Izabela REJER
WEST POMERANIAN UNIVERSITY OF TECHNOLOGY OF SZCZECIN,
Piastów 17, 70-310 Szczecin

INTEGR - A Method for Integrating Expert Knowledge and Knowledge Derived from a Data Set

Ph.D. Izabela REJER

She works at West Pomeranian University of Technology. She has participated and managed some government projects, mainly related with applications of neural networks in economics. Her interests are connected with economic application of neural networks and fuzzy logic – especially for bankruptcy and unemployment prediction. In 2003 she built an expert system for the Polish government (based on neural networks approach) which task was to evaluate a possibility of a bankruptcy of a given corporation.

e-mail: irejer@wi.zut.edu.pl

Abstract

The aim of the paper is to present a new approach for integrating expert knowledge with knowledge derived from a data set, called INTEGR. INTEGR approach is based on the method of training an expert fuzzy model with a set of data points but eliminates main drawbacks of this method. The paper presents both – the theoretical description of INTEGR algorithm and its practical application.

Keywords: fuzzy model, knowledge integration, expert model, model training.

INTEGR - metoda integracji wiedzy eksperckiej z wiedzą wydobytą z danych pomiarowych

Streszczenie

Zarówno modele rozmyte budowane na podstawie zbioru danych pomiarowych jak i modele rozmyte budowane przy wykorzystaniu wiedzy eksperckiej mają specyficzne dla siebie wady i zalety. Model ekspercki jest modelem przybliżonym, ale obowiązującym w całej dziedzinie analizowanej zależności, natomiast model zbudowany na podstawie zbioru danych pomiarowych jest modelem dokładnym, ale wiarygodnym tylko w pewnym, ściśle określonym fragmencie dziedziny. Wynika z tego, że aby zwiększyć precyzję rozmytych modeli eksperckich należy do nich dołączyć wiedzę zawartą w danych pomiarowych i analogicznie, żeby poszerzyć zakres stosowalności modeli rozmytych opracowanych na podstawie zbioru danych pomiarowych należy dołączyć do nich wiedzę ekspercką. Celem niniejszego artykułu jest przedstawienie nowej metody integracji wiedzy eksperckiej z wiedzą wydobytą z danych pomiarowych (metody INTEGR). Opisywana metoda jest oparta na uczeniu eksperckiego modelu rozmytego przy pomocy zbioru danych pomiarowych, ale eliminuje podstawowe wady tego podejścia. W artykule zaprezentowano zarówno teoretyczny opis metody, jak i jej praktyczne wykorzystanie na przykładzie modelu rozmytego przeznaczonego do wyceny samochodów używanych.

Słowa kluczowe: model rozmyty, integracja wiedzy, model ekspertowy, uczenie modelu.

1. Introduction

There are two general approaches which can be used when a fuzzy model is to be created – create model automatically on the basis of numeric data or build model manually with an assistance of a domain expert. It is difficult to decide which approach gives better results because both have their own drawbacks and benefits. Expert models are complete and credible in the whole input domains of analyzed systems and describe only physically sound regions of these domains. On the other hand, however, they need domain expert knowledge (which is often difficult to gain) and in case of highly non-linear systems can generate less precise results than models built automatically with a set of data points.

Models of the second type – it is models utilizing knowledge derived from a set of data points, are in most cases much more precise than expert fuzzy models and their creation is not as time consuming as interviewing a domain expert. Unfortunately, the overall quality of this type of fuzzy models depends on the quality of data set which in reality is often insufficient to built a model producing credible results in the whole physically sound input domain of the analyzed system. As a result, models built for most practical applications give precise results but only in a very small part of the whole input domain – this part which is properly covered by data points [1].

Hence, in order to preserve benefits of both types of fuzzy models and simultaneously avoid their drawbacks - in other words, in order to built a fuzzy model producing credible results in the whole input domain and simultaneously precise in the region covered by data points used in the process of estimating model parameters - knowledge from both sources (expert and set of data points) should be utilized. There are only a few methods which can be used to join expert and data knowledge (like gray-box modeling [2], semi-mechanistic modeling [3], maintaining a set of rule bases joined with meta-rules [4], Bayesian networks [5] etc.) and none of them can be used in order to build a model which will be credible in the whole, physically possible, input domain of the analyzed relation, like an expert model and, simultaneously, will be highly precise in the region covered by data points used in the estimation process, like a model created automatically on the basis of a data set.

One of the best alternatives for joining expert and data knowledge is a method of training a fuzzy expert model with a set of data points [6]. This method is often used in practical applications, however, due to applying knowledge derived from data set after expert knowledge, it does not utilizes the total amount of gathered knowledge. Depending on the trained model parameters this method omits some information contained in the data set (which causes that the resulting model is less precise) or omits some information given by the domain expert (which causes that the resulting model loses its extrapolation capabilities).

The aim of this paper is to present a new method for integrating expert knowledge and knowledge derived from a data set, called INTEGR, which is based on the method of training of an expert model but which avoids main drawbacks of this method – it is which utilizes the whole amount of expert and data knowledge. The content of the paper is as follows. Section II shortly describes the main concepts of fuzzy modeling; Section III presents the method of training an expert model with a data set; Section IV introduces proposed method for integrating a fuzzy expert model with a fuzzy model built automatically over a data set (INTEGR) and finally Section V presents an application of INTEGR method for a car price evaluation problem.

2. Fuzzy model

The most natural mathematic representation of both knowledge types (knowledge provided by a domain expert and knowledge derived from a data set) is a fuzzy model. Therefore, when both knowledge types are to be used simultaneously, the best idea is to use the fuzzy logic theory. A fuzzy model is a model which is defined in terms of membership functions and rules. The mathematic form of a fuzzy model depends on applied mathematics engine. For example, the mathematic form of a fuzzy model used in the research part of the paper, is characterized by the following parameters:

- Larsen model [7],

– grid partitioning of an input space,
– singleton membership functions of output variable,
– PROD-MAX inference mechanism,
– weighted average sum defuzzification method.

The form is following [8]:

$$y_0 = \frac{\sum_{i=1}^{m} y_i \left(\mu_{B_i}(y_i) \prod_{j=1}^{s} \mu_{A_{ij}}(x_j) \right)}{\sum_{i=1}^{m} \left(\mu_{B_i}(y_i) \prod_{j=1}^{s} \mu_{A_{ij}}(x_j) \right)}, \tag{1}$$

where: y_0 - output variable, x_j - input variable j (j=1,...,s), y_i – conclusion of i rule (i=1,...,m), $\mu_{Aij}(x_j)$ - degree of activation of j premise of i rule, $\mu_{Bi}(y_j)$ - degree of activation of i rule conclusion.

In order to create a fuzzy model given by (1), the following scheme should be performed:

– define one set of membership functions per each model variable (input and output),
– create a rule net of the fuzzy model by joining cores of fuzzy sets of succeeding input variables,
– define a conclusion per each physically possible combination of fuzzy sets defined over domains of succeeding input variables,
– apply a chosen mathematic engine.

All steps of the above algorithm can be performed automatically on the basis of numeric data or manually with an assistance of a domain expert.

3. Training of a fuzzy expert model

One of the methods used for joining expert knowledge with knowledge derived from a data set, commonly used in practice, is a method of training an expert model with a set of data points. The algorithm of this method has three main steps:

– built an expert fuzzy model,
– use the parameters of the expert model as initial parameters of the fuzzy neural network,
– train the network with a set of data points.

There are three approaches which can be used in the third step of this algorithm: training rules conclusions, training membership functions used in rules premises, training rules conclusions and membership functions used in rules premises simultaneously. The first approach – training rules conclusions – is used for changing conclusions of rules which support points are situated in the region of input space covered by data points. Hence, this approach improves the precision of an expert model in the region of data points. Unfortunately, this improvement is often very small because only part of total amount of information contained in the data set is utilized. Remaining information – information which could be used to change rules premises is lost.

While automatic training of rules conclusions changes the model mostly inside the region of input domain covered by data points, training of membership functions used in rules premises (second approach) influences the model also in the domain covered by expert rules. This is due to the fact that after changing supporting point of one membership function, all rules containing this function in their premises are changed automatically. This situation is very unwelcome because the model loses the main advantage of the expert model – the credibility in the whole input domain (the region of model credibility shrinks to the region covered by data points). That means that the expert knowledge is removed from the model and the effort taken to create the expert model was completely unnecessary.

Moreover, it often happens that by shifting rule nodes (with fixed conclusions), the model performance gets worse not only in the region covered by expert rules but also in the region covered by data points. It is due to the fact that the task of the training process in this case is to shift the rule nodes in such a way to match fixed rules conclusions with a set of data points.

The last of mentioned approaches used in the training process - training of rules conclusions and rules premises simultaneously – at first seems to be the most promising because it gives the most precise results. Unfortunately, this approach has the same drawback as the second one – the results of the fuzzy model are credible only in the region of input domain covered by data points. The application of this model beyond this region is unjustified.

4. INTEGR method

In order to eliminate the main drawbacks of the method described in previous section, the following algorithm (called INTEGR) is proposed:

1. Two fuzzy models of the analyzed system are created – an expert model and a model based on data set (called: E-model and D-model, respectively).
2. Rule nets of both fuzzy models are integrated by joining pairs of sets of membership functions describing succeeding input variables.
3. Conclusions of all rules from E-model are rewritten to the final model.
4. Conclusions of all newly created rules are calculated using the closest neighborhood approach.
5. Conclusions of all rules of the final model are tuned in the process of training neuro-fuzzy network.

In the first step two fuzzy models of the same relation – a model based on a data set (D-model) and a model defined by a domain expert (E-model) – are built (using a grid partitioning of the input space). Then a rule net of the final fuzzy model is created (step 2). To deal with this task pairs of sets of one-dimensional membership functions describing succeeding input variables used in D-model and E-model are joined together. In this way one coherent set of membership functions per each input variable is created. Cores of these functions are then combined and a rule net of the final model is created.

After creating the rule net in the input space, rules conclusions have to be established. Of course, there is no need to calculate conclusions for all rules – some of them are known from elementary models. Since the final model has to be valid in the whole input space, the priority is given to E-model, which means that at the beginning all rule conclusions from E-model are rewritten to the final model (step 3).

The whole rule net of the final model consists of Lr nodes. Only some of them are rewritten from E-model – rest of them have to be established with other method. This task is performed in steps 4 and 5. At first (step 4), conclusions of all new rules are established on the basis of conclusions of neighboring rules. According to this approach, in order to establish a conclusion of an individual rule, all rules which antecedents are situated next to the antecedents of the analyzed rule in the whole multi-dimensional input space have to be revealed. Then, the conclusions of all rules are gathered together and a conclusion of the analyzed rule is calculated as a simple or weighted average.

Conclusions obtained after applying the closest neighborhood approach are more or less general. Some of them can be additionally improved by using knowledge contained in the data set. Hence, in the final step of the proposed algorithm rules conclusions of the whole model are tuned in the training process. Obviously, although the whole model is trained, only rules covered by data points are changed. Conclusions of remaining rules remain untouched on the level given by the expert or on the level calculated with the closest neighborhood approach.

The important observation here is that steps 3-5 (and their order) were designed in such a way to ensure that the main advantages of both fuzzy models (D-model and E-model) will be preserved. The algorithm steps order preserves knowledge taken from both sources (data set and expert) and ensures that knowledge derived from data set is applied inside the region covered by data points, and knowledge provided by the domain expert is applied outside this region. Thanks to this, the final model is credible in the whole domain and highly precise in the region covered by data points.

5. Case Study

In order to find out how INTEGR method works in practice, a model for car price evaluation was created. Data for the study were taken from UCI Machine Learning Repository (file: Automobile) [9]. The model input variables were *engine size* and *highway mpg* (miles per gallon), and the model output variable was *car price*. INTEGR algorithm was implemented in Matlab environment and ran on 2-processor IBM PC 2*2,2 GHz under Windows XP operating system.

The first stage of the experiment was to build two fuzzy models of the relation between *car price* and *engine size* and *highway mpg*. The basic parameters of both models were the same, it is:

- model type – Larsen model, given by (1),
- input membership functions - asymmetrical triangular functions (3 functions per each input variable),
- output membership functions – 9 singleton functions.

Before creating the first model – model based on a data set (D-model) – the whole data set, composed of 201 data points, was divided into two sets: training set (189 data points) and testing set (12 data points). While the training set was created in order to estimate the model parameters, the testing set was created in order to compare performance of all three fuzzy models (D-model, E-model and the integrated model). A distinct feature of the testing set was that it was composed mainly of observations situated outside the interpolation region of D-model. Due to low dimensionality of the analyzed relation, there was no need to reserve data points for D-model testing process.

To prepare data from the training set for estimating D-model parameters, all variables (input and output) were normalized to the interval <0, 1> with max-min method. In order to estimate D-model parameters (it is cores of input and output membership functions), the model was trained with a fuzzy-neural network (of architecture matched to Larsen model given by (1)). The training process was carried out by 200 epochs according to the backpropagation algorithm with momentum rate. The average accuracy of the model was calculated according to mean absolute error (MAE) [10]:

$$MAE = \frac{\sum_{k=1}^{n} \left| y_k^* - y_k \right|}{n} \times 100\% , \qquad (4)$$

where: y_k* - real values, y_k – theoretical values.

MAE of D-model calculated over the training set was equal to 6.15%. The model surface is presented in Fig. 1a.

Next, the expert model (E-model) was built with an assistance of a domain expert. In order to built this model, the expert was interviewed in regard to the following questions:

1. How should each variable be interpreted?
2. How should the potential values of succeeding input variables be aggregated to form the overall evaluation?

The first question was a question about parameters of membership functions which should be used to describe model input and output variables. Addressing this question, the expert provided information (given in Tab. 1) about:

- number of linguistic terms which should be used to describe each input and output variable,
- names of these terms,
- numeric cores of these terms.

The second question was a question about connections between input variables and output variable, it is about model rules. Answering this question, the expert built a rule base presented in Tab. 2.

When membership functions and rules given by the expert were joined together with mathematic engine of a Larsen model, a model of a surface presented in Fig. 1b. was created. Mean absolute error of this model, calculated over the data set used for estimating D-model parameters, was equal to 9.35%.

Tab. 1. Linguistic terms and numeric cores assigned to succeeding variables
Tab. 1. Etykiety lingwistyczne i rdzenie numeryczne przypisane do kolejnych zmiennych

Variable	Very low value (VLV)	Low value (LV)	Average value (AV)	High value (HV)	Very high value (VHV)
engine size	-	61	193.5	326	-
highway mpg	-	16	35	54	-
car price	5000	10000	15000	25000	45500

Tab. 2. Expert rule base
Tab. 2. Ekspercka baza reguł

		Engine size		
		Low value	Average value	High value
Highway mpg	Low value	AV	HV	VHV
	Average value	LV	AV	HV
	High value	VLV	VLV	VLV

a)

b)

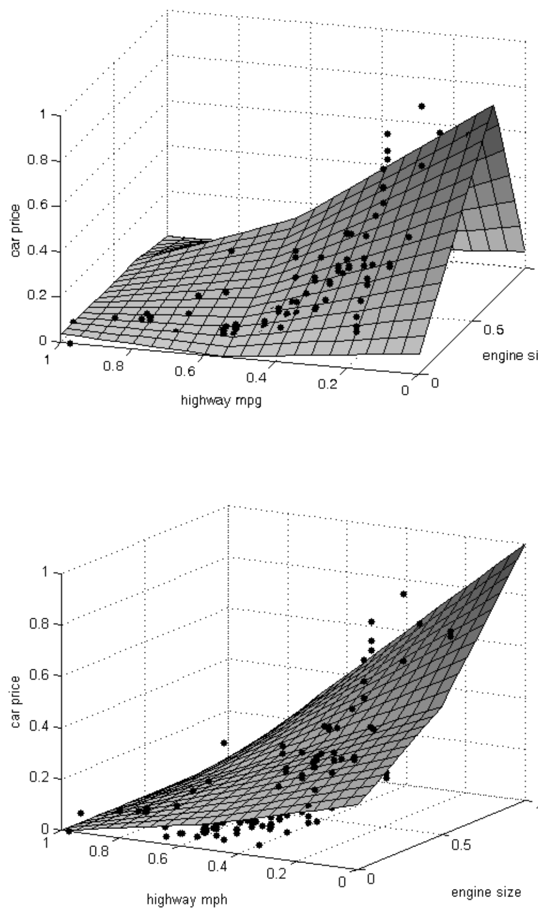
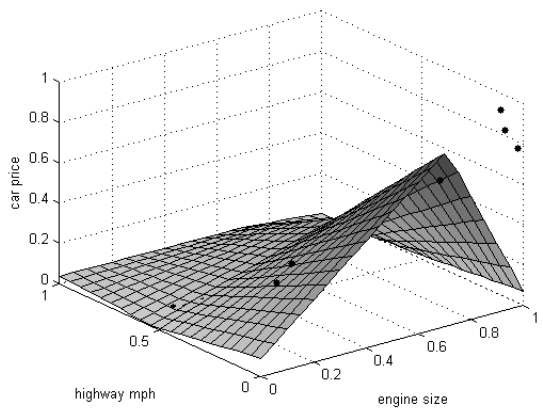


Fig. 1. Surface of D-model (a) and E-model (b) over data from the training set
Rys. 1. Powierzchnia D-modelu (a) i E-modelu (b) na tle danych ze zbioru uczącego

In order to compare the performance of both elementary models (D-model and E-model), the testing set was used. Fig. 2 presents distribution of testing observations over the surface of D-model (Fig. 2a) and E-model (Fig. 2b). The models errors (MAE) were equal to:
- 19.88% - in case of D-model,
- 8.32% - in case of E-model.

a)



b)

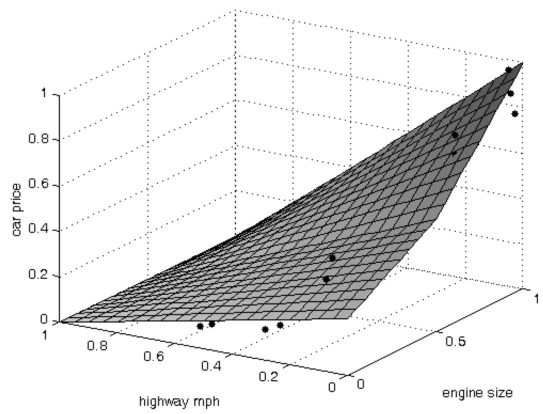


Fig. 2. Surface of D-model (a) and E-model (b) over data from the testing set
Rys. 2. Powierzchnia D-modelu (a) i E-modelu (b) na tle danych ze zbioru testowego

Comparing errors of both models, it should be said that while the error of E-model is quite satisfactory (of course only in case of an expert model), D-model performance is unacceptable. So large, almost equal to 20%, error is not acceptable in any real market (it is of no importance whether it is a car selling market, financial market or any other). In reality, a model of so large error is practically useless. However, the low training error of this model proves that it correctly maps data from the training set. Therefore, it can be used for car price evaluation but only for observations coming from the region covered by training data - after this region it needs support from E-model.

Hence, after creating both elementary models, their rule nets were joined together and the rule net of the final model was obtained. Next, three last algorithm steps (3, 4 and 5) aimed at rules conclusions calculation were performed. After each step the final model performance was tested with both - the training and testing set. The results of the testing process are presented in Tab. 3.

Tab. 3. Mean Absolute Errors of all created models
Tab. 3. Średnie błędy bezwzględne kolejnych modeli

	MAE [%] of		MAE [%] of the final model after:		
	D-model	E-model	3rd step	4th step	5th step
Training set	6.15	9.35	8.84	8.84	6.15
Testing set	19.88	8.52	8.05	8.05	6.27

6. Conclusions

According to results gathered in tab. 3, the application of INTEGR method gave very promising results. The testing error of the final model, obtained after the fifth algorithm step (6.27%) was about three times lower than the error of D-model (19.88%) and was also lower than error of E-model (8.52%). That means that the model performance was improved inside the interpolation region (controlled by D-model) and also was not deteriorated outside this region (controlled by E-model). That also means that the integration process was performed correctly and that knowledge derived from data set was used inside the interpolation region and expert knowledge outside this region.

At first look it can be surprising that after the forth algorithm step, the model performance did not improve. The reason for this is that the aim of this step was to complete the model, not to improve its performance. Without this step the model would be an incomplete one and therefore impossible to apply in practice.

The work was supported by finances of West Pomeranian Provincial Administration.

7. References

[1] Piegat A.: Fuzzy Modeling and Control, New York: Physica-Verlag (2001).
[2] Lindskog P.: Fuzzy Identification from a Grey Box Modeling Point of View. In: Hellendoorn H., Driankov D. (eds.): Fuzzy Model Identification, pp. 3-50, SpringerVerlag (1997).
[3] Abonyi J., Roubos H., Babuška R., Szeifert F.: Interpretable Semi-Mechanistic Fuzzy Models by Clustering, OLS and FIS Model Reduction. Modeling and the interpretability-accuracy trade-off. Part I, Interpretability Issues. Chapter 10. In: Casillas J., Cordon O., Herrera F., Magdalena L.: Studies in Fuzziness and Soft Computing, Physica-Verlag (2003).
[4] Wang Ch., Hong T., Tseng S.: Integrating membership functions and fuzzy rule sets from multiple knowledge sources. Fuzzy Sets and Systems, No. 112 (2000).
[5] O'Leary D. E.: Knowledge Acquisition from Multiple Experts: An Empirical Study. Management Science, vol. 44, no. 8, August (1998).
[6] Rejer I.: Training a fuzzy expert model with a set of data points. Proceedings on conference on Human System Interaction. IEEE Catalog Number: 08EX19995C, ISBN: 1-4244-1543-8, Kraków (2008).
[7] Butkiewicz B. S.: Metody wnioskowania przybliżonego. Właściwości i zastosowania. Oficyna Wyd. Pol. Warszawskiej, Warszawa (2001).
[8] Rutkowska D., Piliński M., Rutkowski L.: Sieci neuronowe, algorytmy genetyczne i systemy rozmyte. Wydawnictwo Naukowe PWN Sp. z o.o., Warszawa (1999).
[9] Asuncion A. & Newman D.J.: UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science (2007).
[10] Aczel, A.D.: Complete business statistics. Richard D. Irwin Inc., Sydney (1993).

otrzymano / received: 13.07.2010
przyjęto do druku / accepted: 01.09.2010 *artykuł recenzowany*