# Decomposition of Fuzzy Relations

Witold Pedrycz

Dept. of Electrical & Computer Engineering
University of Manitoba
Winnipeg, Manitoba, Canada R3T 5V6
pedrycz@eeserv.ee.umanitoba.ca

**Abstract** The paper introduces some optimization mechanisms for decomposition of fuzzy relations defined in finite universes of discourse. Two generic problems being posed involve a single- and multiple-step decomposition. In the first formulation a fuzzy relation R: $\mathbf{Y} \times \mathbf{X} \rightarrow [0,1]$ is decomposed into two fuzzy relations $G_2$: $\mathbf{Y}$ x $\mathbf{Z}$, $G_1$: $\mathbf{Z}$ x $\mathbf{X}$ whose max-min composition produces R, namely $R = G_2 \circ G_1$. The multi-step decomposition concerns a series of "intermediate" fuzzy relations $G_1$: $\mathbf{Z}_1 \times \mathbf{X} \rightarrow [0,1]$, $G_2$: $\mathbf{Z}_2 \times \mathbf{Z}_1 \rightarrow [0,1]$ ... $G_p$: $\mathbf{Y} \times \mathbf{Z}_p \rightarrow [0,1]$ such that $R = G_p \circ G_{p-1} \circ \ldots \circ G_1$. The pertinent optimization techniques proposed in the paper are those exploiting some standard gradient-descent optimization schemes and the mechanisms of fuzzy neurocomputations. Some fundamental links between the original decomposition problems and their neural network representation are analyzed. The detailed computational schemes are provided as well.

**Keywords**: decomposition of fuzzy relations, optimization, learning, network representation, multilevel fuzzy neural networks, fuzzy interpolation.

## 1. Introduction and Problem Formulation

The decomposition of fuzzy relations has become a significant research domain in fuzzy relation calculus. The reader may refer to [1] [3] that provides an updated coverage of the relevant material. The concept of decomposition has also been found useful in fuzzy system analysis, cf. [2 ].

In this paper we pursue a novel avenue by studying and developing optimization mechanisms for the decomposition problem. We look at the relevant network architectures and design their topologies to make them capable of representing the structure of the problem. As a prerequisite, we proceed with a brief overview of the decomposition problem and formulate its generic version along with some extensions and modifications. Let $R=[r_{ij}]$, i,j=1,2,...,n, be a binary fuzzy relation defined in $\mathbf{X} \times \mathbf{X}$, card($\mathbf{X}$)=n. The idea is to decompose R into G : $\mathbf{X} \times \mathbf{X} \rightarrow [0,1]$ such that the max-min composition of G with itself yields R,

$$R = G \circ G$$

or in other words the membership values yield the expression,

$$r_{ji} = \bigvee_{k=1}^{n} (g_{jk} \wedge g_{ki})$$

i,j=1,2,...,n. Note also that the above formulation constitutes a generalization of the well known problem studied in two-valued (Boolean) matrices, cf. [4].

The main thrust of our study is concentrated on the two different formulations substantially generalizing the above problem. We will distinguish:

(i) a single level decomposition: for the given binary fuzzy relation R defined in $\mathbf{Y} \times \mathbf{X}$ determine two fuzzy relations G and W, G: $\mathbf{Z} \times \mathbf{X} \rightarrow [0,1]$ and W: $\mathbf{Y} \times \mathbf{Z} \rightarrow [0,1]$, card($\mathbf{X}$)=n, card($\mathbf{Y}$)=m, card ($\mathbf{Z}$) = h, such that their max-min convolution W o G yields R, that is

$$R = W \circ G \tag{1}$$

i.e.,

$$r_{ji} = \bigvee_{k=1}^{h} (w_{jk} \wedge g_{ki})$$

j=1,2,...,m, i=1,2,...,n.

(ii) a multilevel decomposition: this decomposition problem involves a series of successive max-min compositions of fuzzy relations $G_1$: $\mathbf{Z}_1 \times \mathbf{X} \to [0,1]$, $G_2$: $\mathbf{Z}_2 \times \mathbf{Z}_1 \to [0,1]$ ,... $G_p$ :$\mathbf{Y} \times \mathbf{Z}_p \to [0,1]$ $card(\mathbf{Z}_1)=h_1$, $card(\mathbf{Z}_2)=h_2$,...,$card(\mathbf{Z}_p)=h_p$, where

$$R = G_p o\, G_{p-1} o \ldots o G_2 o\, G_1 \tag{2}$$

viz.

$$r_{ji} = \bigvee_{k_p=1}^{h_p} \bigvee_{k_{p-1}=1}^{h_{p-1}} \ldots \bigvee_{k_1=1}^{h_1} \left(g_{jk_p}^{(p)} \wedge g_{k_p k_{p-1}}^{(p-1)} \wedge \ldots \wedge g_{k_1 i}^{(1)}\right)$$

(the superscripts are used to designate the corresponding fuzzy relations). One can refer to (2) as a p-th order decomposition; in this terminology (1) is just a second order fuzzy relation decomposition. Note also that (2) can be made more specific by narrowing the problem down to the form in which $G_1=G_2=...=G_p$ and $\mathbf{X}=\mathbf{Y}$; in this case R is nothing but the p-th power of G.

## 2. An Optimization algorithm for the single step decomposition problem

The two different optimization approaches will be studied: the first one provides with a standard gradient-based optimization scheme- in such a way the decomposition is sought as a relevant minimization task. The second one is more structured and is aimed at building a fuzzy neural network that is functionally equivalent with the decomposition problem. In sequel the solution to the decomposition problem embedded in this framework is then achieved through the training of the resulting fuzzy neural network.

### 2.1. Decomposition-induced optimization

The decomposition of R as formulated by (1) can be realized by determining W and G through the optimization (minimization) of the following mean squared error(MSE) performance index,

$$Q = \sum_{j=1}^{m} \sum_{i=1}^{n} \left(r_{ji} - \bigvee_{k=1}^{h} (w_{jk} \wedge g_{ki})\right)^2$$

that is

$$\min_{W,G} Q$$

The straightforward iterative optimization scheme can be written down as follows,

$$w_{jk} = w_{jk} - \xi \frac{\partial Q}{\partial w_{jk}}$$

$$g_{ki} = g_{ki} - \xi \frac{\partial Q}{\partial g_{ki}}$$

i=1,2,...,n, j=1,2,...,m, k=1,2,...,h, where the updates of the fuzzy relations are realized based upon the gradient of Q. The learning rate $\xi$ is used to control the speed of changes (updates) of the membership values of W and G. As usually, its choice becomes a result of an obvious compromise between the speed of learning and stability of the overall optimization procedure. The relevant value of $\xi$ usually emerges as a result of some experiments; no "universal" value of the learning rate could be recommended. The complete formulas of the optimization procedure are included in Appendix A. They are straightforward although one should pay an extra attention to the min and max operators as these are not differentiable in a standard way and may affect the gradient descent computations. The selection of "h" can be made based upon the produced values of the performance index: too small size of the hidden layer may lead to excessively high values of Q.

## 2.2. Neural network representation of the decomposition problem

The decomposition task can be substituted by an equivalent problem that afterwards could be conveniently solved by a certain fuzzy neural network. The solution to the problem is derived through learning of this network. Before proceeding with the complete learning scheme, we need some fundamental prerequisites. In particular, as it will be further revealed, the entire algorithmic considerations are dwelled upon the input-output representation of fuzzy relations.

### 2.2.1. Input-output representation of fuzzy relations

Let us consider a fuzzy relation $R : Y \times X \rightarrow [0, 1]$. The max-min composition of $x : X \rightarrow [0, 1]$ and R yields another fuzzy set $y : Y \rightarrow [0, 1]$ such that

$$y = x \circ R \qquad (3)$$

or equivalently

$$y_j = \bigvee_{i=1}^{n} (r_{ji} \wedge x_i) = \bigvee_{i=1}^{n} (x_i \wedge r_{ji}).$$

j=1,2,...,m. Regarding **x** as the inputs (input fuzzy set) and y as the associated outputs (output fuzzy set) of the network, one can represent R as the connections of a fuzzy neural network composed of " m" OR neurons, cf. [5] [6][7]. In particular, the entry $r_{ji}$ describes the connection (weight) between the i-th input and the j-th output. The array of all the connections is equivalent with the fuzzy relation R. Especially, the j-th neuron can be concisely described in the form

$$y_j = OR(x_j , r_j) \qquad (4)$$

$j = 1,2, \ldots , m$, where $x = [x_1, x_2, \ldots x_n]$ and $r_j = [r_{j1}, r_{j2}, \ldots r_{jn}]$ (being the connections of this neuron) summarizes the j-th row of R.

One can look at this fuzzy neural network from the point of view of its inputs and outputs. The fundamental question emerges: how could one represent the network in terms of some input-output pairs of fuzzy sets $(x_k , y_k)$ $k = 1, 2 \ldots K^*$ such that this input-output representation is equivalent to the original network. The following proposition holds:

**Proposition 1** (Input-output equivalency representation). The fuzzy neural network (3) is completely described by the input-output pairs of fuzzy data of the form.

$$\begin{array}{ll} x_1 = [1\,0 \ldots 0] & y_1 = [r_{11}\ r_{21}\ \ldots\ r_{m1}] \\ x_2 = [0\,1\,0 \ldots 0] & y_2 = [r_{12}\ r_{22}\ \ldots\ r_{m2}] \\ x_j = [0 \ldots 0\,1\,0 \ldots 0] & y_j = [r_{1j}\ r_{2j}\ \ldots\ r_{mj}] \\ \quad\ \ \uparrow \text{ i-th entry} & \\ \vdots & \\ x_n = [0 \ldots 0\,1] & y_n = [r_{1n}\ r_{2n}\ \ldots\ r_{mn}] \end{array} \qquad (5)$$

(notice that $K^* = n$)

The remark to be made here is that the selection of the inputs as disjoint pairwise singletons plays a crucial role in establishing the equivalency representation.

### 2.2.2. LEARNING

Considering these input - outputs pairs as the training set, the connections of the network can be determined through its (parametric) learning cf. [ 6] [7]. It is interesting to inspect how the learning performs under the circumstances raised by the specific training family. Let us investigate this in more detail, by exploiting the update scheme used previously and currently applied in its on-line version. We get

$$r_{ji} = r_{ji} - \xi \frac{\partial Q}{\partial r_{ji}} \qquad (6)$$

$j = 1, m,\ i = 1 \ldots n$, where Q is viewed as the Euclidean distance between the required output $y_k$ and the result produced by the network, namely

$$[OR(x_k, r_1) \quad OR(x_k, r_2) \ldots OR(x_k, r_m)]$$

The following proposition holds

**Proposition 2** Consider the input-output pair (x, y) of fuzzy data, $r_{ji} = r_{ji} - \xi \frac{\partial Q}{\partial r_{ji}}$ $x \in [0,1]^n$, $y \in [0,1]$, where **x** is a fuzzy singleton, of the form $x_k = \delta_{kl} = \{ \begin{matrix} 1, \text{ if } k=l \\ 0, \text{ otherwise} \end{matrix}$. Then the optimization scheme

$$r_i = r_i - \xi \frac{\partial Q}{\partial r_i}$$

where

$$Q = \left[ y - \bigvee_{k=1}^{n} (r_k \wedge x_k) \right]^2$$

with $\xi = \frac{1}{2}$ converges within a single learning epoch producing a zero value of the performance index.

**Proposition 3** Consider the input-output pairs $(x_k, y_k)$, k=1,2,...,n where $x_k$ are pairwise disjoint fuzzy singletions, card(X)=n. Then the update scheme

$$r_i = r_i - \xi \frac{\partial Q}{\partial r_i}$$

minimizing the performance index

$$Q = \sum_{k=1}^{n} Q_k = \sum_{k=1}^{n} \left[ y_k - \bigvee_{i=1}^{n} (r_i \wedge x_{ki}) \right]^2$$

and realized with $\xi$ equal to $\frac{1}{2}$ produces Q=0 within a single training epoch ( here the epoch consists of a sequence of $x_k$'s each being shown once to the network).

The above results could be generalized as follows,

**Corollary** Consider the input-output pairs $(x_k, y_k)$, k=1,2,..,n where $x_k$ are pairwise disjoint fuzzy singletions, card(X)=n while $y \in [0,1]^m$. The update scheme

$$r_{ji} = r_{ji} - \xi \frac{\partial Q}{\partial r_{ji}}$$

i=1,2,...,n, j=1,2,...,m with $\xi = \frac{1}{2}$ produces a zero value of Q.

The original decomposition problem can be made equivalent to the learning in the network with a single hidden layer and comprising OR neurons described as

$$y_j = OR(\mathbf{z}, \mathbf{w}_j)$$
$$z_i = OR(\mathbf{x}, \mathbf{g}_i)$$

$\mathbf{z} \in [0,1]^h$, j=1,2,...,m, i=1,2,...,n. The learning is realized for the training data set given in the format specified above. It is clear that the description of the network is functionally equivalent with the decomposition problem as the fuzzy relations W and G are distributed within the layers. The parametric learning of the connections are carried out by modifying their values according to the values of the gradient of Q.

The scope of applicability of the findings of Proposition 2 and 3, even though these are very much encouraging, could not be overestimated as they pertain to the networks without hidden layer(s). When it comes to the hidden layers to be included in the architecture of the network, the general analysis and its so optimistic results are not valid. The learning process will definitely be spanned beyond several epochs

### 3. Multilevel decomposition - neural network optimization

The multilevel decomposition can conceptually be handled in the same way as the previous problem. First, the fuzzy relation R to be decomposed is represented as a pair of input-output fuzzy sets $(x_k, y_k)$,k=1,2,...n. Then the topology of the network is structured by selecting the number of the hidden layers and their dimensions. On the whole, the p-level decomposition calls for the network with "p-1" hidden layers.

Two learning scenarios can be envisioned:

(i) one phase learning. The learning completed in the network embraces all the connections. This task could be quite demanding especially for high values of "p" (as the chains of the corresponding derivatives become longer). A special attention should be devoted to avoidance of local minima during the learning process.

(ii) successive decomposition. The idea here is to decompose the fuzzy relation successively, i.e., relation by relation. A single hidden layer introduced first decomposes R into $G_p$ and $\widetilde{G}$, namely

$$R = G_p \circ \widetilde{G}$$

Afterwards $\widetilde{G}$ is decomposed into $\widetilde{G}'$ and $G_{p-1}$,

$$\widetilde{G} = G_{p-1} \circ \widetilde{G}'$$

Considering that the first phase of this decomposition is error-free (viz. R is single-step decomposable), becomes decomposed based on the training set formed out of the input-output data associated with $\widetilde{G}'$.

The entire procedure is performed iteratively until the required number of decomposition levels has been achieved. It should be underlined that in case of imperfect decomposition any further decomposition might be associated with error accumulation that becomes carried over through the input-output training data provided by this intermediate fuzzy relation.

### 4. Concluding Remarks

The decomposition problems have been formulated in the conceptual and optimization framework of fuzzy neural networks. By identifying the one-to-one correspondence between fuzzy relations and induced pairs of fuzzy sets one is able to translate the decomposition problem into the task of learning of the network.

One among potential generalizations of the discussed problem might involve the use of triangular norms instead of the previous max and min operations. This extension makes that the problem is not manageable with the scope of analytical methods; hence the optimization techniques are left as the only constructive option worth pursuing. Furthermore a dual decomposition problem based on the min-max composition could be solved in an analogous way by studying fuzzy neural networks composed of AND fuzzy neurons.

### Acknowledgment

Support from the Natural Sciences and Engineering Research Council of Canada and MICRONET is gratefully acknowledged.

### 5. References

1. A. Di Nola, S. Sessa, W. Pedrycz, E. Sanchez, Fuzzy Relational Equations and Their Applications in Knowledge Engineering, Kluwer Academic Press, Dordrecht, 1989.
2. A. Di Nola, W. Pedrycz, S. Sessa, Decomposition problem of fuzzy relations, Int. J. Gen. Systems, 10, 1985,

123-133.
3. A. Di Nola, W. Pedrycz, S. Sessa, M. Higashi, Minimal and maximal solutions of a decomposition problem of fuzzy relations, Int. J. General Systems, 11, 1985, 103-116.
4. K. H. Kim, Boolean Matrix Theory and Applications, M. Dekker, New York, Bassel, 1982.
5. W. Pedrycz, Neurocomputations in relational systems, IEEE Trans. on Pattern Analysis and Machine Intelligence, 13, 1991, 289-296.
6. W. Pedrycz, Fuzzy Control and Fuzzy Systems, 2nd extended edition, Research Studies Press/ J.Wiley, Taunton/New York, 1993.
7. W. Pedrycz, A.F. Rocha, Fuzzy set based models of neurons and knowledge-based networks, IEEE Trans. on Fuzzy Systems, 1, 1993, 254-166.

## APPENDIX A

Here we derive the formulas of the detailed optimization scheme for a single-level decomposition problem as it has been posed by (1). The objective function to be minimized is given as

$$Q = \sum_{j=1}^{m} \sum_{i=1}^{n} \left[ r_{ji} - \bigvee_{k=1}^{h} \left( w_{jk} \wedge g_{ki} \right) \right]^2$$

The updates of w are worked out as

$$w_{st} = w_{st} + 2\alpha \sum_{j=1}^{m} \sum_{i=1}^{n} \left[ r_{ji} - \bigvee_{k=1}^{h} \left( w_{jk} \wedge g_{ki} \right) \right] \frac{\partial}{\partial w_{st}} \left( \bigvee_{k=1}^{k} \left( w_{sk} \wedge g_{ki} \right) \right)$$

$s = 1, 2, ..., m$, $t = 1, 2, ..., n$. The inner derivative, as it involves the lattice (max and min) operations, should be treated carefully to prevent the learning scheme from running into one of the local minima or being eventually trapped into a nonstationary point during the search. The latter may happen due to the zeroing of the derivatives. Noting that, let us re-define these derivatives. Observe that

$$\min(a,x) = \begin{cases} x, & \text{if } x \le a \\ a, & \text{if } x > a \end{cases}$$

$$\max(a,x) = \begin{cases} x, & \text{if } x \ge a \\ a, & \text{if } x < a \end{cases}$$

This implies that the derivatives associated with these predicates produce Boolean values ({0,1}) can formally be defined as

$$\frac{\partial \min(a,x)}{\partial x} = \begin{cases} 1, & \text{if } x \le a \\ 0, & \text{if } x > a \end{cases}$$

$$\frac{\partial \max(a,x)}{\partial x} = \begin{cases} 1, & \text{if } x \ge a \\ 0, & \text{if } x < a \end{cases}$$

Put these expressions in a different way: these derivatives are just truth values of the two-valued predicates " less or equal" and "greater or equal", respectively. E.g.,

$$\frac{\partial \min(a,x)}{\partial x} = \text{truth value ( "x is less or equal to a")}$$

A natural way to make these definitions more flexible would be by relaxing the above constraints and admitting their multivalued counterparts. Thus rather than talking about the truth value of

" x less or equal to a"

we are interested in a degree of satisfaction of the inclusion relationship

"x less than a"

viz. a degree to which "x is less than a". In more detail, the degree of inclusion, $x \leq a$, can be modelled as a multivalued implication defined as $x \rightarrow a = \sup\{c \in [0,1] \mid x \; t \; c \leq a\}$ where "t" denotes a certain t-norm. In particular, the Lukasiewicz implication might be of interest as it produces a piecewise linear form of this relationship, namely

$$\| x \leq a \| = x \rightarrow a = \begin{cases} 1 - x + a, & x > a \\ 1, & x \leq a \end{cases}$$

Taking advantage of this notion we modify the computations accordingly,

$$\frac{\partial}{\partial w_{st}} \left( \bigvee_{k=1}^{h} (w_{sk} \wedge g_{ki}) \right) = \| \bigvee_{\substack{k \neq t}}^{h} (w_{sk} \wedge g_{ki}) \leq ( w_{st} \wedge g_{ti} \| \quad \| w_{st} \leq g_{ti} \|$$

Similarly, the connections g are subject to the following updates:

$$g_{st} = g_{st} + \alpha \sum_{j=1}^{m} \left[ r_{jt} - \bigvee_{k=1}^{h} (w_{jk} \wedge g_{kt}) \right] \frac{\partial}{\partial g_{st}} \left( \bigvee_{k=1}^{h} (w_{jk} \wedge g_{kt}) \right)$$

and

$$\frac{\partial}{\partial g_{st}} \left( \bigvee_{k=1}^{h} (w_{jk} \wedge g_{kt}) \right) = \| \bigvee_{\substack{k \neq s}}^{h} (w_{jk} \wedge g_{kt}) \leq (w_{js} \wedge g_{st}) \| \quad \| g_{st} \leq w_{js} \|$$