Imprecise regression and regression on fuzzy data A preliminary discussion

Mathieu Serrurier, Henri Prade

Imprecise regression and regression on fuzzy data A preliminary discussion

Mathieu Serrurier
IRIT
University of Toulouse III
31062 Toulouse Cedex 9, France
E-mail: serrurie@irit.fr

Henri Prade
IRIT
University of Toulouse III
31062 Toulouse Cedex 9, France
E-mail: prade@irit.fr

Abstract— **The paper provides a discussion of the possibilistic regression method originally proposed by H. Tanaka. This method has the advantage of allowing the learning of an imprecise model, in the form of an interval-valued function. It may lead to an imprecise model even in presence of precise data, which is satisfactory from a learning point of view. Indeed, finding a precise model that perfectly represents the concept to be learned is illusory, due to the existence of the bias caused by the choice of a modeling representation space, the limited amount of data, and the possibility of missing relevant data. However, what is obtained with possibilistic regression is more an imprecise model than a genuine fuzzy one. The paper illustrates and emphasizes this point on environmental data and suggest two different approaches for learning genuine fuzzy regression models from precise data.**

I. Introduction

Machine learning aims at building models that describe concepts from data. If the concept to be learned consists in being able to assign one of a finite number of classes to an object, the problem is referred to classification. If the concept takes the form of a continuous function, we face a regression problem.

Classical regression methods provide a summarization of data under the form of precise, linear or not linear, functions. This approach has been extended to fuzzy data by adapting the least square fitting criterion to the fuzzy case [2], [3] (see also [12], [14], [1], [4] for overviews and general discussions). These approaches handles fuzzy inputs and/or fuzzy outputs.

Besides, a possibilistic regression method has been proposed [13], [9], [11], which aims at learning an imprecise or fuzzy model even when the data are non-fuzzy. In this way, the learning process does not force the building of a model that is too precise, and thus provides a better account of the data. This view fits quite well with the very nature of learning, namely having a precise model that exactly describes the data is illusory and may not be very efficient for prediction.

This paper first provides a discussion of the two types of fuzzy regression methods. Then, we focus on the learning of imprecise, or fuzzy models from precise data. We recall the approach proposed by H. Tanaka and his colleagues and illustrate it on an application on environmental data. It is pointed out that what is computed in this type of approach is an imprecise model rather than a genuine fuzzy one. This observation motivates the two proposals made in the last two sections. The first approach amounts to summarize precise output data belonging to a sliding window, into fuzzy data and then to apply an appropriate fuzzy regression algorithm. This idea applies to time series data and uses fuzzy similarity-based statistics for building fuzzy data from classical data. The second approach proposes a general framework for learning models together with their uncertainty and imprecision, here specialized for fuzzy regression, and which is based on the minimization of a criteria combining a measure of imprecision of the obtained model together with a measure of accuracy with respect to the data.

II. Two types of fuzzy regression methods

In this section, we emphasize the difference between fuzzy regression models that only reflect the fuzziness of the data and fuzzy regression methods that account for a desired imprecision of the resulting model. In the following data are viewed as pairs of input data (which may be a vector made of several components) and output data, which correspond to what we want to be able to predict.

A. Classical regression adapted to fuzzy data

The first type of fuzzy regression deals with fuzzy output data, and maybe fuzzy inputs, which have to be described by a fuzzy regression function. This type of method has been proposed for dealing with imprecise data by several authors [2], [3]. The method proposed by Diamond [3] is based on fuzzy least squares fitting. In this scope, Diamond define a metrics for fuzzy sets. Having this metrics, the settings of regression are the same than the classical one. The approach described in [2] is based on a similarity measure. Such methods are of interest when fuzzy data are available. Such fuzzy data may be given as such, or may result from a preprocessing of classical data. There are two types of preprocessing. We may either change classical data into fuzzified data whose labels are easier to interpret for the end-user, or obtain fuzzy data from a fuzzy statistical analysis, as exemplified in section 4.

B. Possibilistic regression from non-fuzzy data

The second type of approach is named possibilistic regression and handles ordinary data. A linear model [13] has been initially proposed, that learns a fuzzy regression function from crisp inputs and crisp or interval-valued outputs. In the first step, a linear regression function that produces an interval from crisp data is learned by solving a linear programming problem. Then, an interval-valued linear regression function is deduced. This function associates an interval (rather than a crisp value) to a crisp input. This method has been extended to non linear possibilistic regression in [9], [11]. The advantage of the method is that it is very efficient in terms of computation time and that it can handle noise to some extent.

In particular, the non-linear extension of possibilistic regression based on neural networks [9] has several advantages. First, in contrast with standard possibilistic regression [13], it allows us to be more accurate with respect to the data by inducing a non-linear function. Moreover, the method in [13] was based on linear programming, which may become very costly when dealing with a large amount of data. However, the neural networks approach is more efficient [9] in terms of computation time and the users can fix the trade-off between the computation time and the accuracy of the results. Moreover, it is known that neural networks perform extrapolations that are statistically meaningful.

Nevertheless, the method has some limitations too. Neural networks have difficulties for learning some types of functions such as periodic ones. Neural networks are also very sensitive to outliers (marginal points of the output). An extension based on support vector machine [11], [10] has been proposed for dealing with outliers.

III. Discussion on possibilistic regression

We first recall the algorithm for learning neural networks. Then we use this algorithm for possibilistic regression. Finally we apply the possibilistic regression to environmental data and we discuss the obtained results.

A. Non-linear possibilistic regression with neural networks

Neural networks are a general learning device that induces a function from pairs of input-output data. The function that is learned can then be used for classification or regression purpose. A neural network is made of several nodes, organized in layers, related by edges with weights. Each layer computes a weighted sum of the results of a sigmoid function applied to the outputs of the previous layer. Inputs of the first layer are inputs of the function to be induced. For learning the function, a pair of input-output data is chosen randomly in the set of examples, and the output is compared with the neural network output. Then, the neural network weights are adjusted with respect to a discrepancy function between the output data and the networks output [9]. This learning step is repeated a number of times that is fixed by the user. However, an excessive number of iterations could lead to an over precise learning and maybe to extrapolation anomalies.

We consider m examples of the form $\overrightarrow{x}_p =< (x_{p1}, ..., x_{pn}), y_p >$ where $(x_{p1}, ..., x_{pn})$ are the inputs for the example p and y_p is the output. The goal of a neural network is to learn a function g that minimizes the error with respect to the examples. So, we search a neural network that defines the function g such that

$$\sum_{p=1}^{m} \frac{1}{2}(y_p - g(x_{p1}, ..., x_{p1}))^2$$

is minimized. For our purpose, we only need a neural network with two layers. The first layer is the input unit, the second layer is the hidden unit.

Input units is defined by n neurons $O_{11}, .., O_{1n}$. The value of the neurons when a new example $\overrightarrow{x}_p$ is presented is :

$$O_{1i}(\overrightarrow{x}_p) = x_{pi}$$

The hidden layer contains n' neurons (n' is a parameter fixed by the user) $O_{21}, .., O_{2n'}$. Each neuron of the input layer O_{1i} is linked with each neuron of the hidden layer O_{2j} by the synaptic weight w_{ji}. The value of the neurons when a new example $\overrightarrow{x}_p$ is presented is :

$$O_{2j}(\overrightarrow{x}_p) = sig(net_j(\overrightarrow{x}_p))$$

with

$$sig(x) = \frac{1}{1 + exp(-x)}$$

and

$$net_j(\overrightarrow{x}_p) = \sum_{i=1}^{n} w_{ji} * O_{1i}(\overrightarrow{x}_p) + \theta_j$$

The value of the function g defined by the neural network is given by the output neuron O which is linked with each neuron O_{2j} of the hidden layer by a synaptic weight w_j. The value of the function given by the output neuron for an input $\overrightarrow{x}_p$ is :

$$g(\overrightarrow{x}_p) = O(\overrightarrow{x}_p) = sig(net(\overrightarrow{x}_p))$$

with

$$net_j(\overrightarrow{x}_p) = \sum_{j=1}^{n'} w_j * O_{2j}(\overrightarrow{x}_p) + \theta$$

At each step of the learning process an example is randomly chosen in the training set. Then the error of g is back propagated in the neural network. The values of the synaptic weights w_{ji},w_j and the values θ_j and θ are updated in order to minimize the error.

The possibilistic regression [9] learns two neural networks, one for the upper bound regression, one for the lower bound regression. Then, given any input data, an interval output can be computed. More generally, it can interpreted as a

fuzzy interval, taking this interval, with some tolerance, as the support of the fuzzy interval. The core of the fuzzy set is given by the median value of the support. The level cuts of the fuzzy sets can be viewed as confidence intervals. However, this fuzzy reading of the interval found remains a bit adhoc.

The upper bound function is denoted as g^*. For learning this function, we learn a neural network that minimizes the value :

$$\sum_{p=1}^{m} \frac{1}{2} * \omega_p * (y_p - g^*(\overrightarrow{x}_p))^2$$

where the value of ω_p depends on whether y_p is less or greater than $g^*(\overrightarrow{x}_p)$. More precisely, we use the following weighting scheme :

$$\omega_p = \begin{cases} 1 \text{ if } y_p > g^*(\overrightarrow{x}_p) \\ \omega \text{ if } y_p \leq g^*(\overrightarrow{x}_p) \end{cases}$$

where ω is a small positive value in the open interval $]0, 1[$. If r is the number of learning steps of the algorithm, at step t we have :

$$\omega(t) = \frac{1}{1 + (\frac{t}{r/20})^3}$$

This weight allows us to have a function that is just greater than the target function. In the same way, the lower bound g_* is computed by considering the minimization of :

$$\sum_{p=1}^{m} \frac{1}{2} * \omega_p * (y_p - g_*(\overrightarrow{x}_p))^2$$

where

$$\omega_p = \begin{cases} 1 \text{ if } y_p < g_*(\overrightarrow{x}_p) \\ \omega \text{ if } y_p \geq g^*(\overrightarrow{x}_p) \end{cases}$$

Finally, given a new input $\overrightarrow{x}$, possibilistic regression computes an interval defined by $[g_*(\overrightarrow{x}), g^*(\overrightarrow{x})]$.

B. Illustration on environmental data

The problem considered here for illustrative purpose is to learn the concentration of a polluting agent in a water spring. Possibilistic regression is used since it handles crisp inputs and outputs. The main benefits of possibilistic regression with respect to classical regression is that possibilitic regression both describes the general tendency of the function (as classical regression does) and the amount of imprecision around the general tendency. In this application, allowing for imprecise models enables the user to know to what extent the prediction can be made in a precise way. Thus an interval that expresses the uncertainty of the prediction provides a more valuable piece of information for a comparison with a reference threshold (for firing alerts for instance).

In the first experiment (Fig. 1 and 2), input is time and output corresponds to the observe concentration of atrazine pollutant. For each output, results are presented for the learning data period plus five years, the learning data period plus twenty years. For each curve, we present the results with crisp interval (the middle function represents the result of classical non-linear regression) and its representation with triangular fuzzy sets (the support is the interval and the peak of the triangle is defined by the value of the classical regression). In the second experiments (Fig 3 and 4), the outputs is the concentration of deethylatrazine pollutant.

For the two experiments we use neural networks with 5 hidden neurons. Learning is made with $100,000,000$ of iterations. Data contains more than 300 pairs of input-output pieces of data.

Several comments are in order. What is satisfactory here is that the found models fit the data quite well as can be seen on Figure 1. Moreover, what is predicted is an interval for the pollution rate, which is more satisfactory than a precise value as classical regression would provide. However, since the upper and lower bounding functions are learned separately, there is no guaranty that the two curves do not cut each other outside the data period in the prediction area, which would make no sense. We observe it in the Figure 2. Moreover, if a standard regression function is learned separately, there is also no guaranty that this function will be consistent with upper and lower bound functions, see also Figure 2.

Besides, what is learned is just an imprecise model rather than a fuzzy one. We may think of fuzzifying this interval by associating the output result with a triangular fuzzy number rather than keeping a crisp interval, as suggested by Tanaka. The peak of the fuzzy number may be chosen as the center of the interval or better as the corresponding point on a standard regression curve (consistent with the upper and lower bounds). The support of the fuzzy number is then the interval we start with. Clearly, this is just a way of having the result dressed in a fuzzy manner, which is not very meaningful, and is not fully satisfactory.

In the following, we suggest two extensions where a fuzzy interval function is more naturally obtained.

IV. Toward genuine fuzzy regression for time series learning

In this section, we start from the remark that regression methods aims at revealing the main tendency in a function, e.g. in time series, while ignoring the small variations that are not very meaningful.

Thus, the idea is to perform a preprocessing of the precise data by summarizing them in a fuzzy way in order to get rid of small variations. Then this fuzzily summarized data will be processed by a fuzzy regression algorithm of the first type described in section 2.A.

The summarization process can be described as follows. Let us first consider an interval window W on the input data domain (the time for instance). Let G be the graph of the

Fig. 1. Atrazine w.r.t. time with intervals

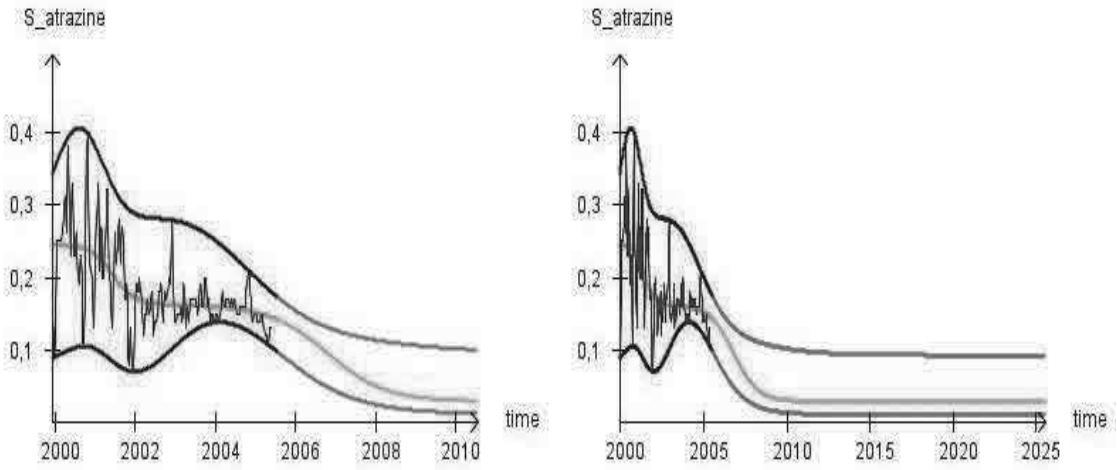
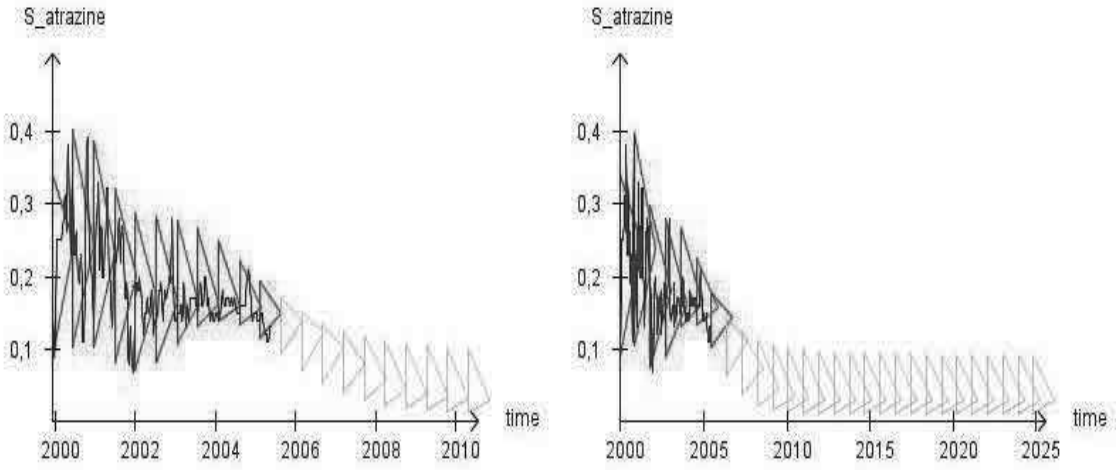


Fig. 2. Atrazine w.r.t. time with fuzzy sets



piecewise linear function obtained by relating each piece of data with its immediate neighbors with respect to the input domain that is used for defining W. Then for each value y of the output data range we compute the cardinality

$$P(y) = |\{x \in W | y = G(x)\}|.$$

Once normalized, the histogram obtained from the $P(y)$'s yields a probability distribution p. This probability distribution p can be turned into a possibility distribution [6]

$$\pi(y) = \int_R min(p(y), p(z))dz.$$

This can be generalized by using a fuzzy window W with membership function μ_W (e.g. triangular-shaped) in order to give more importance to the close neighbors. Then P can be written

$$P(y) = \sum_{x \in W | y = G(x)} \mu_W(x)$$

provided that W as a finite support and there is no horizontal plateau in G, using the idea of fuzzy summarization first expressed in [8].

Finally, we use a sliding window around each piece of input data. Thus, each piece of data $< x, y >$ is fuzzily summarized into a pair $< x, \pi_x >$.

One expected advantage of the approach is that the neighborhood is taken into account for each example. Moreover, performing this preprocessing of the data together with a

Fig. 3. Deethylatrazine w.r.t. time with intervals

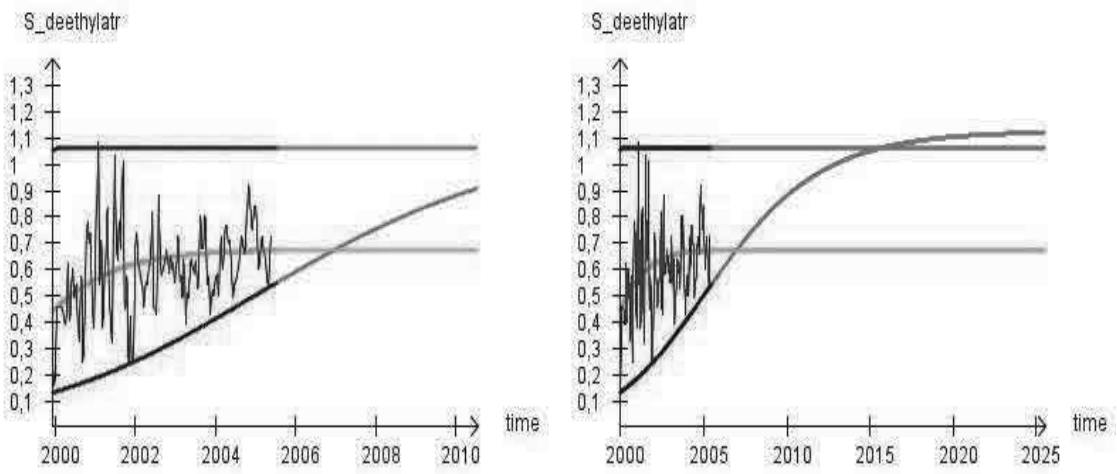
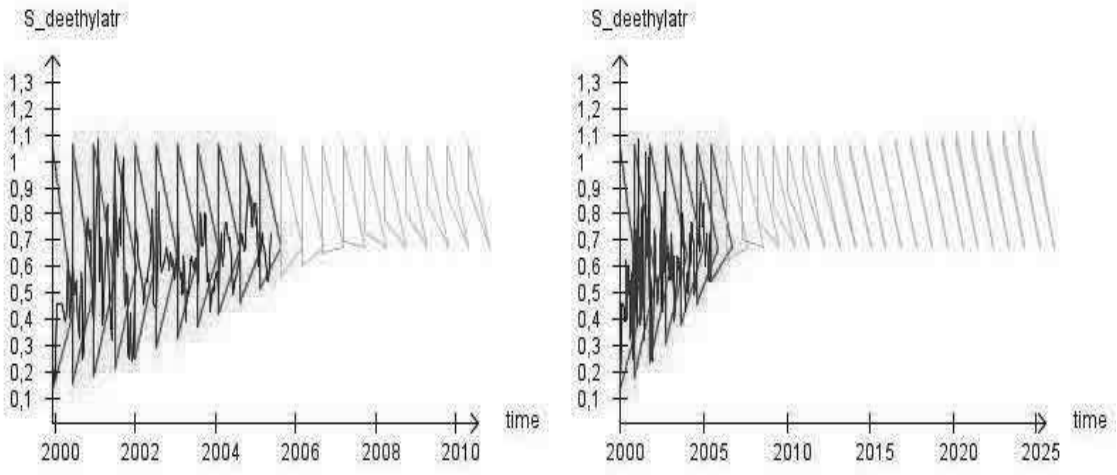


Fig. 4. Deethylatrazine w.r.t. time with fuzzy sets



regression method for fuzzy data provides a meaningful way of obtaining a fuzzy regression function from non-fuzzy data.

V. A GENERAL FRAMEWORK FOR IMPRECISE REGRESSION

In this section, we suggest a more general framework for imprecise or uncertain regression from precise data, and instantiate it in the fuzzy framework. Given a regression problem, the goal of imprecise regression is to find a model that makes as little error as possible and that is as precise as possible. In this context, we assume that the error is related to the imprecision, i.e., the more errors the model makes, the more imprecise the model is.

This problem can be stated as follows. Given $\mathcal{X}$ the domain of input data and R the range of output data, a fuzzy function f is a function from $\mathcal{X}$ to $(R \rightarrow [0,1])$ that associates the possibility distribution π_x to the input x. Let us define Pr a precision index from the set of possibility distributions to $[0,1]$ with the following properties :

- $Pr(\pi) = 1$ iff $\exists! y \in R$ such that $\pi(y) = 1$ and $\pi(y') = 0$ for $y' \neq y$.
- $Pr(\pi) = 0$ iff $\forall y \in R, \pi(y) = 1$.
- $Pr(\pi) \geq Pr(\pi')$ iff $\pi' \geq \pi$.

The first property states that the only type of fully precise function are classical ones, and the second property states that the least precise function is the one that gives no information. The last property states that precision ordering is reversed with respect to fuzzy set inclusion.

Then the function that we are led to maximize is for a set of m examples $< x_i, y_i >$, $x_i \in \mathcal{X}$, $y_i \in R$:

$$\sum_{i=1}^{m} \pi_{x_i}(y_i) * Pr(\pi_{x_i}).$$

A possible choice for π can be triangular-shaped fuzzy sets [5] and precision index could be (e.g. see [7]) :

$$Pr(\pi) = \int_0^1 \frac{1}{|\pi_\alpha|} d\alpha$$

with $\pi_\alpha = \{y, \pi(y) \geq \alpha\}$.

VI. Conclusion

The non-linear possibilistic regression method based only on neural networks has been illustrated on environmental data. Non linear regression seems to be able to both identify the general tendency and the dispersion of the data. The benefit with respect to standard regression methods is clear : considering the upper bound curves that is induced, one can more easily predict the risk of exceeding reference levels.

However, some limitations of possibilistic regression have been identified. In particular, the possibilistic regression framework only induces interval-valued functions rather than genuine fuzzy functions. We have outlined two approaches that cope with this limitation. In the first approach we propose to turn the precise output data into fuzzy output data in time series. This is done by using a statistical analysis around the output data concerned. Then, fuzzy regression methods that deal with fuzzy output data can be used. In the second approach, we suggest a general framework for fuzzy learning from crisp data. In the future, these two approaches need to be experimented in order to compare them with possibilistic regressions.

References

[1] H. Bandemer and W. Naether. *Fuzzy Data Analysis*. Kluwer Academic Publishers, Dordrecht, Boston, London 1992, 1992.

[2] A. Celmins. Least squares model fitting to fuzzy vector data. *Fuzzy Sets Syst.*, 22(3):245–269, 1987.

[3] P. Diamond. Fuzzy least squares. *Inf. Sci.*, 46(3):141–157, 1988.

[4] P. Diamond and H.Tanaka. Fuzzy regression analysis. *Fuzzy Sets in Decision Analysis, Operations Research and Statistics. The Handbooks of Fuzzy Sets Series. Kluwer Academic Publishers*, pages 349–387, 1998.

[5] D. Dubois, L. Foulloy, G. Mauris, and H. Prade. Probability-possibility transformations, triangular fuzzy sets, and probabilistic inequalities. *Reliable Computing*, 10:273–297, 2004.

[6] D. Dubois and H. Prade. Unfair coins and necessity measures: towards a possibilistic interpretation of histograms. *Fuzzy Sets and Systems*, 10:15–20, 1983.

[7] D. Dubois and H. Prade. Properties of measures of information in evidence and possibility theories. *Fuzzy Sets and Systems*, 24:161–182, 1987.

[8] D. Dubois and H. Prade. On data summarization with fuzzy sets. In *5th Inter. Fuzzy Systems Assoc. Cong. 93*, pages 465–468, Seoul, Korea, 1993.

[9] H. Ishibuchi and H. Tanaka. Fuzzy regression analysis using neural networks. *Fuzzy Sets and Systems*, 50:57–65, 1992.

[10] j. Jin-Tsong, C. Chen-Chia, and S. Shun-Feng. Support vector interval regression networks for interval regression analysis. *Fuzzy Sets Syst.*, 138(2):283–300, 2003.

[11] J. T. Jenga, C. C. Chuang, and S. F. Su. Support vector interval regression networks for interval regression analysis. *Fuzzy Sets and Systems*, 138 (2):283–300, 2003.

[12] J. Kacprzyk and M. Fedrizzi (Eds.). *Fuzzy Regression Analysis*. Omnitech Press, Dordrecht, Boston, London 1992, 1992.

[13] H. Tanaka. Fuzzy data analysis by possibilistic linear models. *Fuzzy Sets Syst.*, 24(3):363–376, 1987.

[14] H. Tanaka and P.J. Guo. *Possibilistic Data Analysis for Operations Research*. Studies in Fuzziness and Soft Computing, Vol. 29. Warsaw, and Physica, Heidelberg, 1999.