Fuzzy Rough Sets: from Theory into Practice

Chris Cornelis¹, Martine De Cock¹, Anna Maria Radzikowska²

¹ Dept. of Applied Mathematics and Computer Science
Ghent University, Krijgslaan 281 (S9), 9000 Gent, Belgium
{Chris.Cornelis, Martine.DeCock}@UGent.be

² Faculty of Mathematics and Information Science
Warsaw University of Technology,
Plac Politechniki 1, 00-661 Warsaw, Poland
annrad@mini.pw.edu.pl

Abstract

Fuzzy sets and rough sets address two important, and mutually orthogonal, characteristics of imperfect data and knowledge: while the former allow that objects belong to a set or relation to a given degree, the latter provide approximations of concepts in the presence of incomplete information. In this chapter, we explain how these notions can be combined into a hybrid theory that is able to capture the best of different worlds. At the heart of this synergy lie well thought out definitions of lower and upper approximations of fuzzy sets under fuzzy relations. To make the various concepts easier to grasp, we provide an illustrative application of fuzzy rough sets in information retrieval.

1 Introduction

Fuzzy sets (Zadeh [31], 1965), as well as the slightly younger rough sets (Pawlak [19], 1982), have left an important mark on the way we represent and compute with imperfect information nowadays. Each of them has fostered a broad research community, and their impact has also been clearly felt at the application level. Although it was recognized early on that the associated theories are complementary rather than competitive, perceived similarities between both concepts and efforts to prove that one of them subsumes the other, have somewhat stalled progress towards shaping a hybrid theory that combines their mutual strengths.

Still, seminal research on fuzzy rough set theory flourished during the 1990’s and early 2000’s (e.g. [8, 11, 12, 14, 16, 17, 23, 26, 30]). Recently, cross-disciplinary research has also profited from the popularization and widespread adoption of two important computing paradigms: granular computing—with its focus on clustering information entities
into granules in terms of similarity, indistinguishability, etc.—has helped the theoretical
underpinnings of the hybrid theory to come of age, while soft computing—a collection of
techniques that are tolerant of typical characteristics of imperfect data and knowledge, and
hence adhere closer to the human mind than conventional hard computing techniques—has
stressed the role of fuzzy sets and rough sets as partners, rather than adversaries, within
a panoply of practical applications.

Within the hybrid theory, Pawlak’s rough set framework for the construction of lower
and upper approximations of a concept \( C \) given incomplete information (a subset \( A \)
of a given universe \( X \), containing examples of \( C \)), and an equivalence relation \( R \) in \( X \) that
models “indiscernibility” or “indistinguishability”, has been extended in two ways:

1. The set \( A \) may be generalized to a fuzzy set in \( X \), allowing that objects can belong
to a concept (i.e., meet its characteristics) to varying degrees.

2. Rather than modeling elements’ indistinguishability, we may assess their similarity
(objects are similar to a certain degree), represented by a fuzzy relation \( R \). As a
result, objects are categorized into classes, or granules, with “soft” boundaries based
on their similarity to one another.

In this chapter, we consider the general problem of defining lower and upper approxi-
mations of a fuzzy set \( A \) by means of a fuzzy relation \( R \). A key ingredient to our exposition
is the fact that elements of \( X \) can belong, to varying degrees, to several “soft granules”
simultaneously. Not only does this property lie right at the heart of fuzzy set theory, a
similar phenomenon can already be observed in crisp, or traditional, rough set theory as
soon as the assumption that \( R \) is an equivalence relation (and hence induces a partition of
\( X \)) is abandoned. Within fuzzy rough set theory, the impact of this property—which plays
a crucial role towards defining the approximations—is felt still more strongly, since even fuzzy \( T \)-equivalence relations, the natural candidates for generalizing equivalence relations,
are subject to it.

The chapter is structured as follows. In Section 2, we first highlight those concepts
from rough set theory and fuzzy set theory that are crucial for the hybridization process.
Section 3 reviews various proposals for the definition of lower and upper approximation
in fuzzy rough set theory and examines their respective properties. To make the various
concepts easier to grasp, we end the chapter with an illustrative application of fuzzy rough
sets for query refinement in information retrieval. In this application, a thesaurus, or term-
term relation, defines an approximation space in which the query, which is defined as a set
of terms, can be approximated from the upper and the lower side. These approximations
correspond to modified queries that can be used subsequently to enhance the search process.
The use of fuzzy sets allows both the query and the thesaurus to be weighted, indicating
respectively the importance of certain query terms over others, and the strength of the
connections between specific terms.
2 Preliminaries

2.1 Rough Sets

Rough set analysis makes statements about the membership of some element \( y \) of \( X \) to the concept of which \( A \) is a set of examples, based on the indistinguishability between \( y \) and the elements of \( A \). Usually, indistinguishability is described by means of an equivalence relation \( R \) in \( X \); in this case, \((X, R)\) is called a standard, or Pawlak, approximation space. More generally, it is possible to replace \( R \) by any binary relation in \( X \), not necessarily an equivalence relation; we then call \((X, R)\) a generalized approximation space. In particular, the case of a reflexive \( R \), and of a tolerance, i.e. reflexive and symmetric, relation \( R \) have received ample attention in the literature. In all cases, \( A \) is approximated in two ways, resulting in the lower and upper approximation of the concept\(^1\).

2.1.1 Rough Sets in Pawlak Approximation Spaces

In a Pawlak approximation space \((X, R)\), an element \( y \) of \( X \) belongs to the lower approximation \( R \downarrow A \) of \( A \) if the equivalence class to which \( y \) belongs is included in \( A \). On the other hand \( y \) belongs to the upper approximation \( R \uparrow A \) of \( A \) if its equivalence class has a non-empty intersection with \( A \). Formally, the sets \( R \downarrow A \) and \( R \uparrow A \) are defined by, for \( y \) in \( X \),

\[
y \in R \downarrow A \quad \text{iff} \quad [y]_R \subseteq A \tag{1}
\]
\[
y \in R \uparrow A \quad \text{iff} \quad [y]_R \cap A \neq \emptyset \tag{2}
\]

In other words

\[
y \in R \downarrow A \quad \text{iff} \quad (\forall x \in X)((x, y) \in R \Rightarrow x \in A) \tag{3}
\]
\[
y \in R \uparrow A \quad \text{iff} \quad (\exists x \in X)((x, y) \in R \land x \in A) \tag{4}
\]

The underlying meaning is that \( R \downarrow A \) is the set of elements necessarily satisfying the concept (strong membership), while \( R \uparrow A \) is the set of elements possibly belonging to the concept (weak membership).

Some basic and easily verified properties of lower and upper approximation are summarized in Table 1. From 2., it holds that \( R \downarrow A \subseteq R \uparrow A \). If \( y \) belongs to the boundary region \( R \uparrow A \setminus R \downarrow A \), then there is some doubt, because in this case \( y \) is at the same time indistinguishable from at least one element of \( A \) and at least one element of \( X \) that is not in \( A \). Following [23], we call \((A_1, A_2)\) a rough set (in \((X, R)\)) as soon as there is a set \( A \) in \( X \) such that \( R \downarrow A = A_1 \) and \( R \uparrow A = A_2 \).

\(^1\)For completeness we mention that another stream concerning rough sets in the literature was initiated by Iwinski [10] who did not use an equivalence relation or tolerance relation as an initial building block to define the rough set concept. Although this formulation provides an elegant mathematical model, the absence of the equivalence relation makes his model hard to interpret and to use in practical applications. We therefore do not deal with it in this chapter; a more detailed comparison of the different views on rough set theory can be found in e.g. [29].
Table 1: Properties of lower and upper approximation in a Pawlak approximation space 
\((X, R)\); \(A\) and \(B\) are subsets of \(X\), and \(\text{co}\) denotes set-theoretic complement.

1. \(R^\uparrow A = \text{co}(R \downarrow (\text{co}A))\)
   \(R \downarrow A = \text{co}(R^\uparrow (\text{co}A))\)
2. \(R \downarrow A \subseteq A \subseteq R^\uparrow A\)
3. \(A \subseteq B \Rightarrow \begin{cases} R \downarrow A \subseteq R \downarrow B \\ R^\uparrow A \subseteq R^\uparrow B \end{cases}\)
4. \(R \downarrow (A \cap B) = R \downarrow A \cap R \downarrow B\)
   \(R^\uparrow (A \cap B) \subseteq R^\uparrow A \cap R^\uparrow B\)
5. \(R \downarrow (A \cup B) \supseteq R \downarrow A \cup R \downarrow B\)
   \(R^\uparrow (A \cup B) = R^\uparrow A \cup R^\uparrow B\)
6. \(R \downarrow (R \downarrow A) = R \downarrow A\)
   \(R^\uparrow (R^\uparrow A) = R^\uparrow A\)

2.1.2 Rough Sets in Generalized Approximation Spaces

For an arbitrary binary relation \(R\) in \(X\), the role of equivalence classes in Pawlak approximation spaces (cfr. formulas (1) and (2)) can be subsumed by the more general concept of \(R\)-foreset. For \(y\) in \(X\), the \(R\)-foreset \(Ry\) is defined by

\[Ry = \{x \mid x \in X \land (x, y) \in R\}\]

It is well known that an equivalence relation \(R\) induces a partition of \(X\), so if we consider two equivalence classes then they either coincide or are disjoint. It is therefore not possible for \(y\) to belong to two different equivalence classes at the same time. If \(R\) is a non-equivalence relation in \(X\), however, then it is quite normal that different foresets may partially overlap, as the following example illustrates.

**Example 1** Consider \(X = \{x_1, x_2, x_3, x_4\}\) and the relation \(R\) in \(X\) defined by

\[
\begin{array}{c|cccc}
R & x_1 & x_2 & x_3 & x_4 \\
\hline
x_1 & 1 & 0 & 1 & 0 \\
x_2 & 1 & 1 & 0 & 1 \\
x_3 & 0 & 1 & 1 & 0 \\
x_4 & 1 & 1 & 0 & 1 \\
\end{array}
\]

\(R\) is reflexive but neither symmetric nor transitive. The \(R\)-foreset of the elements of \(X\) are:

- \(Rx_1 = \{x_1, x_2, x_4\}\)
- \(Rx_2 = \{x_2, x_3, x_4\}\)
- \(Rx_3 = \{x_1, x_3\}\)
- \(Rx_4 = \{x_2, x_4\}\)
By the definition used in Pawlak approximation spaces, \( y \) belongs to the lower approximation of \( A \) if \( R_y \) is included in \( A \). In view of the discussion above however, in generalized approximation spaces it makes sense to consider also other \( R \)-foresets that contain \( y \), and to assess their inclusion into \( A \) as well for the lower approximation, and their overlap with \( A \) for the upper approximation. This idea, first explored by Pomykal [21], results in the following (inexhaustive!) list of candidate definitions for the lower and the upper approximation of \( A \):

1. \( y \) belongs to the lower approximation of \( A \) iff 
   - (a) all \( R \)-foresets containing \( y \) are included in \( A \)
   - (b) at least one \( R \)-foreset containing \( y \) is included in \( A \)
   - (c) \( R_y \) is included in \( A \)

2. \( y \) belongs to the upper approximation of \( A \) iff 
   - (a) all \( R \)-foresets containing \( y \) have a non-empty intersection with \( A \)
   - (b) at least one \( R \)-foreset containing \( y \) has a non-empty intersection with \( A \)
   - (c) \( R_y \) has a non-empty intersection with \( A \)

Paraphrasing these expressions, we obtain the following definitions:

1. The tight, loose and (usual) lower approximation of \( A \) are defined as 
   - (a) \( y \in R \downarrow \downarrow A \) iff \( (\forall z \in X)(y \in Rz \Rightarrow Rz \subseteq A) \)
   - (b) \( y \in R \downarrow \uparrow A \) iff \( (\exists z \in X)(y \in Rz \land Rz \subseteq A) \)
   - (c) \( y \in R \downarrow A \) iff \( R_y \subseteq A \) 
   for all \( y \) in \( X \).

2. The tight, loose and (usual) upper approximation of \( A \) are defined as 
   - (a) \( y \in R \uparrow \downarrow A \) iff \( (\forall z \in X)(y \in Rz \Rightarrow Rz \cap A \neq \emptyset) \)
   - (b) \( y \in R \uparrow \uparrow A \) iff \( (\exists z \in X)(y \in Rz \land Rz \cap A \neq \emptyset) \)
   - (c) \( y \in R \uparrow A \) iff \( R_y \cap A \neq \emptyset \) 
   for all \( y \) in \( X \).

The terminology “tight” refers to the fact that we take all \( R \)-foresets into account, giving rise to a strict or tight requirement. For the “loose” approximations, we only look at “the best one” which is clearly a more flexible demand. For an equivalence relation \( R \), all of the above definitions coincide, but in general they can be different as the following example shows.
Example 2 Consider $X$ and $R$ as defined in Example 1, and the set $A = \{x_1, x_3\}$. The various lower and upper approximations of $A$ in $(X, R)$ are:

\[
R \downarrow A = \{x_3\} \quad R \uparrow A = \{x_1, x_2, x_3\} \\
R \downarrow \downarrow A = \emptyset \quad R \uparrow \uparrow A = \{x_1, x_3\} \\
R \downarrow \uparrow A = \{x_1, x_3\} \quad R \uparrow \downarrow A = \{x_1, x_3\}
\]

In the remainder of this section, we assume that $R$ is a tolerance relation, which is a basic requirement if $R$ is supposed to model indistinguishability. We refer to the resulting generalized approximation space $(X, R)$ as a reflexive and symmetric approximation space. In this case too it is not uncommon at all for the various lower and upper approximations to be different. This will be noticeable e.g. in the elaborate example in Section 4.

The symmetry of $R$ allows to verify the following important relationships between the approximations:

\[
R \downarrow \downarrow A = R \downarrow (R \downarrow A) \quad (6) \\
R \downarrow \uparrow A = R \uparrow (R \downarrow A) \quad (7) \\
R \downarrow \uparrow A = R \downarrow (R \uparrow A) \quad (8) \\
R \downarrow \downarrow A = R \uparrow (R \uparrow A) \quad (9)
\]

Table 2 lists the properties of the different approximations. Interesting observations to make from this table include:

1. By 1., there are three pairs of dual approximation operators w.r.t. set complement.

2. Property 2. shows the relationship between the approximations in terms of inclusion, and how $A$ itself fits into this picture. Note how these relationships nicely justify the terminology.

3. Loose lower, resp. tight upper, approximation satisfies only a weak interaction property w.r.t. set intersection, resp. union (Property 4. and 5.).

4. By Property 6. of Table 1, when $R$ is an equivalence relation, lower and upper approximation are idempotent. This means that in Pawlak approximation spaces, maximal reduction and expansion are achieved within one approximation step. By Property 6. of Table 2, the same holds true for loose lower and tight upper approximation in a reflexive and symmetric approximation space, but not for the other operators; for these, a gradual reduction/expansion process is obtained by successively taking approximations. We discuss this further in Section 3.

2.2 Fuzzy Sets

In the context of fuzzy rough set theory, $A$ is a fuzzy set in $X$, i.e. an $X \rightarrow [0, 1]$ mapping, while $R$ is a fuzzy relation in $X$, i.e. a fuzzy set in $X \times X$. For fuzzy sets $A$ and $B$ in $X$,
Table 2: Properties of lower and upper approximation in a reflexive and symmetric approximation space \((X, R)\); \(A\) and \(B\) are subsets of \(X\), and \(\text{co}\) denotes set-theoretic complement.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>(R\uparrow A = \text{co}(R\downarrow (\text{co}A)))</td>
</tr>
<tr>
<td></td>
<td>(R\downarrow A = \text{co}(R\uparrow (\text{co}A)))</td>
</tr>
<tr>
<td></td>
<td>(R\uparrow\downarrow A = \text{co}(R\downarrow\uparrow (\text{co}A)))</td>
</tr>
<tr>
<td></td>
<td>(R\downarrow\uparrow A = \text{co}(R\uparrow\downarrow (\text{co}A)))</td>
</tr>
<tr>
<td>4.</td>
<td>(R\downarrow (A \cap B) = R\downarrow A \cap R\downarrow B)</td>
</tr>
<tr>
<td></td>
<td>(R\uparrow (A \cap B) \subseteq R\uparrow A \cap R\uparrow B)</td>
</tr>
<tr>
<td></td>
<td>(R\downarrow (A \cap B) \subseteq R\downarrow A \cap R\downarrow B)</td>
</tr>
<tr>
<td></td>
<td>(R\uparrow (A \cap B) \subseteq R\uparrow A \cap R\uparrow B)</td>
</tr>
<tr>
<td>2.</td>
<td>(R\downarrow A \subseteq R\uparrow A \subseteq R\uparrow\downarrow A \subseteq A)</td>
</tr>
<tr>
<td></td>
<td>(A \subseteq R\downarrow\uparrow A \subseteq R\uparrow A \subseteq R\uparrow\downarrow A)</td>
</tr>
</tbody>
</table>
| 3. | \(A \subseteq B \Rightarrow \begin{cases} 
R\downarrow A \subseteq R\downarrow B \\
R\uparrow A \subseteq R\uparrow B \\
R\downarrow\uparrow A \subseteq R\downarrow\uparrow B \\
R\downarrow\downarrow A \subseteq R\downarrow\downarrow B \\
\end{cases}\) |
| 5. | \(R\downarrow (A \cup B) \supseteq R\downarrow A \cup R\downarrow B\) |
|   | \(R\uparrow (A \cup B) = R\uparrow A \cup R\uparrow B\) |
|   | \(R\downarrow (A \cup B) \supseteq R\downarrow A \cup R\downarrow B\) |
|   | \(R\uparrow (A \cup B) = R\uparrow A \cup R\uparrow B\) |
| 6. | \(R\downarrow (R\downarrow\uparrow A) = R\downarrow\uparrow A\) |
|   | \(R\uparrow (R\downarrow\uparrow A) = R\downarrow\uparrow A\) |

\(A \subseteq B\) if \(A(x) \leq B(x)\) for all \(x\) in \(X\). For all \(y\) in \(X\), the \(R\)-foreset of \(y\) is the fuzzy set \(R_y\) defined by

\[
R_y(x) = R(x, y)
\]

for all \(x\) in \(X\). The fuzzy logical counterparts of the connectives in (3) and (4) play an important role in the generalization of lower and upper approximation; we therefore recall some important definitions.

First, a negator \(N\) is a decreasing \([0, 1] \rightarrow [0, 1]\) mapping satisfying \(N(0) = 1\) and \(N(1) = 0\). \(N\) is called involutive if \(N(N(x)) = x\) for all \(x\) in \([0, 1]\). The standard negator \(N_s\) is defined by

\[
N_s(x) = 1 - x
\]

A negator \(N\) induces a corresponding fuzzy set complement \(\text{co}_N\): for any fuzzy set \(A\) in \(X\) and every element \(x\) in \(X\),

\[
\text{co}_N(A) = N(A(x))
\]
defined by

\[(A \cap_T B)(x) = T(A(x), B(x))\] (13)

\[(A \cap_S B)(x) = S(A(x), B(x))\] (14)

for all \(x\) in \(X\). Throughout this chapter, \(A \cap_T B\) and \(A \cup_S B\) are abbreviated to \(A \cap B\) and \(A \cup B\) and called standard intersection and union, respectively.

Finally, an implicator \(I\) is any \([0, 1] \rightarrow [0, 1]\)-mapping satisfying \(I(0, 0) = 1\) and \(I(1, x) = x\), for all \(x\) in \([0, 1]\). Moreover we require \(I\) to be decreasing in its first, and increasing in its second component. If \(T\) is a t-norm, the mapping \(I_T\) defined by, for all \(x\) and \(y\) in \([0, 1]\),

\[I_T(x, y) = \sup\{\lambda | \lambda \in [0, 1] \text{ and } T(x, \lambda) \leq y\}\] (15)

is an implicator, usually called the residual implicator of \(T\). If \(T\) is a t-norm and \(N\) is an involutive negator, then the mapping \(I_{T,N}\) defined by, for all \(x\) and \(y\) in \([0, 1]\),

\[I_{T,N}(x, y) = N(T(x, N(y)))\] (16)

is an implicator, usually called the S-implicator induced by \(T\) and \(N\).

An implicator \(I\) induces a corresponding negator \(N_I\), defined by \(N_I(x) = I(x, 0)\), for all \(x\) in \(X\). If \(I = I_{T,N}\), then \(N_I = N\). In Table 4, we mention some important S- and residual implicators; the S-implicators are induced by the standard negator \(N_s\).

**Table 3: Well-known t-norms and t-conorms; \(x\) and \(y\) in \([0, 1]\).**

<table>
<thead>
<tr>
<th>t-norm (T)</th>
<th>t-conorm (S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(T_M(x, y) = \min(x, y))</td>
<td>(S_M(x, y) = \max(x, y))</td>
</tr>
<tr>
<td>(T_P(x, y) = xy)</td>
<td>(S_P(x, y) = x + y - xy)</td>
</tr>
<tr>
<td>(T_W(x, y) = \max(x + y - 1, 0))</td>
<td>(S_W(x, y) = \min(x + y, 1))</td>
</tr>
</tbody>
</table>

**Table 4: Well-known implicators; \(x\) and \(y\) in \([0, 1]\).**

<table>
<thead>
<tr>
<th>S-implicator (I)</th>
<th>residual implicator (I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I_{T_M,N_s}(x, y) = \max(1 - x, y))</td>
<td>(I_{T_M}(x, y) = \begin{cases} 1, &amp; \text{if } x \leq y \ y, &amp; \text{otherwise} \end{cases})</td>
</tr>
<tr>
<td>(I_{T_P,N_s}(x, y) = 1 - x + xy)</td>
<td>(I_{T_P}(x, y) = \begin{cases} 1, &amp; \text{if } x \leq y \ \frac{y}{x} &amp; \text{otherwise} \end{cases})</td>
</tr>
<tr>
<td>(I_{T_W,N_s}(x, y) = \min(1 - x + y, 1))</td>
<td>(I_{T_W}(x, y) = \min(1 - x + y, 1))</td>
</tr>
</tbody>
</table>

In fuzzy rough set theory, we require a way to express that objects are similar to each other to some extent. In the remainder of this chapter, similarity is modelled by a fuzzy tolerance relation \(R\), that is
hold for all \( x \) and \( y \) in \( X \). Additionally, \( T \)-transitivity (for a particular t-norm \( T \)) is sometimes imposed: for all \( x, y \) and \( z \) in \( X \),

\[
T(R(x, y), R(y, z)) \leq R(x, z) \quad (T\text{-transitivity})
\]

\( R \) is then called a fuzzy \( T \)-equivalence relation; because equivalence relations are used to model equality, fuzzy \( T \)-equivalence relations are commonly considered to represent approximate equality. In general, for a fuzzy tolerance relation \( R \), we call \( R_y \) the “fuzzy similarity class” of \( y \).

3 Fuzzy Rough Sets

3.1 Definitions

Research on fuzzifying lower and upper approximation in the spirit of Pawlak emerged in the late 1980’s. Chronologically, the first proposals are due to Nakamura [16], and to Dubois and Prade [8] who drew inspiration from an earlier publication by Fariñas del Cerro and Prade [9].

In developing the generalizations, the central focus moved from elements’ indistinguishability (for instance, w.r.t. their attribute values in an information system) to their similarity: objects are categorized into classes with “soft” boundaries based on their similarity to one another. A concrete advantage of such a scheme is that abrupt transitions between classes are replaced by gradual ones, allowing that an element can belong (to varying degrees) to more than one class. An example at hand is an attribute “age” in an information table: in order to restrict the number of equivalence classes, classical rough set theory advises to discretize age values by a crisp partition of the universe, e.g. using intervals \([0, 10], [10, 20], \ldots \). This does not always reflect our intuition, however: by imposing such harsh boundaries, a person who has just turned eleven will not be taken into account in the \([0, 10]\) class, even when she is only at a minimal remove from full membership in that class.

Guided by that observation, many people have suggested alternatives for defining fuzzified approximation operators, e.g. using axiomatic approaches [14], based on Iwinski-type rough sets [17], in terms of \( \alpha \)-cuts [30], level fuzzy sets [13] or fuzzy inclusion measures [11], etc. Some authors (e.g. [26, 30]) explicitly distinguish between rough fuzzy sets (approximations of a fuzzy set in a crisp approximation space) and fuzzy rough sets (approximations of a crisp set in a fuzzy approximation space, i.e., defined by a fuzzy relation \( R \)).

A fairly general definition of a fuzzy rough set, absorbing earlier suggestions in the same direction, was given by Radzikowska and Kerre [23]. They paraphrased formulas (3) and (4), which hold in the crisp case, to define the lower and upper approximation of a fuzzy set \( A \) in \( X \) as the fuzzy sets \( R\downarrow A \) and \( R\uparrow A \) in \( X \), constructed by means of an implicator.
\( \mathcal{T} \), a t-norm \( \mathcal{T} \) and a fuzzy \( \mathcal{T} \)-equivalence relation \( R \) in \( X \),

\[
\begin{align*}
(R \downarrow A)(y) &= \inf_{x \in X} \mathcal{T}(R(x, y), A(x)) \\
(R \uparrow A)(y) &= \sup_{x \in X} \mathcal{T}(R(x, y), A(x))
\end{align*}
\]

for all \( y \) in \( X \). \((A_1, A_2)\) is called a fuzzy rough set (in \((X, R)\)) as soon as there is a fuzzy set \( A \) in \( X \) such that \( R \downarrow A = A_1 \) and \( R \uparrow A = A_2 \). Formulas (17) and (18) for \( R \downarrow A \) and \( R \uparrow A \) can also be interpreted as the degree of inclusion of \( Ry \) in \( A \) and the degree of overlap of \( Ry \) and \( A \) respectively, which indicates the semantical link with (1) and (2).

What this definition does not take into account, however, is the fact that if \( R \) is a fuzzy \( \mathcal{T} \)-equivalence relation then it is quite normal that, because of the intermediate degrees of membership, different foresets are not necessarily disjoint. The following example, taken from [7], illustrates this.

**Example 3** In applications \( \mathcal{T}_W \) is often used as a t-norm because the notion of fuzzy \( \mathcal{T}_W \)-equivalence relation is dual to that of a pseudo-metric [3]. Let the fuzzy \( \mathcal{T}_W \)-equivalence relation \( R \) in \( \mathbb{R} \) be defined by

\[
R(x, y) = \max(1 - |x - y|, 0)
\]

for all \( x \) and \( y \) in \( \mathbb{R} \). Figure ?? depicts the \( R \)-foresets of 1.3, 2.2, 3.1 and 4.0. They are clearly different. Now let us focus on the \( R \)-foresets of 3.1 and 4.0, i.e. on the fuzzy sets \( R(3.1) \) and \( R(4.0) \). From

\[
R(3.1, 3.5) = \max(1 - |3.1 - 3.5|, 0) = 0.6 \\
R(4.0, 3.5) = \max(1 - |4.0 - 3.5|, 0) = 0.5
\]

we obtain that 3.5 belongs to degree 0.6 to \( R(3.1) \), and to degree 0.5 to \( R(4.0) \). Hence

\[
(R(3.1) \cap_{\mathcal{T}_W} R(4.0))(3.1) = \mathcal{T}_W(0.6, 0.5) = \max(0.6 + 0.5 - 1, 0) = 0.1
\]

In other words 3.5 belongs to degree 0.1 to the \( \mathcal{T}_W \)-intersection of the \( R \)-foresets of 3.1 and 4.0, i.e., these \( R \)-foresets are not disjoint even though they are different.

In other words, the traditional distinction between equivalence and non-equivalence relations is lost when moving on to a fuzzy \( \mathcal{T} \)-equivalence relation, so it makes sense to exploit the fact that an element can belong to some degree to several \( R \)-foresets of *any* fuzzy relation \( R \) at the same time. Natural generalizations to the definitions from Section 2.1.2 were therefore proposed in [5, 7].

**Definition 1** Let \( R \) be a fuzzy relation in \( X \) and \( A \) a fuzzy set in \( X \).

1. The tight, loose and (usual) lower approximation of \( A \) are defined as
\[(a) \quad (R_{\downarrow\downarrow} A)(y) = \inf_{z \in X} I(Rz(y), \inf_{x \in X} I(Rz(x), A(x)))\]

\[(b) \quad (R_{\uparrow\downarrow} A)(y) = \sup_{z \in X} T(Rz(y), \inf_{x \in X} I(Rz(x), A(x)))\]

\[(c) \quad (R_{\downarrow} A)(y) = \inf_{x \in X} I(Ry(x), A(x))\]

for all \(y \in X\).  

2. The tight, loose and (usual) upper approximation of \(A\) are defined as

\[(a) \quad (R_{\downarrow\uparrow} A)(y) = \inf_{z \in X} I(Rz(y), \sup_{x \in X} T(Rz(x), A(x)))\]

\[(b) \quad (R_{\uparrow\uparrow} A)(y) = \sup_{z \in X} T(Rz(y), \sup_{x \in X} T(Rz(x), A(x)))\]

\[(c) \quad (R_{\uparrow} A)(y) = \sup_{x \in X} T(Ry(x), A(x))\]

for all \(y \in X\).

\[\text{Example 4} \quad \text{Let } X = [0,1] \text{ and } A \text{ be the fuzzy set in } X \text{ defined as } A(x) = x, \text{ for all } x \text{ in } X. \text{ Let the reflexive and symmetric fuzzy relation } R \text{ in } X \text{ be defined as}\]

\[
R(x, y) = \begin{cases} 
1 & \text{if } |x - y| < 0.1 \\
0 & \text{otherwise}
\end{cases}
\]

(19)

for all \(x\) and \(y\) in \(X\). One can verify that for \(y\) in \(X\), and for any implicator \(I\):

\[
R_{\downarrow} A(y) = \inf_{z \in X} I(Rz(y), A(z)) \\
= \inf\{z \mid z \in X \land z \in [y - 0.1, y + 0.1]\} \\
= \max(0, y - 0.1)
\]

(20)

Hence \(R_{\downarrow} A(0.95) = 0.85\). Furthermore, for any t-norm \(T\),

\[
R_{\uparrow} A(y) = \sup_{z \in X} T(A(z), R(z, y)) \\
= \sup\{z \mid z \in X \land z \in [y - 0.1, y + 0.1]\} \\
= \min(1, y + 0.1)
\]

(21)

hence \(R_{\uparrow} A(0.05) = 0.15\). We verify

\[
(R_{\downarrow\downarrow} A)(0.95) = \inf_{z \in X} I(Rz(0.95), \max(0, z - 0.1)) \\
= \inf\{\max(0, z - 0.1) \mid z \in [0.85, 1]\} = 0.75
\]

(22)

and
\[(R\uparrow\downarrow A)(0.95) = \sup \{ R(z, 0.95), \max(0, z - 0.1) \} \]
\[= \sup \{ \max(0, z - 0.1) | z \in [0.85, 1] \} = 0.9 \tag{23} \]

In the same way one can verify that \((R\uparrow\uparrow A)(0.05) = 0.25\) and \((R\downarrow\uparrow A)(0.05) = 0.1\). This illustrates that all the approximations from Definition 1 are different.

### 3.2 Properties of Fuzzy Rough Sets

In this section, we assume that \(R\) is a fuzzy tolerance relation, i.e., a reflexive and symmetric fuzzy relation, in \(X\). Some properties require additional \(T\)-transitivity of \(R\); whenever this is the case we mention it explicitly. An overview of the properties discussed in this section is given in Table 5.

#### 3.2.1 Links between the Approximations

Just like in the crisp case, tight and loose approximation operators can be expressed in terms of the usual ones, due to the symmetry of \(R\).

**Proposition 1** [7] For every fuzzy set \(A\) in \(X\)
\[
R\downarrow\downarrow A = R\downarrow (R\downarrow A) \tag{24}
\]
\[
R\uparrow\downarrow A = R\uparrow (R\downarrow A) \tag{25}
\]
\[
R\downarrow\uparrow A = R\downarrow (R\uparrow A) \tag{26}
\]
\[
R\uparrow\uparrow A = R\uparrow (R\uparrow A) \tag{27}
\]

The monotonicity of the approximations follows easily due to the monotonicity of the fuzzy logical operators involved. This is reflected in the next proposition.

**Proposition 2** For every fuzzy set \(A\) and \(B\) in \(X\)
\[
A \subseteq B \Rightarrow \left\{ \begin{array}{l}
R\downarrow A \subseteq R\downarrow B \\
R\uparrow A \subseteq R\uparrow B \\
R\downarrow\uparrow A \subseteq R\uparrow\downarrow B \\
R\uparrow\uparrow A \subseteq R\downarrow\uparrow B \\
R\downarrow\uparrow A \subseteq R\uparrow\uparrow B \\
R\uparrow\downarrow A \subseteq R\uparrow\downarrow B
\end{array} \right. \tag{28}
\]

The following proposition supports the idea of approximating a concept from the lower and the upper side.

**Proposition 3** [23] For every fuzzy set \(A\) in \(X\)
\[
R\downarrow A \subseteq A \subseteq R\uparrow A \tag{29}
\]
For the tight and loose approximations, due to Propositions 1, 2 and 3, we can make the following general observations.

**Proposition 4** For every fuzzy set \( A \) in \( X \)

\[
\begin{align*}
R \downarrow \downarrow A & \subseteq R \downarrow A \subseteq A \subseteq R \uparrow A \subseteq R \uparrow \uparrow A \\
R \downarrow A & \subseteq R \downarrow \downarrow A \subseteq R \uparrow A \\
R \downarrow A & \subseteq R \downarrow \uparrow A \subseteq R \uparrow A \\
\end{align*}
\]

However, the proposition does not give any immediate information about a direct relationship between the loose lower and the tight upper approximation in terms of inclusion, and about how \( A \) itself fits in this picture. The following proposition sheds some light on this matter.

**Proposition 5** [1] If \( T \) is a left-continuous t-norm and \( I \) is its residual implicator, then for every fuzzy set \( A \) in \( X \)

\[
R \uparrow \downarrow A \subseteq A \subseteq R \downarrow \uparrow A
\]

Proposition 5 does not hold in general for other choices of t-norms and implicators, in particular for S-implicators, as the next example illustrates.

**Example 5** Consider the fuzzy \( T \)-equivalence relation \( R \) in \( X = \{a, b\} \) given by

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>1.0</td>
<td>0.2</td>
</tr>
<tr>
<td>( b )</td>
<td>0.2</td>
<td>1.0</td>
</tr>
</tbody>
</table>

and the fuzzy set \( A \) in \( X \) defined by \( A(a) = 1.0 \) and \( A(b) = 0.8 \). Furthermore let \( T = T_M \) and \( I = I_{T_M,N_s} \). Then

\[
(R \uparrow A)(a) = \max(\min(R(a, a), R(a)), \min(R(b, a), R(b))) = \max(\min(1.0, 1.0), \min(0.2, 0.8)) = 1.0
\]

\[
(R \uparrow A)(b) = \max(\min(R(a, b), R(a)), \min(R(b, b), R(b))) = \max(\min(0.2, 1.0), \min(1, 0.8)) = 0.8
\]

hence

\[
(R \downarrow A)(a) = \min(\max(1 - R(a, a), (R \uparrow A)(a)), \max(1 - R(b, a), (R \uparrow A)(b))) = \min(\max(0.0, 1.0), \max(0.8, 0.8)) = 0.8
\]

which makes it clear that \( A \not\subseteq R \downarrow \uparrow A \). This example reveals that S-implicators do not lend themselves for working with loose lower and tight upper approximation.

From all of the above we obtain, for any fuzzy tolerance relation \( R \) in \( X \),

\[
R \downarrow \downarrow A \subseteq R \downarrow A \subseteq R \downarrow \uparrow A \subseteq A \subseteq R \uparrow \downarrow A \subseteq R \uparrow A \subseteq R \uparrow \uparrow A
\]

provided that \( T \) and \( I \) satisfy the conditions of Proposition 5.
3.2.2 Interaction with Set-Theoretic Operations

The following proposition shows that, given some elementary conditions on the involved connectives, the usual lower and upper approximation are dual w.r.t. fuzzy set complement.

**Proposition 6** [4] If $T$ is a t-norm, $N$ an involutive negator and $I$ the corresponding S-implicator; or, if $T$ is a left-continuous t-norm, $I$ its residual implicator and $N_T$ is an involutive negator, then

$$R \uparrow A = co_N(R \downarrow (co_N A)) \quad (35)$$
$$R \downarrow A = co_N(R \uparrow (co_N A)) \quad (36)$$

Combining this result with Proposition 1, it is easy to see that under the same conditions, tight upper and loose lower approximation are dual w.r.t. set complement, as are loose upper and tight lower approximation.

**Proposition 7** [23] For any fuzzy sets $A$ and $B$ in $X$

$$R \downarrow (A \cap B) = R \downarrow A \cap R \downarrow B \quad (37)$$
$$R \uparrow (A \cap B) \subseteq R \uparrow A \cap R \uparrow B \quad (38)$$
$$R \downarrow (A \cup B) \subseteq R \downarrow A \cup R \downarrow B \quad (39)$$
$$R \uparrow (A \cup B) = R \uparrow A \cup R \uparrow B \quad (40)$$

Again by Proposition 1, one can also verify the following equalities

$$R \downarrow \downarrow (A \cap B) = R \downarrow \downarrow A \cap R \downarrow \downarrow B \quad (41)$$
$$R \uparrow \uparrow (A \cup B) = R \uparrow \uparrow A \cup R \uparrow \uparrow B \quad (42)$$

whereas for the remaining interactions, the same inclusions hold as in the crisp case (see Table 2).

3.2.3 Maximal Expansion and Reduction

Taking an upper approximation of $A$ in practice corresponds to expanding $A$, while lower approximation is meant to reduce $A$. However this refining process does not go on forever. The following property says that with the loose lower and the tight upper approximation maximal reduction and expansion are achieved within one approximation.

**Proposition 8** [1] If $T$ is a left-continuous t-norm and $I$ its residual implicator then for every fuzzy set $A$ in $X$

$$R \uparrow \downarrow (R \downarrow \downarrow A) = R \uparrow \downarrow A \text{ and } R \downarrow \uparrow (R \uparrow \uparrow A) = R \downarrow \uparrow A \quad (43)$$
To investigate the behaviour of the loose upper and tight lower approximation w.r.t. expansion and reduction, we first establish links with the composition of $R$ with itself. Recall that the composition of fuzzy relations $R$ and $S$ in $X$ is the fuzzy relation $R \circ S$ in $X$ defined by

\[(R \circ S)(x, z) = \sup_{y \in X} T(R(x, y), S(y, z))\] (44)

for all $x$ and $z$ in $X$.

**Proposition 9** [7] If $T$ is a left-continuous t-norm then for every fuzzy set $A$ in $X$

\[R^{\uparrow \uparrow} A = (R \circ R)^\uparrow A\] (45)

**Proposition 10** [7] If $T$ is a t-norm and $\mathcal{I}$ is an implicator that is left-continuous in its first component and right-continuous in its second component, and if $T$ and $\mathcal{I}$ satisfy the shunting principle\(^2\)

\[\mathcal{I}(T(x, y), z) = \mathcal{I}(x, \mathcal{I}(y, z))\] (46)

then for every fuzzy set $A$ in $X$

\[R^{\downarrow \downarrow} A = (R \circ R)^\downarrow A\] (47)

Let us use the following notation, for $n > 1$,

\[R^1 = R \text{ and } R^n = R \circ R^{n-1}\] (48)

From Proposition 9, it follows that taking $n$ times successively the upper approximation of a fuzzy set under $R$ corresponds to taking the upper approximation once under the composed fuzzy relation $R^n$. Proposition 10 states a similar result for the lower approximation. For the particular case of a fuzzy $T$-equivalence relation, we have the following important result.

**Proposition 11** [23] If $R$ is a fuzzy $T$-equivalence relation in $X$ then

\[R \circ R = R\] (49)

In other words, using a $T$-transitive fuzzy relation $R$, options (1a) and (1c) of Definition 1 coincide, as well as options (2b) and (2c). The following proposition states that under slightly stronger conditions, they also coincide with (1b), respectively (2a).

**Proposition 12** [1, 23] If $R$ is a fuzzy $T$-equivalence relation in $X$, $T$ is a left-continuous t-norm and $\mathcal{I}$ its residual implicator then for every fuzzy set $A$ in $X$

\[R^{\downarrow \uparrow} A = R \downarrow A \text{ and } R^{\uparrow \downarrow} A = R \uparrow A\] (50)

\(^2\)The conditions of this proposition are satisfied both by a left continuous t-norm and its residual implicator [18] as well as by a continuous t-norm and an S-implicator induced by it [23].
Table 5: Properties of lower and upper approximation in a reflexive and symmetric fuzzy approximation space $(X, R)$; $A$ and $B$ are fuzzy sets in $X$.

<table>
<thead>
<tr>
<th>Property</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $R \uparrow A = co_N(R \downarrow (co_N A))$</td>
<td>$N$ involutive, $\mathcal{I} = \mathcal{I}_{T,N}$ (S-implicator);</td>
</tr>
<tr>
<td>$R \downarrow A = co_N(R \uparrow (co_N A))$</td>
<td>or</td>
</tr>
<tr>
<td>$R \uparrow A = co_N(R \downarrow (co_N A))$</td>
<td>$\mathcal{T}$ left-continuous, $\mathcal{I} = \mathcal{I}_T$ (residual implicator)</td>
</tr>
<tr>
<td>$R \downarrow A = co_N(R \uparrow (co_N A))$</td>
<td>and $N_T$ involutive</td>
</tr>
<tr>
<td>$R \downarrow A = co_N(R \uparrow (co_N A))$</td>
<td>(Proposition 6)</td>
</tr>
<tr>
<td>$R \uparrow A = co_N(R \downarrow (co_N A))$</td>
<td>(Proposition 4 and 5)</td>
</tr>
<tr>
<td>2. $R \downarrow A \subseteq R \uparrow A \subseteq R \uparrow \downarrow A \subseteq A \subseteq R \uparrow \uparrow A$</td>
<td>(Proposition 2)</td>
</tr>
<tr>
<td>$R \downarrow A \subseteq R \uparrow A \subseteq R \uparrow \downarrow A \subseteq A \subseteq R \uparrow \uparrow A$</td>
<td>\begin{cases} R \downarrow A \subseteq R \uparrow B \ R \uparrow A \subseteq R \uparrow B \ R \downarrow A \subseteq R \downarrow B \ R \uparrow A \subseteq R \downarrow B \ R \downarrow A \subseteq R \downarrow B \end{cases}</td>
</tr>
<tr>
<td>3. $A \subseteq B$</td>
<td>Always (Proposition 2)</td>
</tr>
<tr>
<td>4. $R \downarrow (A \cap B) = R \downarrow A \cap R \downarrow B$</td>
<td>Always (Proposition 7)</td>
</tr>
<tr>
<td>$R \uparrow (A \cap B) \subseteq R \uparrow A \cap R \uparrow B$</td>
<td>\begin{cases} R \downarrow A \subseteq R \downarrow B \ R \uparrow A \subseteq R \downarrow B \ R \downarrow A \subseteq R \downarrow B \ R \uparrow A \subseteq R \downarrow B \ R \downarrow A \subseteq R \downarrow B \end{cases}</td>
</tr>
<tr>
<td>5. $R \downarrow (A \cup B) \supseteq R \downarrow A \cup R \downarrow B$</td>
<td>Always (Proposition 7)</td>
</tr>
<tr>
<td>$R \uparrow (A \cup B) = R \uparrow A \cup R \uparrow B$</td>
<td>\begin{cases} R \downarrow A \subseteq R \downarrow B \ R \uparrow A \subseteq R \downarrow B \ R \downarrow A \subseteq R \downarrow B \ R \uparrow A \subseteq R \downarrow B \ R \downarrow A \subseteq R \downarrow B \end{cases}</td>
</tr>
<tr>
<td>6. $R \uparrow (R \downarrow A) = R \uparrow A$</td>
<td>$\mathcal{T}$ left-continuous, $\mathcal{I} = \mathcal{I}_T$ (Proposition 8)</td>
</tr>
<tr>
<td>$R \uparrow (R \downarrow A) = R \downarrow A$</td>
<td>(Proposition 8)</td>
</tr>
<tr>
<td>$R \downarrow A = R \downarrow A = R \downarrow A$</td>
<td>$R$ a fuzzy $\mathcal{T}$-equivalence relation in $X$,</td>
</tr>
<tr>
<td>$R \downarrow A = R \uparrow A = R \downarrow A$</td>
<td>$\mathcal{T}$ left-continuous, $\mathcal{I} = \mathcal{I}_T$ (Proposition 11 and 12)</td>
</tr>
</tbody>
</table>
This means that, using a fuzzy $T$-equivalence relation to model similarity, we obtain maximal reduction or expansion in one phase, regardless of which of the approximations from Definition 1 is used. As Example 2 already illustrated for the crisp case, when we abandon $(T \cdot)$transitivity, this behaviour is not always exhibited. In general, when $R$ is not $T$-transitive and the universe $X$ is finite, it is known that the $T$-transitive closure of $R$ is given by $R^{T \cdot (X-1)}$ (assuming $|X| \geq 2$) [15], hence

$$R \circ R^{T \cdot (X-1)} = R^{T \cdot (X-1)}$$ (51)

In other words with the lower and upper approximation, maximal reduction and expansion will be reached in at most $|X - 1|$ steps, while with the tight lower and the loose upper approximation it can take at most $\lceil |X - 1|/2 \rceil$ steps.

4 Application to Information Retrieval

4.1 Query refinement

One of the most common ways to retrieve information from the WWW is keyword based search: the user inputs a query consisting of one or more keywords and the search system returns a list of web documents, ranked according to their relevance to the query. In the basic approach, documents are not returned as search results if they do not contain (one of) the exact keywords of the query. There are various reasons why such an approach might fall short. On one hand there are word mismatch problems: the user knows what he is looking for and he is able to describe it, but the query terms he uses do not exactly correspond to those in the document containing the desired information because of differences in terminology. This problem is even more significant in the context of the WWW than in other, more focussed information retrieval applications, because of the very heterogeneous sources of information expressed in different jargon or even in different natural languages.

Besides differences in terminology, it is also not uncommon for a user not to be able to describe accurately what he is looking for: the well known “I will know it when I see it” phenomenon. Furthermore, many terms in natural language are ambiguous. For example, a user querying for java might be looking for information about either the programming language, the coffee, or the island of Indonesia.

To satisfy users who expect search engines to come up with “what they mean and not what they say”, it is clear that more sophisticated techniques are needed than a straightforward returning of the documents that contain (one of) the query terms given by the user. One option is to adapt the query. Query refinement has already found its way to popular web search engines, and is even becoming one of those features in which search engines aim to differentiate in their attempts to create their own identity. Simultaneously with search results, Yahoo$^{3}$ shows a list of clickable expanded queries in an “Also Try” option under the search box. These queries are derived from logs containing

$^{3}$http://search.yahoo.com/
queries performed earlier by others. Google Suggest also uses data about the overall popularity of various searches to help rank the refinements it offers, but unlike the other search engines, the suggestions pop up in the search box while you type, i.e., before you search. Ask.com provides a zoom feature, allowing users to narrow or broaden the field of search results, as well as view results for related concepts.

In this chapter we focus on query expansion, i.e., the process of adding related terms to the query. This technique goes back a long way before the existence of the WWW. Over the last decades several important techniques have been established. The main idea underlying all of them, is to extend the query with words related to the query terms. One option is to use an available thesaurus, i.e., a term-term relation, such as WordNet, expanding the query by adding synonyms [27]. Related terms can also be automatically discovered from the searchable documents, taking into account statistical information such as co-occurrences of words in documents or in fragments of documents. The more terms co-occur, the more they are assumed to be related. In [28] several of these approaches are discussed and compared. In [2], correlations between terms are computed based on their co-occurrences in query logs instead of in documents. In this chapter we use both a crisp thesaurus in which terms are related or not, as well as a graded thesaurus in which terms are related to some degree.

Example 6 Table 6 shows a small sample fuzzy thesaurus $R$ based on the co-occurrences of terms in web pages found by Google [6]. $R$ is a fuzzy tolerance relation. The $T_W$-transitive closure $R^{[X^{-1}]}$ of $R$, i.e., the smallest $T_W$-transitive fuzzy relation in which $R$ is included, is shown in Table 7. In our running example, to compute upper and lower approximations, we will keep on using the t-norm $T_W$ as well as its residual implicator $I_{T_W}$. Finally, a crisp

### Table 6: Graded thesaurus.

<table>
<thead>
<tr>
<th></th>
<th>mac</th>
<th>computer</th>
<th>apple</th>
<th>fruit</th>
<th>pie</th>
<th>recipe</th>
<th>store</th>
<th>emulator</th>
<th>hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>mac</td>
<td>1.00</td>
<td>0.89</td>
<td>0.89</td>
<td>0.00</td>
<td>0.01</td>
<td>0.00</td>
<td>0.75</td>
<td>0.83</td>
<td>0.66</td>
</tr>
<tr>
<td>computer</td>
<td>1.00</td>
<td>0.94</td>
<td>0.44</td>
<td>0.44</td>
<td>0.56</td>
<td>0.25</td>
<td>1.00</td>
<td>0.83</td>
<td>0.99</td>
</tr>
<tr>
<td>apple</td>
<td>1.00</td>
<td>0.83</td>
<td>0.99</td>
<td>0.83</td>
<td>0.83</td>
<td>0.25</td>
<td>0.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fruit</td>
<td>1.00</td>
<td>0.44</td>
<td>0.66</td>
<td>1.00</td>
<td>0.00</td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pie</td>
<td>1.00</td>
<td>1.00</td>
<td>0.97</td>
<td>0.00</td>
<td>0.00</td>
<td>0.06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>recipe</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>store</td>
<td>1.00</td>
<td>0.34</td>
<td></td>
<td></td>
<td>0.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>emulator</td>
<td></td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hardware</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

---

4http://labs.google.com/suggest/
5http://www.ask.com/
6http://wordnet.princeton.edu/
Table 7: Transitive closure of graded thesaurus.

\[
R^8
\]

<table>
<thead>
<tr>
<th></th>
<th>mac</th>
<th>computer</th>
<th>apple</th>
<th>fruit</th>
<th>pie</th>
<th>recipe</th>
<th>store</th>
<th>emulator</th>
<th>hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>mac</td>
<td>1.00</td>
<td>0.89</td>
<td>0.89</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.89</td>
<td>0.89</td>
<td>0.89</td>
</tr>
<tr>
<td>computer</td>
<td>1.00</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>apple</td>
<td>1.00</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fruit</td>
<td></td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pie</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.99</td>
<td>0.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>recipe</td>
<td></td>
<td>1.00</td>
<td></td>
<td>0.99</td>
<td>0.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>store</td>
<td></td>
<td></td>
<td></td>
<td>1.00</td>
<td>0.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>emulator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hardware</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.00</td>
</tr>
</tbody>
</table>

(i.e., non-graded) thesaurus can be constructed by taking the 0.5-level of \( R \), defined as

\[(x, y) \in R_5 \text{ iff } R(x, y) \geq 0.5\] (52)

for all \( x \) and \( y \) in \( X \). In other words, in the crisp thesaurus, depicted in Table 8, two terms are related if and only if the strength of their relationship in the graded thesaurus \( R \) of Table 6 is at least 0.5. It can be easily verified that \( R_5 \) is not transitive. For example, \textit{fruit} is related to \textit{store} and \textit{store} is related to \textit{hardware}, but \textit{fruit} is not related to \textit{hardware}. For comparison purposes, in the remainder, we also include the transitive closure \((R_5)^8\).

Table 8: Crisp thesaurus.

\[
R_5
\]

<table>
<thead>
<tr>
<th></th>
<th>mac</th>
<th>computer</th>
<th>apple</th>
<th>fruit</th>
<th>pie</th>
<th>recipe</th>
<th>store</th>
<th>emulator</th>
<th>hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>mac</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>computer</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>apple</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>fruit</td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pie</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>recipe</td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>store</td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>emulator</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hardware</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Once the relationship between terms is known, either through a lexical aid such as \textit{WordNet}, or automatically generated from statistical information, the original query can be expanded in various ways. The straightforward way is to extend the query with all
the words that are related to at least one of the query terms. Intuitively, this corresponds to taking the upper approximation of the query. Indeed, a thesaurus characterizes an approximation space in which the query, which is a set of terms, can be approximated from the upper (and the lower) side. By definition, the upper approximation will add a term to the query as soon as it is related to one of the words already in the query. This link between query expansion and rough set theory has been established in [24], even involving fuzzy logical representations of the term-term relations and the queries. This approach works fairly well in general. When one or more of the query terms are ambiguous, it might fall short as we illustrate next.

**Example 7** Consider the query

\[ \text{apple, pie, recipe} \]

This is a difficult case for query expansion because of the ambiguity of the word *apple*, which can refer both to a piece of fruit and to a computer company. This query corresponds to a subset \( A \) of the universe of \( X \) of all terms, as shown in the second column in Table 9 under the heading \( A \). Let us start by examining the upper approximation of \( A \) under

<table>
<thead>
<tr>
<th>Term</th>
<th>( R \uparrow A )</th>
<th>( R \uparrow R \uparrow A )</th>
<th>( R^8 \uparrow A )</th>
</tr>
</thead>
<tbody>
<tr>
<td>mac</td>
<td>0.00</td>
<td>0.89</td>
<td>0.89</td>
</tr>
<tr>
<td>computer</td>
<td>0.00</td>
<td>0.94</td>
<td>0.99</td>
</tr>
<tr>
<td>apple</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>fruit</td>
<td>0.00</td>
<td>0.83</td>
<td>1.00</td>
</tr>
<tr>
<td>pie</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>recipe</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>store</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>emulator</td>
<td>0.00</td>
<td>0.25</td>
<td>0.99</td>
</tr>
<tr>
<td>hardware</td>
<td>0.00</td>
<td>0.99</td>
<td>0.99</td>
</tr>
</tbody>
</table>

the \( T \)-transitive fuzzy thesaurus \( R^8 \), i.e. \( R^8 \uparrow A \), shown in the last column. All the terms are added with high degrees, even though terms like *mac* and *computer* have nothing to do with the semantics of the original query. This process can be slowed down a little bit by using the non \( T \)-transitive fuzzy thesaurus and computing \( R \uparrow A \) which allows for some gradual refinement. However an irrelevant term such as *emulator* shows up to a high degree in the second iteration, i.e. when computing \( R \uparrow (R \uparrow A) \). The problem is even more prominent when using a crisp thesaurus as shown in Table 10.

### 4.2 Handling Ambiguous Query Terms

In [27] it is pointed out that the query expansion approach described above requires sense resolution of ambiguous words. Indeed, the precision of retrieved documents is likely to
Table 10: Upper approximation based query expansion with crisp thesaurus.

<table>
<thead>
<tr>
<th></th>
<th>$A$</th>
<th>$R_5 \uparrow A$</th>
<th>$R_5 \uparrow (R_5 \uparrow A)$</th>
<th>$(R_5 \uparrow)^8 \uparrow A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mac</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>computer</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>apple</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>fruit</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>pie</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>recipe</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>store</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>emulator</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>hardware</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

decrease when expanding a query such as apple, pie, recipe with the term mac. Even though this term is highly related to apple as a computer company, it has little or nothing to do with the intended meaning of apple in this particular query, namely a piece of fruit. An option to automate sense disambiguation is to only add a term when it is related to at least two words of the original query; experimental results are however unsatisfactory [27].

In [2], the most popular sense gets preference. For example, if the majority of users use windows to search for information about the Microsoft product, the term windows has much stronger correlations with terms such as Microsoft, OS and software, rather than with terms such as decorate, door and house. The approaches currently taken by Yahoo! and Google Suggest seem to be in line with this principle. Note, however, that these search engines do not apply query expansion automatically but leave the final decision up to the user. In [22], a virtual term is created to represent the general concept of the query. Terms are selected for expansion based on their similarity to this virtual term. In [28], candidate expansion terms are ranked based on their co-occurrence with all query terms in the top ranked documents.

The fuzzy rough set approach discussed below, first introduced in [6] and taken up also in [25], differs from all techniques mentioned above, and takes into account the lower approximation as well. The lower approximation will only retain a term in the query if all the words that it is related too are also in the query. It is obvious that the lower approximation will easily result in the empty query, hence in practice it is often too strict for query refinement. On the other hand, it is not hard to imagine cases where the upper approximation is too flexible as a query expansion technique, resulting not only in an explosion of the query, but possibly even worse, in the addition of non relevant terms due to the ambiguous nature of one or more of the query words. This is due to the fact that the upper approximation expands each of the query words individually but disregards the query as a whole.

However, it is possible to combine the flexibility of the upper approximation with the strictness of the lower approximation by applying them successively. As such, first the
query is expanded by adding all the terms that are known to be related to at least one of the query words. Next, the expanded query is reduced by taking its lower approximation, thereby pruning away all previously added terms that are suspected to be irrelevant for the query. The pruning strategy targets those terms that are strongly related to words that do not belong to the expanded query.

Example 8 The last column of Table 11 shows that the tight upper approximation is different from and clearly performs better than the traditional upper approximation for our purpose of web query expansion: irrelevant words such as mac, computer and hardware are still added to the query, but to a significantly lower degree. The difference becomes even more noticeable when using a crisp thesaurus as illustrated in Table 12.

Table 11: Comparison of upper and tight upper approximation based query expansion with graded thesaurus.

<table>
<thead>
<tr>
<th></th>
<th>$A$</th>
<th>$R\uparrow A$</th>
<th>$R^8\uparrow A$</th>
<th>$R\downarrow A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mac</td>
<td>0.00</td>
<td>0.89</td>
<td>0.89</td>
<td>0.42</td>
</tr>
<tr>
<td>computer</td>
<td>0.00</td>
<td>0.94</td>
<td>0.99</td>
<td>0.25</td>
</tr>
<tr>
<td>apple</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>fruit</td>
<td>0.00</td>
<td>0.83</td>
<td>1.00</td>
<td>0.83</td>
</tr>
<tr>
<td>pie</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>recipe</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>store</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.83</td>
</tr>
<tr>
<td>emulator</td>
<td>0.00</td>
<td>0.25</td>
<td>0.99</td>
<td>0.25</td>
</tr>
<tr>
<td>hardware</td>
<td>0.00</td>
<td>0.99</td>
<td>0.99</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table 12: Comparison of upper and tight upper approximation based query expansion with crisp thesaurus.

<table>
<thead>
<tr>
<th></th>
<th>$A$</th>
<th>$(R_5 \uparrow A$</th>
<th>$(R_5)^8 \uparrow A$</th>
<th>$R_5 \downarrow A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mac</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>computer</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>apple</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>fruit</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>pie</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>recipe</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>store</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>emulator</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>hardware</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
The main problem with the query expansion process as used in Example 7, even if it is gradual, is a fast growth of the number of less relevant or irrelevant keywords that are automatically added. This effect is caused by the use of a flexible definition of the upper approximation in which a term is added to a query as soon as it is related to one of its keywords. However, using the tight upper approximation as in Example 8, a term \( y \) will only be added to a query \( A \) if all the terms that are related to \( y \) are also related to at least one keyword of the query. First the usual upper approximation of the query is computed, but then it is stripped down by omitting all terms that are also related to other terms not belonging to this upper approximation. In this way terms that are sufficiently relevant, hence related to most keywords in \( A \), will form a more or less closed context with few or no links outside, while a term related to only one of the keywords in \( A \) in general also has many links to other terms outside \( R^{\uparrow} A \) and hence is omitted by taking the lower approximation.

This technique can be used both with a crisp thesaurus in which terms are related or not, as with a graded thesaurus in which terms are related to some degree. Furthermore it can be applied for weighted as well as for non-weighted queries. Whenever the user does not want to go through the effort of assigning individual weights to query terms, they are all given the highest weight by default. When a graded thesaurus is used, the query refinement turns the original query automatically into a weighted query. The original user-chosen terms maintain their highest weight, and new terms are added with weights that do not only reflect the strength of the relationship with the original individual query terms as reflected by the thesaurus, but also take into account their relevance to the query as a whole.

5 Summary

Fuzzy sets and rough sets each address an important characteristic of imperfect data and knowledge: while the former allow that objects belong to a set or relation to a given degree, the latter provide approximations of concepts in the presence of incomplete information. Fuzzy rough set theory aims to combine the best of both worlds. At the heart of this synergy lie well-chosen definitions of lower and upper approximations of fuzzy sets under fuzzy relations.

In a traditional Pawlak approximation space, indistinguishability is described by means of an equivalence relation \( R \). Well-known properties of the lower approximation \( R^{\downarrow} A \) and the upper approximation \( R^{\uparrow} A \) are recalled in Table 1. In a generalized approximation space, characterized by a tolerance relation \( R \), the tight lower approximation \( R^{\downarrow \downarrow} A \), the loose lower approximation \( R^{\downarrow \uparrow} A \), the tight upper approximation \( R^{\downarrow \uparrow} A \), and the loose upper approximation \( R^{\uparrow \uparrow} A \) are useful supplements to the traditional approximations. Their properties are summarized in Table 2.

All of these approximations can be generalized to approximate fuzzy sets in an approximation space characterized by a fuzzy tolerance relation \( R \). The preservation of previous properties depends on a careful choice of the fuzzy logical operators involved, as becomes
clear in Table 5. It is especially interesting to note that when $R$ is $T$-transitive, $T$ is a continuous t-norm and $I_T$ its residual implicator, then all three lower approximations coincide, as do all three upper approximations. However, this coincidence, and hence the $T$-transitivity of $R$, is not always desirable in applications, as an example about query refinement illustrates.

In this application, a query is perceived as a fuzzy set $A$ of terms, while $R$ is a fuzzy tolerance relation among terms. Extending the query with all the words that are related to at least one of the query terms corresponds to taking the upper approximation $R \uparrow A$. This query refinement technique works fairly well in practice, especially when using a non-transitive fuzzy thesaurus $R$ that allows for a gradual expansion process in subsequent steps. Finally, the tight upper approximation $R \downarrow \uparrow A$ helps to keep out irrelevant terms when one or more of the original query words are ambiguous.

**Acknowledgment**

Chris Cornelis would like to thank the Research Foundation–Flanders for funding his research.

**References**


