define $a \oplus b$ as the usual sum of natural numbers provided the result is in L, else let $a \oplus b = 30$. Then L is a dbs-chain. Moreover, the partial addition associated to $\oplus$ is the restriction to the pairs $a, b \in L$ such that $a + b$, i.e. the sum of a and b taken as natural numbers, is in L.

$(L; \leq, \oplus, 0)$ is not representable by any to-group. Assume to the contrary that $\varphi\colon\ L \to G$ is an embedding of L in some to-group $(G; \leq, +, 0)$. Then from $2 \cdot 12 < 9 + 16$ and $3 \cdot 9 < 12 + 16$ it follows $3 \cdot \varphi(9) + 2 \cdot \varphi(12) < \varphi(9) + \varphi(12) + 2 \cdot \varphi(16)$, that is, $2 \cdot \varphi(9) + \varphi(12) < 2\varphi(16) = \varphi(32)$, as opposed to $2 \cdot 9 + 12 > 32$.

7 Conclusion

We deloped a way to represent totally ordered MTL-algebras by means of totally ordered abelian groups. Whenever such a representation exists, the task to analyse the MTL-algebra is greatly simplified since we know the structure of to-groups.

Such a representation does not exist in general, but for an apparantly large class of MTL-algebras including those appearing in applications.

The question to be solved next is to find the exact, or at least reasonably weak, conditions for an MTL-algebra to be representable in a to-group.

References

[EsGo] F. Esteva, L. Godo, Monoidal t-norm based logic: Towards a logic for left-continuous t-norms, *Fuzzy Sets Syst.* **124** (2001), 271 - 288.

[EGHM] F. Esteva, L. Godo, P. Hájek, F. Montagna, Hoops and fuzzy logic, *J. Log. Comput.* **13** (2003), 531 - 555.

[EKMMW] K. Evans, M. Konikoff, J. J. Madden, R. Mathis, G. Whipple, Totally ordered commutative monoids, *Semigroup Forum* **62** (2001), 249 - 278.

[Fuc] L. Fuchs, "Partially ordered algebraic systems", Pergamon Press, Oxford 1963.

[Jen] S. Jenei, Structure of left-continuous triangular norms with strong induced negations. I: Rotation construction, *J. Appl. Non-Class. Log.* **10** (2000), 83 - 92.

[JeMo] S. Jenei, F. Montagna, A proof of standard completeness for Esteva and Godo's logic MTL, *Stud. Log.* **70** (2002), 183 - 192.

[Vet1] T. Vetterlein, Weak effect algebras, submitted.

[Vet2] T. Vetterlein, MTL-algebras arising from partially ordered groups, in: S. Gottwald, U. Höhle, P. Klement, "Fuzzy logics and related structures"; to appear.

Image fusion using fuzzy transform

Martina Daňková and Radek Valášek

April 25, 2006

University of Ostrava
Institute for Research and Application of Fuzzy Modeling
30.dubna 22, 701 03 Ostrava, Czech Republic

1 Introduction

Fuzzy transformation [1] is the powerful tool for approximation of continuous functions. Since it locally minimize weighted arithmetic mean it allows to extract different frequency. Due to this property, we decided to use it in the problematic of image fusion. But similarly as wavelet transformation [2] it might become useful in another applications.

It is necessary to extract different frequences in one turn, therefore, we have to apply fuzzy transform more than once. We found a formula using which we are able to express an original function using partial sums with an arbitrary precision. Whenever we work with a discrete data set, we can transform it completely and so, we will speak about a full fuzzy transformation. For the details see [4]. The aim of this contribution is to present an alternative approach to the solution of image fusion problem and its comparison to the one based on wavelet transform.

The main task is to find a formula using which we may express original function $f : X \mapsto Y$ using partial sums with an arbitrary precision, i.e.

$$f_T(x) = f_{T,1}(x) + f_{T,2}(x) + f_{T,3}(x) + \ldots = \sum_{i=0}^{\infty} f_{T,i}(x) = f(x). \qquad (1)$$

Then we may transform functions $f_1, \ldots, f_p$ into $f_{1_T}, \ldots, f_{p_T}$ and operate on each level $i \in \mathbb{N}$ with $f_{1_{T,i}}, \ldots, f_{p_{T,i}}$. Moreover a composition needs to be specified which says us how to fuse functions.

Let us denote a partial sum

$$S_n(x) = \sum_{i=0}^{n} f_{T,i}(x). \qquad (2)$$

Fixed fuzzy transform: Now we give the original definition of the F-transform taken from [1] for the 1-dimensional case. In the sequel, we assume that $X, Y \subseteq \mathbb{R}$.

Let $c_i = a + h(i-1)$ be nodes on X where $h = (b-a)/(k-1)$, $k \geq 2$ and $i \in I = \{1, \ldots, k\}$. We say that fuzzy sets $A_1, \ldots, A_k \subset_{\underset{\sim}{S}} X$ create *base* if each of them fulfils the following conditions:

- $A_i(c_i) = 1$,
- $A_i(x) = 0$ if $x \notin (c_{i-1}, c_{i+1})$ where $c_{-1} = a,\ c_{k+1} = b$,
- $A_i(x)$ is continuous,
- $A_i(x)$ strictly increases on $[c_{i-1}, c_i]$ and strictly decreases on $[c_i, c_{i+1}]$,
- $\sum_{i \in I} A_i(x) = 1$, for all $x \in X$

This base forms Ruspini's fuzzy partition introduced in [3], i.e. a fuzzy partition of a set X is defined as a finite family $\{A_i \subset_{\underset{\sim}{S}} X\}_{i \in I}$ such that $\sum_{i \in I} A_i(x) = 1$ for any $x \in X$.

Definition 1.1 *Let $F_1, \ldots, F_k$ be given by $F_i = \frac{\int_a^b f(x) A_i(x)\, dx}{\int_a^b A_i(x)\, dx}$. The function*

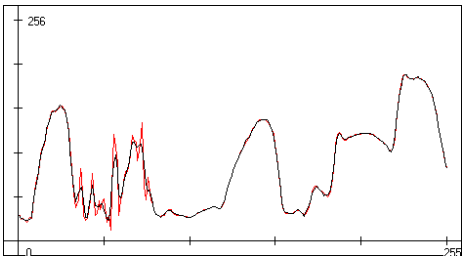
$$T_{f,k}(x) = \sum_{i \in I} F_i A_i(x) \tag{3}$$

will be called the fixed F-transform of f over the k nodes.

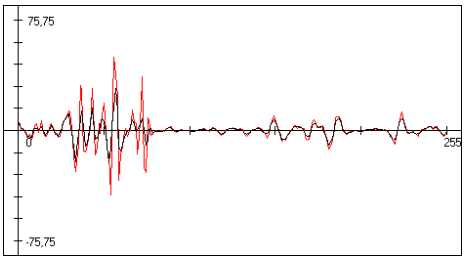
Coefficients F_i of the F-transform serve us as a discrete representation of values of f above supports of A_i's. In fact, we are averaging all the values above such intervals $[c_{i-1}, c_{i+1}]$ and these fuzzy sets A_i are used as weights in this averaging.

Full fuzzy transform and fusion settings:

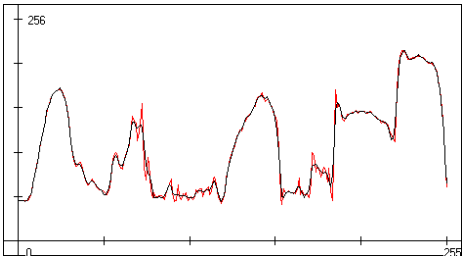
- $f_{T,0}(x)$ stands for arithmetic mean of $f(x)$ and error function $e_0 = f(x) - f_{T,0}(x)$.
- For $i \geq 1$,
$$f_{T,i}(x) = T_{e_{i-1}, 2^i},$$
represents fuzzy transform of e_{i-1} in 2^i nodes and
$$e_i(x) = e_{i-1}(x) - f_{T,i}(x).$$
- Fusion function operates over the coefficients of fuzzy transforms of $f_{1_{T,i}}, \ldots, f_{p_{T,i}}$ in each level and it is defined by e.g.
$$\kappa(x, y) = \begin{cases} y, & |x| \leq |y| \\ x, & \text{otherwise.} \end{cases}$$



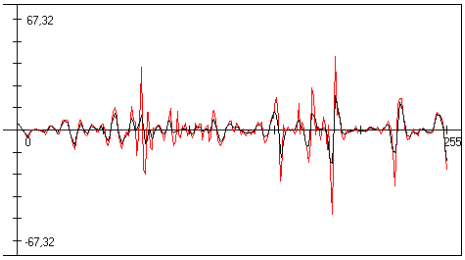
(a) Function f_1 and $S'_{128}(x)$



(b) $f_1 - S'_{128}(x)$ and its further approximation



(c) Function f_2 and $S''_{128}(x)$



(d) $f_2 - S''_{128}(x)$ and its further approximation

Figure 1: Functions to be fused and their approximations.

- Fused function is given by $F_T(x) = \bar{f}_{T,1}(x) + \bar{f}_{T,2}(x) + \bar{f}_{T,3}(x) + \ldots = \sum_{i=0}^{\infty} \bar{f}_{T,i}(x)$, where $\bar{f}_{T,i}(x) = \sum_{i \in I} F_i A_i(x)$ for each i and $F_1, \ldots, F_{2^i}$ are determined on the basis of coefficients of fixed fuzzy transformations $f_{1_{T,i}}, \ldots, f_{p_{T,i}}$ using κ.

Note that we may increase the number of nodes in which we create fixed fuzzy transform arbitrarily. Also the starting approximation can be taken as fixed fuzzy transformation of a higher level (number of nodes > 1). Moreover, there exist lots of different possibilities how to specify κ, see [5].

Example 1.2 *Let us assume two discrete functions $f_1, f_2 : I \mapsto [0,1]$ representing 191'th rows taken from different blurred images of Lenna.BMP.*

Figure 1 illustrates the fixed fuzzy transformations (black lines) of f_1 and f_2 (red lines) over the 128 *nodes and Figure 1 shows the fusion on the basis of κ (blue line) compared with the ideal image row (black line).*

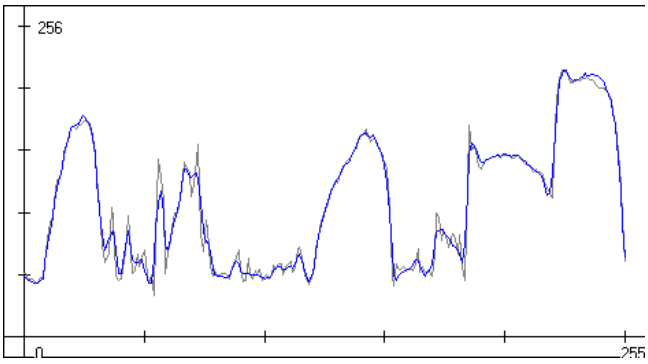


Figure 2: Fused function after 5 iterations.

Example 1.3 *Analogously as in the case of one input variable functions, we may apply the technique of full fuzzy transform to the images, i.e. functions of two variables, see Figure 3.*

2 Conclusions

The application shows that the full fuzzy transform is effective tool in the problematic of function fusion and it is worth to make a deeper study of its properties. In the near future, it is considered to be implemented into the bigger suit of filters and tools for image processing.

Acknowledgements This investigation has been partially supported by project 1M0572 of the MŠMT ČR.

References

[1] Perfilieva, I.(2004). Fuzzy transforms. In: *Transactions on Rough Sets II. Rough Sets and Fuzzy Sets.* (J. F. Peters and A. Skowron Ed.). LNCS 3135, pp. 63-81.

[2] Resnikoff, H.L., Wells, R.O.(1998) *Wavelet analysis : the scalable structure of information.* Springer - New York.

[3] Ruspini, E., H.(1969). A new approach to clustering. *Inform. and Control*, 15, pp. 22-32.

[4] Martina Daňková, Radek Valášek (2005) Fusion of functions using full fuzzy transform, Technical report No. 6, University of Ostrava, IRAFM.

[5] Chibani, Y., Houacine, A. (2003). *Redundant versus orthogonal wavelet decomposition for multisensor image fusion.* Vol. 36, Nr. 4, pp. 879-887.

(a) Image 1

(b) Image 2

(c) Fusion of image 1 and 2.

Figure 3: Example of image fusion.