РАЗМИТИ СИСТЕМИ ЗА ИЗВОД ЧРЕЗ СИСТЕМИ РАЗМИТИ РЕЛАЦИОНАЛНИ УРАВНЕНИЯ

Златко Захариев

Резюме. *Представена е имплементация на размити системи за извод с използване на системи размити релационални уравнения. Разглеждат се правата и обратната задачи в размитите системи релационални уравнения. Даден е пример, както и няколко приложения на представената имплементация.*

FUZZY REASONING THROUGH FUZZY LINEAR SYSTEMS OF EQUATIONS

Zlatko Zahariev

Abstract. *Application of fuzzy linear systems of equations (FLSE) in fuzzy inference module is presented here. It is a technique to represent fuzzy reasoning with FLSE. Direct and inverse problems are studied and an example is given. Applications of this approach are also described.*

1. Introduction

A technique to represent Mamdani type fuzzy inference systems (FIS) with fuzzy linear systems of equations (FLSE) is presented here. Some limitations of this technique are also described as well as possible methods to handle them.

The paper is divided in seven sections. In section 2 FIS are presented. Only information needed for this paper is given. In section 3 FLSE are introduced. Section 4 is the most important part of this paper where a technique to represent FIS with FLSE is described. Also, there is information about the usage, limitations and applications of this technique. An algorithm is also presented in this section. The example in section 5 is chosen to be simple and well known. Section 6 is about some methods to handle the limitations of the presented here technique. Conclusions are given in section 7.

2. Fuzzy inference systems

Fuzzy inference systems are quite classical topic. For this reason only a short description is given here. The purpose of this description is not to study such systems, but just to situate the current paper into a topic. A lot of literature on FIS and their usage can be found in [1], [2], [3], [5], [10], [11] and [12].

In general fuzzy inference process is separated in five steps.

1. Input data
2. Fuzzification
3. Fuzzy reasoning (FR)
4. Aggregation
5. Defuzzification

More about any of these steps can be found in [3]. In this paper a method to implement step 3 using FLSE is proposed and developed. This method gives to any FIS some intriguing improvements which are described in section 4.4.

Most kindly FR is permitted by a system of If-Then rules [3], [10]. Each rule actually defines a fuzzy relation between fuzzy variables or fuzzy sets and this relation can be represented by a fuzzy relational equation. Hence, a system of If-Then rules can be represented by a system of FLSE. Details are presented in section 4.

3. Fuzzy linear systems of equations (FLSE)

FLSE are not the main topic of this paper and because of this, only shallow view on them is presented here. More information on FLSE can be found in [4], [6], [7] and [9].

Max-min FLSE has the form:

$$\left|\begin{array}{l} (a_{11}\wedge x_1)\vee(a_{12}\wedge x_2)\vee\ldots\vee(a_{1n}\wedge x_n)=b_1 \\ (a_{21}\wedge x_1)\vee(a_{22}\wedge x_2)\vee\ldots\vee(a_{2n}\wedge x_n)=b_2 \\ \qquad\qquad\qquad\ldots \\ (a_{m1}\wedge x_1)\vee(a_{m2}\wedge x_2)\vee\ldots\vee(a_{mn}\wedge x_n)=b_m \end{array}\right. \qquad (1)$$

where $a_{ij}, b_i \in [0,1]$, $i=\overline{1,m}$, $j=\overline{1,n}$, are given and $x_j \in [0,1]$, $j=\overline{1,n}$ marks the unknown in the system. Operation $\vee$ between two numbers a and b is $a \vee b = max(a,b)$. Operation $\wedge$ between two numbers a and b is $a \wedge b = min(a,b)$.

The system (1) will be represented in matrix form:

$$A \bullet X = B \qquad (2)$$

where $A=(a_{ij})_{m\times n}$ is a matrix of coefficients, $B=(b_i)_{m\times 1}$ holds the right hand side vector and $X=(x_j)_{1\times n}$ is a vector of unknowns.

3.1. Solutions

The next notions are according to [7]:

A vector $X^0=(x_j^0)_{1\times n}$ with $x_j^0 \in [0,1]$, $j=\overline{1,n}$, is called solution of the system $A \bullet X = B$ if $A \bullet X^0 = B$ holds.

The set of all solutions of $A \bullet X = B$ is called complete solution set and is denoted by $\mathbf{X}^0$. If $\mathbf{X}^0 \neq \emptyset$ then the system is called consistent, otherwise it is called inconsistent.

A solution $X^0_{low} \in \mathbf{X}^0$ is called lower solution of $A \bullet X = B$ if for any $X_{low} \in \mathbf{X}^0$ the relation $X^0 \leq X^0_{low}$ implies $X^0 = X^0_{low}$.

A solution $X^0_u \in \mathbf{X}^0$ is called upper solution of $A \bullet X = B$ if for any $X^0 \in \mathbf{X}^0$ the relation $X^0_u \leq X^0$ implies $X^0 = X^0_u$. If the upper solution is unique, it is called greatest (or maximum) solution.

$(X_1, ..., X_n)$ with $X_j \subset [0,1]$ is called an interval solution of the system $A \bullet X = B$ if any $X^0 = (x^0_j)_{n \times 1}$ with $x^0_j \in X_j$ implies $X^0 = (x^0_j)_{n \times 1} \in \mathbf{X}^0$, for each *j*, $1 \leq j \leq n$. Any interval solution of $A \bullet X = B$ whose components (interval bounds) are determined by a lower solution from the left and by the greatest solution from the right, is called maximal interval solution of $A \bullet X = B$.

For the system (2) if the matrices *A* and *X* are given, computing their product is called direct problem resolution. If the matrices *A* and *X* are given, computing the unknown matrix *X* is called inverse problem resolution.

Direct and inverse problems resolution is very essential to the presented paper. Despite that direct problem resolution is a trivial task, solving the inverse one can be much more complicated. Methods and software packages are presented in [6], [7], [13] and [14].

4. Represent FR through FLSE

4.1. Limitations

There are two general limitations introduced here.

- Only one output value per rule is supported. Still, many input variables are supported.
- Only If-Then rules with OR connection between all their fuzzy values are supported.

In spite of that these limitations are very restrictive, some technique to handle them is proposed in section 6.

4.2. Representation

Representing FR through a FLSE can be proceeding by connecting all the parts from the system of If-Then rules with all the parts from the system (2). The connections are as follows:

- The If-Then rules are represented with the matrix A and the vectors X and B.
- The input is represented by the vector X.
- The output is represented by the vector B.

For the first step, there is need to select each column in the matrix A to represent one fuzzy value from the input variables. In this manner if in the FIS there are two input variables and each has three values, the matrix A should has six columns. Each element in the vector X should map out one fuzzy value from all the input variables. Each element in the vector B should map corresponding fuzzy value from corresponding output variable.

Next, the matrix A should be filled in with all the rules in the system. Each row in A represents one If-Then rule.

Let for some FIS, *r* is the number of all If-Then rules and *v* is the number of all fuzzy values. Each element in the matrix A has the following value:

$$a_{ij} = \begin{cases} 1, \text{ if the j}^{th} \text{ fuzzy value is part of the i}^{th} \text{ rule} \\ 0, \text{ otherwise} \end{cases} \quad (3)$$

where $i = \overline{1,r}$, $j = \overline{1,v}$

Also, the vector X should have *v* elements and vector B should have *r* elements.

The input is represented by the vector $X_{v\times 1}$ and the output by the vector $B_{r\times 1}$.

4.3. Direct problem resolution

Direct problem resolution for system (2) is equivalent to processing FR. After a FIS is represented with the system (2) the FR process can be obtained by composing the matrix *A* and the vector *B* with max-min composition. After defuzification of the result vector *X* the output is obtained.

4.4. Inverse problem resolution

Solving the inverse problem for system (2) is much more interesting task. It is also much harder. In general it means to solve the system (2) when A and B are given and X is a vector of unknowns. In the terms of FR, this means that if the system and the output are given, feasible inputs can be determined. This can be really helpful for diagnostics of a system, adjusting the rules, reducing their numbers, etc. Other interesting issue is that in most cases, FLSE has more than one solution. This means that one output can be obtained through many inputs. Using presented below algorithms and software, all solutions can be obtained.

Because the inverse problem resolution is a complex task, complex theory, algorithms and software should be developed in order to solve the problem. Despite that nowadays the problem appears as almost classical, there are still only few relevant algorithms to solve it and even less software implementations.

More information can be found in [6], [7], [13] and [14].

Algorithm highly based on the algorithm described in [6] is used here. The algorithm is improved by the author in order to make it more efficient and a little bit simpler ([13] and [14]). Also a software package named FC^{2}ore and presented in [14] is used here.

Just a short description of the problem, the algorithm and the software is given here. Detailed information can be found in [14].

4.4.1. Algorithm overview

Algorithm 1: Solving max-min FLSE.

1. Obtain input data for the matrices A and B i.e. the system of If-Then rules and the output.
2. Obtain greatest solution for the system and check for consistency. If the system (2) is consistent this means that the output values can be obtained from the current set of If-Then rules. Also the maximal value for all of the inputs is determined here.
3. If the system is inconsistent the outputs in B cannot be obtained with A, go to step 5.
4. Obtain all minimal solutions for (2). With this the whole solution set of the system is obtained i.e. all possible combinations of values for the inputs to obtain the given output.
5. Exit.

4.5. Applications

This approach gives a lot of new opportunities when working with FIS. Since this paper is not dedicated on applications, some of them are just mentioned in this section. Most general, described here approach makes each FIS (under the mentioned limitations) working both, in direct and inverse directions.

4.5.1. Diagnostics

It is very common approach to use FIS for determining some consequences. All the rules in such systems are with structure similar to: If (Conditions) Then (Consequences). As far as, the conditions are observable, the consequences can be obtained with direct problem resolution.

The opposite approach is when the consequence is already observed but there is need to diagnose the cause. Solving the inverse problem, all possible causes can be obtained.

4.5.2. Verification

Using presented approach a FIS can be verified with checking for consistency and solving it in inverse direction. If for some FIS there are also observed examples, the system can be checked against them for consistency and for correctness.

4.5.3. Adjustment

Adjustment to the FIS can be realized with adjusting the matrix A. As by default it contains only 0 and 1, it can contains any other numbers in $[0,1]$ interval, as both the algorithm and the software support this. This is similar to applying fuzzy modificators and can be used for fine tuning a FIS or even to change its entire behavior.

4.5.4. Minimization

Any FLSE can be checked for linear independence [8]. Dependant rows in some FLSE means that they can be removed from the system and this will keep its behavior. Using this FIS can be minimized by removing unnecessary If-Then rules.

4.5.5. More

More applications can be found in [7].

5. Example

An example of representing FIS with FLSE is given in this section. Solving the direct and inverse problem is also demonstrated.

The chosen example is the famous fuzzy tipper, presented in [3] in the part of the MATLAB's fuzzy logic toolbox. It is selected because it is simple and thus very convenient for such an example. Also it perfectly covers all the limitations and finally it is easy available and thus well known.

5.1.Representation

First the model of FLSE should be obtained from the tipper.

Fuzzy tipper has two input variables: Service(S) and Food(F). Service variable has three values and Food variable has two. This means that the matrix A will have five columns. Their meanings are as follows: S:Poor, S:Good, S:Excellent, F:Rancid and F:Delicious.

There is one output variable called Tip(T) with three values: T:Cheap, T:Average and T:Generous. The tip is measured with a percentage of the bill. Also the "Then" part the rules (presented below) show the meaning for each component in B.

Three rules should be represented in the matrix A. They are:

1. If (S:Poor) or (F:Rancid) Then (T:Cheap)
2. If (S:Good) Then (T:Average)
3. If (S:Excellent) or (F:Delicious) Then (T:Generous)

From these rules matrix A is obtained to be:

$$A_{3\times 5} = \begin{array}{c} \begin{array}{ccccc} S:Poor & S:Good & S:Excellent & F:Rancid & F:Delicious \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

The vector B has three components. Their meaning is taken from the three rules. In this example b_1 stands for T:Cheap, b_2 stands for T:Average and b_3 stands for T:Generous.

With this the system is represented in the form

$$A_{3\times5} \bullet X_{5\times1} = B_{3\times1} \qquad (4)$$

Next step is to solve the system (4). First the direct problem is applied then with concrete values for vectors X and B the inverse. FC^2ore is used for this.

Let the input given values to be S = 2, F = 2. From here, after fuzzification there is: S:Poor = 0.5, S:Good = 0.2 and F:Rancid = 0.2. All other values are equal to 0. This means that $X_{5\times1} = (0.5 \quad 0.2 \quad 0 \quad 0.5 \quad 0)'$.

5.2. Direct problem

Direct problem resolution is equivalent to composing $A_{3\times5} \bullet X_{5\times1} = B_{3\times1}$ with max-min composition.

After this is done the answer is $B_{3\times1} = (0.5 \quad 0.2 \quad 0)'$. And after defuzzification based on centroid function, the final answer is ***Tip = 7.52%.***

5.3. Inverse problem

For solving the inverse problem the last obtained *B* will be used. Now $A_{3\times5}$ and $B_{3\times1}$ are given. Using FC^2ore the FLSE is solved and all possible values of $X_{5\times1}$ are obtained. The FLSE has one greater solution and two lower solutions which are:

X_{gr}	X_{low_1}	X_{low_2}
0.5	0.5	0
0.2	0.2	0.2
0	0	0
0.5	0	0.5
0	0	0

That means that there are two maximal interval solution:

$$X_1 = \begin{pmatrix} 0.5 \\ 0.2 \\ 0 \\ [0, 0.5] \\ 0 \end{pmatrix} \qquad X_2 = \begin{pmatrix} [0, 0.5] \\ 0.2 \\ 0 \\ 0.5 \\ 0 \end{pmatrix}$$

which are all possible values for $X_{5\times1}$ to obtain the same result $B_{3\times1}$. This basically means that either S:Poor or F:Rancid can be arbitrary number between *0* and *0.5* and this will not change the tip.

6. Handling the limitations

6.1. One output value per rule

Solving system with more than one output variables means that vector B should become matrix. This means that so called relational equations should be solved. Description for relational equations and efficient method for solving them can be found in [7].

Also, normally in FIS, for output only AND connection between different fuzzy values in different fuzzy variables are supported. So this limitation can be easy avoided if such a rules are split using that $a \Rightarrow (b \wedge c) \equiv (a \Rightarrow b) \wedge (a \Rightarrow c)$.

In other words one easy way to solve systems with many output variables is to split the system (2) into many systems. Every system should have for its right hand side one column of the matrix B.

6.2. Only OR connections

The limitation that only OR connection between terms are supported is much more complicated. It becomes because of the used composition in the system (2). With max-min composition only OR connection can be represented. It is possible to use min-max composition for AND connection and actually for any connection a convenient composition can be studied and used. Here the problem is that solving the inverse problem in different compositions can be hard task. Also FC^{2}ore can solve FLSE with six different compositions.

Using different composition for different connections unfortunately is not enough to deal with this limitation. Bigger problem here is how to manage the situation (which is very common) where in one FIS there are different rules with different connections and even different connections between the terms in same rule.

The first situation can be avoided again by splinting the system in many systems but this time all rules with same composition are separated in separated systems.

No solution for the second situation can be given by the author.

7. Conclusion

A new approach to present FIS is presented in this paper. It gives a lot of opportunities, but on the price of lot of limitations.

REFERENCES

[1] Baldwin, J.F. "Fuzzy logic and fuzzy reasoning," in Fuzzy Reasoning and Its Applications, E.H. Mamdani and B.R. Gaines (eds.), London: Academic Press, 1981.

[2] Esragh, F. and E. H. Mamdani. A general approach to linguistic approximation, Fuzzy Reasoning and Its Applications, E.H. Mamdani and B.R. Gaines (eds.), London, Academic Press, 1981.

[3] Jang, J. S. R. and N. Gulley. The Fuzzy Logic Toolbox for Use with MATLAB, Massachusetts, MathWorks Inc., 1995.

[4] Klir, G., B. Yuan. Fuzzy Sets and Fuzzy logic: Theory and Applications, Prentice-Hall, PTR, Englewood Cliffs, NJ, 1995.
[5] Mamdan,i E., S. Assilian. An experiment in linguist synthesis with fuzzy logic controller. Int J Man-Mach Stud 7(1), 1975, pp. 1-13.
[6] Peeva, K. Universal algorithm for solving fuzzy relational equations, Italian Journal of Pure and Applied Mathematics 19, 2006, pp. 9-20.
[7] Peeva, K. and Y. Kyosev. Fuzzy Relational Calculus-Theory, Applications and Software (with CD-ROM), In the series Advances in Fuzzy Systems - Applications and Theory, Vol. 22, World Scientific Publishing Company, 2004.
[8] Peeva, K., Z. Zahariev. Software for testing linear dependence in Fuzzy Algebra, in International Scientific Conference Computer Science, September 2005, Chalkidiki, Greece, pp. 294-299.
[9] Sanchez, E. Resolution of composite fuzzy relation equations, Information and Control, 30, 1976, pp. 38-48.
[10] Siler, W, Buckley JJ (2004) Fuzzy expert systems and fuzzy reasoning. Wiley InterScience.
[11] Yager, R.R., and L. A. Zadeh. An Introduction to Fuzzy Logic Applications in Intelligent Systems, Kluwer Academic Publishers, 1991.
[12] Zadeh, L.A. Making computers think like people, I.E.E.E. Spectrum, 8/1984, pp. 26-32.
[13] Zahariev, Z. Solving Max-min Relational Equations. Software and Applications, Proceedings of 34th International Conference AME, Sozopol, American Institute of Physics, 2008, pp. 516-523.
[14] Zahariev, Z. Software package and API in MATLAB for working with fuzzy algebras, In International Conference “Applications of Mathematics in Engineering and Economics (AMEE'09)”, AIP Conference Proceedings, G. Venkov, R. Kovatcheva, V. Pasheva (eds.) American Institute of Physics, ISBN 978-0-7354-0750-9, 434-350.

Received 23.02.2010 **Reviewer** Prof. DSc Keti Peeva