

An Algorithm to Compute the Transitive Closure, a Transitive Approximation and a Transitive Opening of a Fuzzy Proximity

Luis Garmendia¹, Ramón González del Campo¹,
Victoria López¹, Jordi Recasens²

¹ Universidad Complutense de Madrid, Spain

² Universitat Politècnica de Catalunya, Spain

lgarmend@fdi.ucm.es, rgonzale@estad.ucm.es,

vlopez@fdi.ucm.es, j.recasens@upc.edu,

Abstract

A method to compute the transitive closure, a transitive opening and a transitive approximation of a reflexive and symmetric fuzzy relation is given. Other previous methods in literature compute just the transitive closure, some transitive approximations or some transitive openings. The proposed algorithm computes the three different similarities that approximate a proximity for the computational cost of computing just one. The shape of the binary partition tree for the three output similarities are the same.

Keywords: Partition tree; transitive closure; transitive opening; transitive approximation; fuzzy similarity; fuzzy proximity.

1 Introduction

The transitivity property of fuzzy relations is quite often violated in some application, even by decision makers that accept transitivity as a condition for consistency. A standard approach to deal with intransitive fuzzy relations is the search for close enough transitive fuzzy relations, assuming that such a violation is mainly due to decision maker estimation errors.

Analogously to transitivity of crisp relations, the transitive property of fuzzy relations can be understood as a threshold on the degree of the relation (for example, a degree of equality) between two elements, when a degree of relation between those elements and a third element of a universe of discourse is known. The classical concept of transitivity is generalized in fuzzy logic by the T-transitivity property of fuzzy relations, where T is a triangular norm [SCHWEIZER AND SKLAR; 1984]. A very

important family of fuzzy relations are T-indistinguishabilities (reflexive, symmetric and T-transitive fuzzy relations) since they generalize the concepts of (crisp) equivalence relation and equality [TRILLAS E. AND VALVERDE, 1984] and are useful to represent the ideas of similarity and neighbourhood as well. From now on Min-transitivity will be called simply transitivity.

Algorithms that search for the transitivity property can be used in many approximate reasoning applications, including database management systems (DBMS), pattern recognition, expert systems, artificial intelligence (AI), and intelligent systems.

Sometimes our knowledge is modeled in a reflexive and symmetric fuzzy relation R , and we want to compute a transitive fuzzy relation from that knowledge to impose some coherence or to generate a similarity.

The transitive closure [MEYER H., NAESSENS H., DE BAETS B., 2004] has some advantages with respect to other transitive approximations, but suffers from the chaining effect: two objects almost unrelated by the original relation can become very related in the T-transitive closure.

Other methods calculate transitive openings, as complete linkage or growtree methods [DAWYNDT, DE MEYER, DE BAETS; 2003], which are similarities smaller than R but maximal among all similarities smaller than R . We just know a few methods to compute a few T-transitive openings or approximations of proximity [DE BAETS, B., DE MEYER, H., 2003; GARMENDIA, SALVADOR, MONTERO, 2009]. The T-transitive closure of a fuzzy relation exists, and it is unique, however there are many transitive openings of a fuzzy relation. There is still a very interesting open problem about how to find all the T-transitive openings of a given fuzzy proximity. Another open problem is the computation of transitive approximations, which are not comparable with R in the sense that some of the entries are greater while some other entries are smaller than the corresponding entries of R . It is obvious the importance of those methods if we must replace a given fuzzy proximity by a transitive one. In most occasions there exist transitive approximations that are closer (using a distance between fuzzy relations) to a given relation than its corresponding transitive closure or its transitive openings [GARMENDIA L. AND RECASENS J., 2009]. In this paper a simple algorithm to produce three different similarities (and therefore indexed hierarchical trees) from a fuzzy proximity relation R is given. The most interesting feature is that the same algorithm computes the transitive closure, a T-transitive opening and a transitive approximation of a proximity at the same time.

The given algorithm generates different partition trees and transitive approximations than the complete linkage clustering method and the Growtree algorithm [DAWYNDT, DE MEYER, DE BAETS; 2003].

After a section of preliminaries, section 3 provides the algorithm and some examples. It is also proved that the algorithm outputs are similarities.

2 Preliminaries

This section contains some definitions and properties of similarities and some previous methods to generate similarities from fuzzy proximity relations.

Definition 1. [SCHWEIZER, SKLAR; 1984] A binary operator $T: [0, 1] \times [0, 1] \rightarrow [0, 1]$ is a **t-norm** if it satisfies the following axioms:

$$T(1, x) = x$$

$$T(x, y) = T(y, x)$$

$$T(x, T(y, z)) = T(T(x, y), z)$$

If $x \leq x'$ and $y \leq y'$ then $T(x, y) \leq T(x', y')$.

Definition 2. Let $E = \{e_1, \dots, e_n\}$ be a finite set. A fuzzy relation R on E is a map $R: E \times E \rightarrow [0, 1]$. The relation degree value for elements e_i and e_j in E is called e_{ij} . So $e_{ij} = R(e_i, e_j)$.

A fuzzy relation R is **reflexive** if $e_{ii} = 1$ for all $1 \leq i \leq n$.

A fuzzy relation R is **α -reflexive** if $e_{ii} \geq \alpha$ for all $1 \leq i \leq n$.

The relation R is **symmetric** if $e_{ij} = e_{ji}$ for all $1 \leq i, j \leq n$.

A reflexive and symmetric fuzzy relation is called a fuzzy **proximity** relation.

Definition 3. Let T be a triangular norm. A fuzzy relation $R: E \times E \rightarrow [0, 1]$ is **T-transitive** if and only if $T(R(a, b), R(b, c)) \leq R(a, c)$ for all a, b, c in E . So R is T-transitive if $T(e_{ik}, e_{kj}) \leq e_{ij}$ for all $1 \leq i, j, k \leq n$.

Definition 4. [ZADEH 1971] A fuzzy **similarity** is a reflexive, symmetric and min-transitive fuzzy relation.

A **T-indistinguishability** is a reflexive, symmetric and T-transitive fuzzy relation.

Definition 5. The relation A includes the relation B ($A \supseteq B$) if $a_{ij} \geq b_{ij}$ for all $1 \leq i, j \leq n$.

Definition 6. [BANDLER AND KOHOUT, 1988] Let P be a property of fuzzy relations on a universe E . A fuzzy relation R^P is called the P-closure of a fuzzy relation R if:

1) R^P has property P

2) $R \subseteq R^P$

3) If $R \subseteq R'$ and R' has property P then $R^P \subseteq R'$

Theorem 1. [BANDLER AND KOHOUT, 1988] The intersection of any non-empty family of fuzzy relations on E verifying property P , also has property P .

The transitive closure of a crisp relation with respect to a given property has been generalised in the following way:

Proposition 1. *Given a t-norm T and a fuzzy relation R on finite universe, there exists a unique fuzzy T -transitive relation A that includes R , and if a fuzzy T -transitive relation includes R then it also includes A .*

Such T -transitive fuzzy relation is called the **T-transitive closure** of R , and it is the lowest T -transitive fuzzy relation that contains R .

The min-transitive closure of a fuzzy relation is simply called its **transitive closure**. Some known methods to compute the transitive closure of fuzzy relations are given by DUNN [1974] using the Prim maximum weight spanning tree algorithm; by KUNDU [2000], rewriting the Dunn algorithm and obtaining the partition tree in $O(n^2)$ time; by KANDEL AND YELOWITZ [1974], which is a modification of the Floyd algorithm to find the shortest paths between nodes in classical weighed directed graphs; by LARSEN AND YAGER [1990], creating a binary tree representation of a fuzzy similarity; by GUOYAO FU [1992] using an ascending value method; and finally by LEE [2001] constructing the LARSEN AND YAGER [1989] binary tree representation of the transitive closure of a reflexive and symmetric fuzzy relation.

Theorem 2. *Consider an arbitrary universe E and an arbitrary t-norm T . Then any fuzzy relation R on E has a T -transitive closure.*

The T -transitive closure of a symmetric fuzzy relation is also symmetric. Also reflexivity and α -reflexivity are preserved by the T -transitive closure. In particular, the T -transitivity closure of a fuzzy proximity is a T -indistinguishability.

There are several methods to compute the transitive closure of a fuzzy proximity. Let us recall the most important ones.

A matrix construction method to compute the T-transitive closure

Definition 7. *The sup-T composition $Ro_{\text{sup-T}}S$ of two relations R and S on a universe E is defined for all $x, y \in E$ by $Ro_{\text{sup-T}}S(x, z) = \text{Sup}_{y \in E} \{T(R(x, y), S(y, z))\}$.*

If T is a continuous t-norm, then the sup-T product is associative and the powers R^k of R to any positive integer k can be defined recursively by $R^1=R$ and $R^k = Ro_{\text{sup-T}} R^{k-1}$.

A general algorithm used to compute the transitive closure of a fuzzy proximity R on a finite universe E of cardinality n , known as the matrix method, is the calculation of

$$R^T = \bigcup_{k=1}^{n-1} R^k .$$

It carries out $O(n)$ compositions, and one matrix sup-T-composition is $O(n^3)$ time. So the general matrix algorithm demands $O(n^4)$ time complexity in the worst case.

A version of this algorithm to compute the T -transitive closure of a fuzzy relation R is:

- 1) While $R' \neq R$ do $R' = R \cup_{\text{Max}} (Ro_{\text{sup-T}}R)$
- 2) $R^T := R'$.

In the worst case, $O(\log n)$ matrix compositions are required, so this method takes $O(n^3 \log n)$ time complexity in the worst case and takes $O(n^2)$ space complexity.

The Floyd-Warshall method to compute the T-transitive closure

Let R be a fuzzy relation on a finite universe E of dimension n , and let T be a continuous t-norm. Its T-transitive closure is computed as follows:

```
for (int i=1; i<n; i++)
    for (int j=1; j<n; j++)
        for (int k=1; k<n; k++)  $R_{jk} = \max(R_{jk}, T(e_{ji}, e_{ik}))$ .
```

This method takes $O(n^3)$ time complexity and $O(n^2)$ space complexity.

[NAESSENS, DE MEYER, AND DE BAETS; 2002] propose a similar method, but reversing the loop structure.

T-transitive openings

The P-closure of a fuzzy relation R can be regarded as the smallest fuzzy relation having property P and dominating R (see definition 6). A P-opening of R should be defined symmetrically as the largest fuzzy relation having property P and being dominated by R . Nevertheless, in the case of the T-transitive property, there exist a unique T-transitive closure of a fuzzy relation, but there are several T-transitive openings, because there are several maximal T-transitive relations contained in R .

Note that a fuzzy relation can have an infinite number of transitive openings, even on a finite universe. For example, the fuzzy relation given by the matrix

$$R = \begin{pmatrix} 1 & 0.9 & 0.7 \\ 0.9 & 1 & 0.2 \\ 0.7 & 0.2 & 1 \end{pmatrix} \text{ has two different transitive openings } \begin{pmatrix} 1 & 0.9 & 0.2 \\ 0.9 & 1 & 0.2 \\ 0.2 & 0.2 & 1 \end{pmatrix}, \text{ and}$$

$$\begin{pmatrix} 1 & 0.2 & 0.7 \\ 0.2 & 1 & 0.2 \\ 0.7 & 0.2 & 1 \end{pmatrix} \text{ that are contained in } R, \text{ but that are not comparable by the inclusion}$$

relation. Of course, the transitive closure of R is uniquely defined

$$\text{by } \begin{pmatrix} 1 & 0.9 & 0.7 \\ 0.9 & 1 & 0.7 \\ 0.7 & 0.7 & 1 \end{pmatrix}. \text{ There are also several transitive approximations that cannot be}$$

compared with R by the set inclusion \subseteq . Some examples of those transitive

$$\text{approximations of } R \text{ are } \begin{pmatrix} 1 & 0.9 & 0.4 \\ 0.9 & 1 & 0.4 \\ 0.4 & 0.4 & 1 \end{pmatrix} \text{ or } \begin{pmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{pmatrix}. \text{ These relations can}$$

be even ‘closer’ transitive fuzzy relations to R than the transitive closure and openings, where ‘close’ is referred to a given distance or similarity measure for fuzzy relations.

Lemma 1. Let h be a fuzzy subset of E . The fuzzy relation R_h defined by all $x, y \in E$ by $R_h(x, y) = T(\overrightarrow{T}(h(x) | h(y)), \overrightarrow{T}(h(y) | h(x)))$ is a T -indistinguishability, where T is the residuated operator of T, \cdot , defined by $\overrightarrow{T}(x, y) = \{\sup z / T(x, z) \leq y\}$.

Theorem 3. (Representation Theorem) [VALVERDE 1985] Let R be a fuzzy relation on a universe E and $\{h_i\}_{i \in I}$ a family of fuzzy subsets of E . R is a T -indistinguishability if and only if for all $x, y \in E$ $R(x, y) = \inf_{i \in I} R_{h_i}(x, y)$.

Corollary 1. Let R be a fuzzy proximity on a finite universe E of cardinality n and $\{h_x = R(x, \cdot)\}_{x \in E}$ the columns of R . The fuzzy relation I defined for all $x, y \in E$ by $I(x, y) = \inf_{x \in E} R_{h_x}(x, y)$ is a T -indistinguishability operator smaller or equal than R .

3 Algorithm to Compute the Transitive Closure, a Transitive Opening and a Transitive Approximation of a Fuzzy Proximity.

This section presents an algorithm that computes the transitive closure, a transitive opening and several other approximations of a give fuzzy proximity R .

ALGORITHM

Let R be a fuzzy proximity relation on a universe $E = \{e_1, \dots, e_n\}$ with values $e_{ij} = R(e_i, e_j)$. Let us call node to a subset of E (a node is an element of $\wp(E)$). In order to make an easier notation, we consider the elements of E by their natural number of their position.

Input: a proximity R

Output: the transitive closure $A = [a_{ij}]$, a transitive opening $B = [b_{ij}]$ and a transitive approximation $C = [c_{ij}]$ of R .

The algorithm is the following:

- 1) Create a set of nodes N initially with a set of singletons $N_i = \{e_i\}$ for each element e_i in E .
- 2) Set $a_{ii}=1$, $b_{ii}=1$, and $c_{ii}=1$ for all i from 1 to n .
- 3) $n-1$ times (while N is not the universe E) {

$$\text{Compute } m(N_i, N_j) = \max_{i \in N_i, j \in N_j} e_{i,j} \text{ for all pair of nodes } N_i \times N_j$$

with $i \neq j$.

Record (i, j) where $m(N_i, N_j)$ is maximal.

Assign $a_{rs} = a_{sr} := \max_{i \in N_i, j \in N_j} e_{i,j}$ for all $r \in N_i$ and $s \in N_j$.

Assign $b_{rs} = b_{sr} := \min(\min_{i \in N_i, j \in N_j} e_{i,j}, \min_{k,l \in N_i} b_{k,l}, \min_{k,l \in N_j} b_{k,l})$ for all $r \in N_i$
 and $s \in N_j$.
 Assign $c_{rs} = c_{sr} := \min(\text{avg}_{i \in N_i, j \in N_j} e_{i,j}, \min_{k,l \in N_i} c_{k,l}, \min_{k,l \in N_j} c_{k,l})$ for all $r \in N_i$
 and $s \in N_j$, where *avg* is an aggregation operator, for
 example, the arithmetic mean.
 Delete nodes N_i and N_j from N .
 Insert $N_i \cup N_j$ into N .
 }

The algorithm takes just $n-1$ steps, where n is the cardinality of the universe E . It takes $O(n^2)$ space complexity and $O(n^2 \log n)$ average time complexity. Note that the complete linkage clustering algorithm computes $m(N_i, N_j) = \min_{i \in N_i, j \in N_j} e_{i,j}$. The proposed algorithm computes $m(N_i, N_j) = \max_{i \in N_i, j \in N_j} e_{i,j}$ in the first step of the loop, reaching a completely different binary partition tree.

Example 1.

Let R be the fuzzy proximity on a universe $E = \{e_1, \dots, e_6\}$ given by the following matrix:

$$R = \begin{pmatrix} 1 & 1 & 0.5 & 0.3 & 0.2 & 0.3 \\ 1 & 1 & 0.8 & 0.2 & 0.4 & 0.3 \\ 0.5 & 0.8 & 1 & 0.9 & 0.3 & 0.3 \\ 0.3 & 0.2 & 0.9 & 1 & 0.8 & 0.1 \\ 0.2 & 0.3 & 0.3 & 0.8 & 1 & 0.5 \\ 0.3 & 0.2 & 0.3 & 0.1 & 0.5 & 1 \end{pmatrix}$$

The first two loops of part 3) of the algorithm records $m(N_1, N_2) = 1$ and $m(N_3, N_4) = 0.9$.

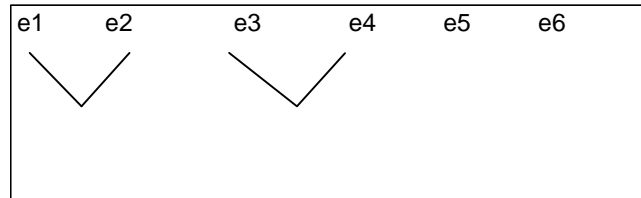


Figure 1: Example 1, steps 1 and 2

The matrices A, B and C have now values

$$\begin{pmatrix} 1 & 1 & & & & \\ 1 & 1 & & & & \\ & & 1 & 0.9 & & \\ & & 0.9 & 1 & & \\ & & & & 1 & \\ & & & & & 1 \end{pmatrix}$$

In the third loop a maximal value it is found with $m(N_3 \cup N_4, N_5) = 0.8$, and the build partition tree is generated as shown in Figure 2.

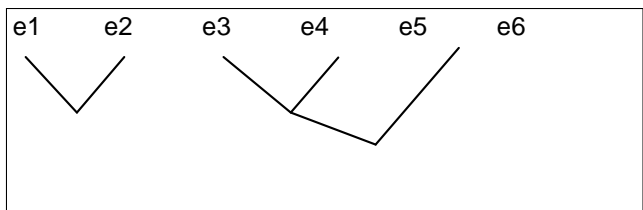


Figure 2: generated binary partition tree in loop 3.

The matrix construction of the transitive closure A, transitive opening B and transitive approximation C is in this step as follows

$$A = \begin{pmatrix} 1 & 1 & & & & \\ 1 & 1 & & & & \\ & & 1 & 0.9 & 0.8 & \\ & & 0.9 & 1 & 0.8 & \\ & & 0.8 & 0.8 & 1 & \\ & & & & & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} 1 & 1 & & & & \\ 1 & 1 & & & & \\ & & 1 & 0.9 & 0.3 & \\ & & 0.9 & 1 & 0.3 & \\ & & 0.3 & 0.3 & 1 & \\ & & & & & 1 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 1 & & & & \\ 1 & 1 & & & & \\ & & 1 & 0.9 & 0.55 & \\ & & 0.9 & 1 & 0.55 & \\ & & 0.55 & 0.55 & 1 & \\ & & & & & 1 \end{pmatrix}$$

Figure 3: Transitive closure A, transitive opening B and transitive approximation C generated in step 3.

In one more step, $N_1 \cup N_2$ and $N_3 \cup N_4 \cup N_5$ are joined, and the generated binary partition tree is

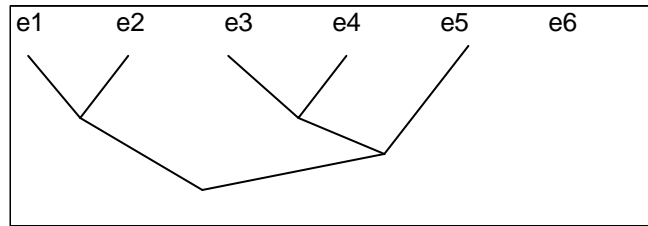


Figure 4: Generated binary partition tree in loop 4.

The matrix construction of the transitive closure A, transitive opening B and transitive approximation C of R is in this step as follows.

$$A = \begin{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} & 0.8 & 0.8 & 0.8 \\ 0.8 & 0.8 & \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix} & 0.8 \\ 0.8 & 0.8 & \begin{pmatrix} 0.9 & 1 \\ 0.8 & 0.8 \end{pmatrix} & 1 \\ 0.8 & 0.8 & \begin{pmatrix} 0.8 & 0.8 & 1 \end{pmatrix} & \end{pmatrix} 1$$

$$B = \begin{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix} & 0.3 \\ 0.2 & 0.2 & \begin{pmatrix} 0.9 & 1 \\ 0.3 & 0.3 \end{pmatrix} & 1 \\ 0.2 & 0.2 & \begin{pmatrix} 0.3 & 0.3 & 1 \end{pmatrix} & \end{pmatrix} 1$$

$$C = \begin{pmatrix} 1 & 1 & 0.38 & 0.38 & 0.38 \\ 1 & 1 & 0.38 & 0.38 & 0.38 \\ 0.38 & 0.38 & 1 & 0.9 & 0.55 \\ 0.38 & 0.38 & 0.9 & 1 & 0.55 \\ 0.38 & 0.38 & 0.55 & 0.55 & 1 \end{pmatrix}$$

Figure 5: Transitive closure A, transitive opening B and transitive approximation C in loop 4.

Finally, the last node is linked in loop 5. Note that there are only five (n-1) loops because the universe E has 6 elements.

The transitive closure of R is then $A = \begin{pmatrix} 1 & 1 & 0.8 & 0.8 & 0.8 & 0.5 \\ 1 & 1 & 0.8 & 0.8 & 0.8 & 0.5 \\ 0.8 & 0.8 & 1 & 0.9 & 0.8 & 0.5 \\ 0.8 & 0.8 & 0.9 & 1 & 0.8 & 0.5 \\ 0.8 & 0.8 & 0.8 & 0.8 & 1 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 1 \end{pmatrix}$.

And the built binary weighted tree for the transitive closure of the fuzzy proximity (which is always a fuzzy similarity) is the following:

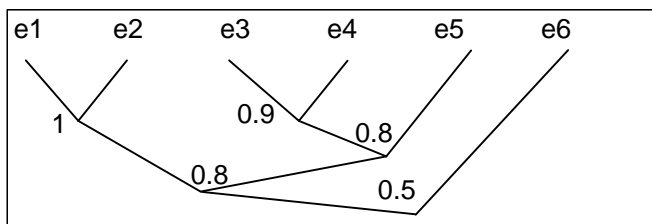


Figure 6: Binary weighted tree for the transitive closure A of the fuzzy proximity

The computed transitive opening of the fuzzy proximity R, and its binary weighted tree are the following:

$$B = \begin{pmatrix} \left(\begin{pmatrix} 1 & 1 \end{pmatrix} \right) & 0.2 & 0.2 & 0.2 & & 0.1 \\ \left(\begin{pmatrix} 1 & 1 \end{pmatrix} \right) & 0.2 & 0.2 & 0.2 & & 0.1 \\ 0.2 & 0.2 & \left(\begin{pmatrix} 1 & 0.9 \end{pmatrix} \right) & 0.3 & & 0.1 \\ 0.2 & 0.2 & \left(\begin{pmatrix} 0.9 & 1 \end{pmatrix} \right) & 0.3 & & 0.1 \\ \left(0.2 & 0.2 \right) & \left(\begin{pmatrix} 0.3 & 0.3 & 1 \end{pmatrix} \right) & & & & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 1 \end{pmatrix}$$

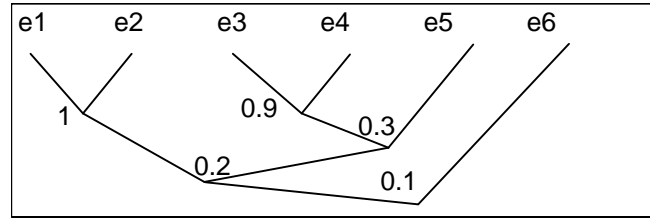


Figure 7: transitive opening B of the fuzzy proximity R, and its binary weighted tree

And finally, the transitive approximation using the arithmetic mean for both cases, which is also not comparable, but it is closer to the original fuzzy proximity R using any distance, is the following:

$$C = \begin{pmatrix} 1 & 1 & 0.38 & 0.38 & 0.38 & 0.28 \\ 1 & 1 & 0.38 & 0.38 & 0.38 & 0.28 \\ 0.38 & 0.38 & 1 & 0.9 & 0.55 & 0.28 \\ 0.38 & 0.38 & 0.9 & 1 & 0.55 & 0.28 \\ 0.38 & 0.38 & 0.55 & 0.55 & 1 & 0.28 \\ 0.28 & 0.28 & 0.28 & 0.28 & 0.28 & 1 \end{pmatrix}$$

Figure 8: Transitive approximation of R computed by the algorithm 1.

Remark 1. The complete linkage clustering method computes different transitive opening than the opening B computed by the algorithm 1.

The complete linkage clustering computes the following matrix and binary tree for the proximity R of example 3 which is not comparable with the opening computed by algorithm 1.

$$R_c = \begin{pmatrix} \left(\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \right) & 0.2 & 0.2 & & 0.1 & 0.1 \\ \left(\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \right) & 0.2 & 0.2 & & 0.1 & 0.1 \\ 0.2 & 0.2 & \left(\begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix} \right) & & 0.1 & 0.1 \\ 0.2 & 0.2 & \left(\begin{pmatrix} 0.9 & 1 \\ 1 & 0.9 \end{pmatrix} \right) & & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & \left(\begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \right) \\ 0.1 & 0.1 & 0.1 & 0.1 & \left(\begin{pmatrix} 0.5 & 1 \\ 1 & 0.5 \end{pmatrix} \right) \end{pmatrix}$$

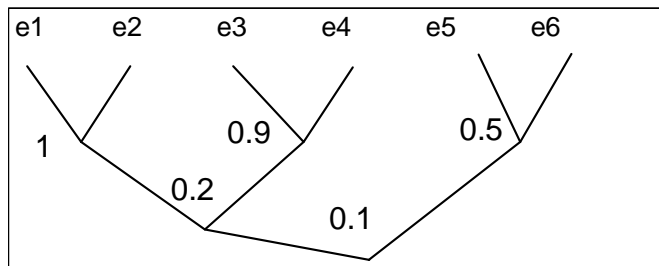


Figure 9: Transitive opening generated from R by the complete linkage clustering method.

Remark 2. The Growtree [DAWYNDT, DE MEYER, DE BAETS, 2003] clustering method compute different transitive opening than the opening B computed by algorithm 1.

The complete the Growtree method computes the following matrix and binary tree for the proximity R of example 3 which is the following:

$$R_c = \begin{pmatrix} \left(\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \right) & 0.5 & 0.2 & 0.2 & 0.1 \\ \left(\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \right) & 0.5 & 0.2 & 0.2 & 0.1 \\ \left(\begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} \right) & 1 & 0.2 & 0.2 & 0.1 \\ 0.2 & 0.2 & 0.2 & \left(\begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix} \right) & 0.1 \\ 0.2 & 0.2 & 0.2 & \left(\begin{pmatrix} 0.8 & 1 \\ 1 & 0.8 \end{pmatrix} \right) & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 1 \end{pmatrix}$$

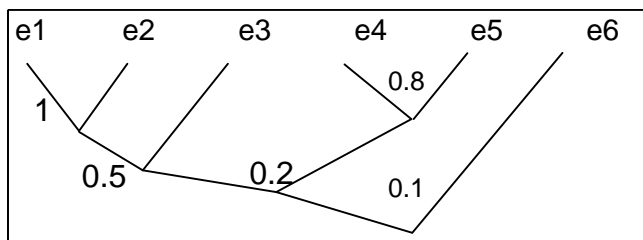


Figure 10: Transitive opening generated from R by the Growtree clustering method.

The correctness of the algorithm is based on the next lemma.

Lemma 2. [LEE 2001] Let C and D be two fuzzy relations and

$$E(f; C, D) = \begin{pmatrix} \boxed{C} & \boxed{F^T} \\ \boxed{F} & \boxed{D} \end{pmatrix} \text{ where all values in the box } F \text{ are } f.$$

If C and D are fuzzy similarities, then $E(f; C, D) = E$ is also a fuzzy similarity, $\forall f \in [0, \min(\min(C), \min(D))]$.

Theorem 3. *The matrix $A = [a_{ij}]$, $B = [b_{ij}]$ and $C = [c_{ij}]$ computed by algorithm 1 are similarities.*

Proof: In step 2, it is set $a_{ii}=1$, $b_{ii}=1$, and $c_{ii}= 1$ for all i from 1 to n , so A , B , and C are reflexive.

Trivially A , B and C are symmetric by the construction method.

To prove that A , B and C are transitive, consider Z to be any of those 3 matrices, but with just the generated values in every loop of the algorithm.

In the first loop, it is recorded (i, j) where $m(N_i, N_j) = \max_{i \neq j} e_{i,j}$ so $z_{ij} = z_{ji} = \max_{i \neq j} e_{i,j}$.

Z is transitive because trivially $\text{Min}(z_{rs}, z_{st}) \leq z_{rt}$ for all generated values (by now the diagonal, z_{ij} and z_{ji}).

Suppose that the generated matrix Z in the loop n of step 3 is transitive, then let us show that the matrix Z generated in the loop $n+1$ of step 3 of the algorithm is also T-transitive:

In loop $n+1$ let (i, j) be the recorded values that makes $m(N_i, N_j) = \max_{i \neq j} e_{i,j}$ maximal.

There exists three possible cases.

Case 1: N_i and N_j are singletons, then $z_{ij} = z_{ji} = e_{ij}$. Z has no generated values in z_{ik} and z_{kj} with $k=1..n$ except for $k=i$ and $k=j$ (the diagonal values in the same row and column). Trivially Z is transitive.

Case 2: Either N_i or N_j is a singleton (just one of them). Suppose $N_j = \{e_j\}$ and let H be the similarity Z of loop n restricted to the set N_i . We have supposed then that H is a similarity in loop n . Then Z in loop $n+1$ has the new values f as follows:

$$Z = \begin{pmatrix} & & & f \\ & (H) & & \dots \\ & & & f \\ f & \dots & f & 1 \end{pmatrix}, \text{ where}$$

$$f = \begin{cases} \max_{i \in N_i, j \in N_j} e_{i,j} \text{ in matrix } A \\ \min(\min_{i \in N_i, j \in N_j} e_{i,j}, \min_{k,l \in N_i} b_{k,l}, \min_{k,l \in N_{ji}} b_{k,l}) \text{ in matrix } B \\ \min(\text{avg}_{i \in N_i, j \in N_j} e_{i,j}, \min_{k,l \in N_i} c_{k,l}, \min_{k,l \in N_{ji}} c_{k,l}) \text{ in matrix } C \end{cases}$$

So, in the three matrixes $f \in [0, \min(\min(H), 1)]$, and then by lemma 2, Z is a similarity in loop $n+1$.

Case 3: Neither N_i nor N_j is a singleton. Let C be the similarity Z of loop n restricted to the set N_i and let D be the similarity Z of loop n restricted to the set N_j . Then Z in loop $n+1$ has the new values z as follows:

$$Z = \begin{pmatrix} \boxed{C} & \boxed{F^T} \\ \boxed{F} & \boxed{D} \end{pmatrix}, \text{ where } F \text{ has the following constant values}$$

$$f = \begin{cases} \max_{i \in N_i, j \in N_j} e_{i,j} & \text{in matrix } A \\ \min(\min_{i \in N_i, j \in N_j} e_{i,j}, \min_{k,l \in N_i} b_{k,l}, \min_{k,l \in N_j} b_{k,l}) & \text{in matrix } B \\ \min(\text{avg}_{i \in N_i, j \in N_j} e_{i,j}, \min_{k,l \in N_i} c_{k,l}, \min_{k,l \in N_j} c_{k,l}) & \text{in matrix } C \end{cases}$$

So, in the three matrixes $f \in [0, \min(\min(C), \min(D))]$ the positions of the highest values of e_{ij} has already been generated in previous loops and then by lemma 2, Z is a similarity in loop $n+1$. \square

Proposition 2. *The matrix A of the algorithm 1 is the transitive closure of the proximity R .*

Proof: By theorem 3, A is a similarity. To show that it is the transitive closure, it must be shown that A is the smallest similarity that contains R

Suppose that there exists a similarity H with $R \subset H \subset A$. Then there exist a value h_{ij} such that $e_{ij} < h_{ij} < a_{ij}$. (1)

Let (i, j) be the highest value of H such that $e_{ij} < h_{ij} < a_{ij}$ (so all the values of A generated in the algorithm before a_{ij} are equal to H).

If a_{ij} has been generated in a loop where N_i and N_j are singletons, then $a_{ij} = a_{ji} := e_{ij}$, which is in contradiction with (1).

In other case, a_{ij} has been generated in a loop where $m(N_i, N_j) = \max_{i \in N_i, j \in N_j} e_{i,j}$ for all

pair of nodes $N \times N$ with $i \neq j$, and $a_{ij} = \max_{i \in N_i, j \in N_j} e_{i,j}$ for all $r \in N_i$ and $s \in N_j$.

By lemma 2 all values of H and A in the position on $N_i \times N_j$ must be equal because H and A are similarities.

Then if $h_{ij} < a_{ij}$, then $h_{ij} < e_{ij}$, and then R is not included in H because $a_{ij} = e_{ij}$, which is a contradiction. And if $e_{ij} < h_{ij}$ then, as $a_{ij} = e_{ij}$, then H is not included in R which is also a contradiction. \square

In a similar way it can be proved that B is a transitive opening.

4 Conclusions

A method to get the transitive closure, a transitive opening and a transitive approximation of a reflexive and symmetric fuzzy relation is given. The binary partition trees of the three output similarities have the same shape.

The transitive approximations given by the algorithm are different than other clustering methods in the literature. Some examples are provided.

It is proved that the output of the algorithm are similarities, and that it computes the transitive closure, a transitive opening, and a ‘closer’ transitive approximation (using a distance between fuzzy relations) of a proximity.

The given algorithm generates all these kind of transitive approximations of a proximity at the same time, so it provides a good tool to solve the problem of approximating fuzzy proximities with similarities, since the user can choose which kind of transitive approximation (closure, opening or just the closest approximation)

fits better in different applications with the computational price of computing just one.

Acknowledgements

Research partially supported by DGICYT projects number TIN2006-14311, TIN2009-07901 and CAM GR58/08

References

- [1] Alsina, C., Trillas, E., Valverde, L. On some logical connectives for fuzzy set theory. *J. Math. Ann. Appl.*, **93** (1983) 15-26.
- [2] Bandler, W. Kohout, J. Special properties, closures and interiors of crisp and fuzzy relations. *Fuzzy Sets and Systems*, **26** (1988) 317-331.
- [3] Boixader, D., On the relationship between T-transitivity and approximate equality. *Fuzzy Sets and Systems*, **33** (2003) 6 – 69.
- [4] Dawyndt, P., De Meyer, H., De Baets, B., The complete linkage clustering algorithm revisited. *Soft Computing*, **9** (2005) 385-392.
- [5] De Baets, B., De Meyer, H., On the existence and construction of T-transitive closures. *Information Sciences*, **152** (2003) 167-179.
- [6] De Baets, B., De Meyer, H., Transitive approximation of fuzzy relations by alternating closures and openings, *Soft Computing* **7** (2003) 210-219.
- [7] De Meyer H., Naessens H., De Baets B., Algorithms for computing the min-transitive closure and associated partition tree of a symmetric fuzzy relation. *European Journal Oper. Res.* **155** (2004) 226-238.
- [8] Dijkstra, E., W., A note on two problems in connexion with graphs. *Numerische Mathematik*, **1** (1959) 269-271.
- [9] Di Nola, A., Kolodziejczyk, W., Sessa, S. Transitive solutions of relational equations on finite sets and linear lattices. *Lecture Notes in Computer Science*, **521** (1991) 173-182.
- [10] Dubois, D., Prade, H., *Fuzzy Sets and Systems: Theory and Applications* Academic Press, New York. 1980.
- [11] Dunn, J. C. Graph-theoretic analysis of pattern classification via Tamura's fuzzy relation. *IEEE Trans. Systems Man Cybernet* **4** (1974) 0–3 3.
- [12] Dunn, J. C. Some recent investigations of a new fuzzy partitioning algorithm and its application to pattern classification problems. *J. Cybernet.* (1974) 4– 5.
- [13] Elorza, J., Burillo, P, On the relation of fuzzy preorders and fuzzy consequence operators, *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems* **7** (1999) 219 - 234 .
- [14] Esteva, F, Garcia, P., Godo, L., Rodriguez, R. O., Fuzzy approximation relations, modal structures and possibilistic logic, *Mathware and Soft Computing* **5** (1998) 151-166.
- [15] Floyd, R., W., Algorithm 245 (shortest path). *Communications of the ACM*, **5** (6): 345. (1962).

- [16] Ford, L. R., Fulkerson D. R., Flows in Networks. Princeton University Press. 1962.
- [17] Garmendia, L., Campo, C., Cubillo, S., Salvador, A. A Method to Make Some Fuzzy Relations T-Transitive. *International Journal of Intelligence Systems* **14** (1999) 873 – 882.
- [17] Garmendia, L, Recasens, J. How to make T-transitive a Proximity relation. *IEEE Transactions on Fuzzy Systems* **17** (2009) 200-207
- [18] Garmendia, L., Salvador, A., Montero, J. Computing a T-transitive lower approximation or opening of a proximity relation. *Fuzzy Sets and Systems* **160** (2009) 2097-2105 .
- [19] Graham R. L., An efficient algorithm for determining the convex Hull of a finite planar set. *Information Processing Letters*, **1** (1972) 132-133..
- [20] Guoyao Fu, An algorithm for computing the transitive closure of a fuzzy similarity matrix. *Fuzzy Sets and Systems* **5** (1992) 89– 94.
- [21] Hashimoto, H. Transitivity of generalised fuzzy matrices. *Fuzzy Sets and Systems*. **17**(1985) 83-90.
- [22] Jacas, J. On the generators of T-indistinguishability operators. *Schocastica* **12** (1988) 49-63.
- [23] Jacas, J., Similarity relations. The calculation of minimal generating families. *Fuzzy Sets and Systems* **35** (1990) 151-162.
- [24] Jacas, J., Recasens, J., Fuzzy T-transitive relations: eigenvectors and generators, *Fuzzy Sets and Systems* **72** (1995) 147–154.
- [25] Jacas, J., Recasens, J., Decomposable indistinguishability operators, Proceedings of the Sixth IFSA Congress, Sao Paulo. 1995.
- [26] Kandel, L. Yelowitz, Fuzzy chains, *IEEE Trans. Systems Man Cybernet.* **4** (1974) 472–475.
- [27] Kruskal J. B., On the shortest spanning subtree of a graph and the travelling salesman problem. *Proceedings of the American Mathematical Society* **7** (1956) 48-50.
- [28] Kundu, S. An optimal $O(N^2)$ algorithm for computing the min-transitive closure of a weighted graph. *Inform. Proc. Lett.* **74** (2000) 215-220.
- [29] Larsen, H. K.; R. Yager, A fast maxmin similarity algorithm, in: J.C. Verdegay, M. Delgado (Eds.), The Interface Between AI and OR in a Fuzzy Environment, *IS 95 Verlag TUV Rheinland, KO oln, Germany*, (1989) 47– 55.
- [30] Larsen H., R. Yager, Efficient computation of transitive closures, *Fuzzy Sets and Systems*, **38** (1990) 81–90.
- [31] Lee, H.-S. An optimal algorithm for computing the max–min transitive closure of a fuzzy similarity matrix. *Fuzzy Sets and Systems* **123** (2001) 129–136.
- [32] Li, S.-Y., The simplest method of ascending value to find fuzzy transitive closure. *Fuzzy Sets and Systems* **38** (1990) 91-96.
- [33] Naessens, H., De Meyer, H., De Baets, B., Algorithms for the Computation of T-Transitive Closures. *IEEE Trans Fuzzy Systems* **10** (2002) 541-551.

- [34] Ovchinnikov, S. Representations of Transitive Fuzzy Relations, in Aspects of Vagueness, H. J. Skala, S. Termini y E. Trillas (Eds.), Reidel Pubs. 105-118. 1984.
- [35] Pfaltz, J.L., Jamilson R.E., Closure systems and their structure. *Information Sciences* **139** (2001) 275-286.
- [36] Prim, R. C. Shortest connection networks and some generalizations. *Bell System Technical Journal*, **36** (1957) 1389-1401.
- [37] Rodriguez, R. O., Esteva, F, Garcia, P., Godo, L. On implicative closure operators in approximate reasoning. *International Journal of Approximate Reasoning* **33** (2003) 159-184.
- [38] Seidel, R. On the all-pairs-shortest-path problem in unweighted undirected graphs. *J. Computer and Systems Sciences* **51**(1995) 400-403.
- [39] Schweizer, B., Sklar A. Probabilistic Metric Spaces, North-Holland, New York. 1984.
- [40] Tarjan R. E. Data Structures and Network Algorithms. Society for Industrial and Applied Mathematics. 1983.
- [41] Trillas, E., Valverde, L., An inquiry into indistinguishability operators, in Aspects of Vagueness, H. J. Skala, S. Termini y E. Trillas (Eds.), Reidel Pubs. 231-256. 1984.
- [42] Valverde, L., On the structure of F-indistinguishability operators. *Fuzzy Sets and Systems* **17** (1985) 313–328.
- [43] Wagenknecht, M., On transitive solutions of fuzzy equations, inequalities and lower approximation of fuzzy relations. *Fuzzy Sets and Systems* **75** (1995) 229-240.
- [44] Xian Xiao, An algorithm for calculating fuzzy transitive closure, *Fuzzy Math.* **5** (1985) 71–73.
- [45] Zadeh, L. A. *Fuzzy sets, Inform. and Control* **8** (1965) 338–353.
- [49] Zadeh, L. A., Similarity relations and fuzzy orderings, *Inform. Sci.* **3** (1971) 177–200.
- [47] Zhen Zhao, An improved algorithm for fuzzy classification analysis, *J. Lanzhou Univ.* **19** (1983) 160–163.