University of Nottingham
UK | CHINA | MALAYSIA

Novel Techniques for Modelling Uncertain Human Reasoning in Explainable Artificial Intelligence

Pasquale D'Alterio

Thesis submitted to the University of Nottingham
for the degree of Doctor of Philosophy

2017 - 2020

Abstract

In recent years, there has been a growing need for intelligent systems that not only are able to provide reliable predictions but can also produce explanations for their outputs. The demand for increased explainability has led to the emergence of explainable artificial intelligence (XAI) as a specific research field. In this context, fuzzy logic systems represent a promising tool thanks to their inherently interpretable structure. The use of a rule-base and linguistic terms, in fact, have allowed researchers to design models with a transparent decision process, from which it is possible to extract human-understandable explanations. The use of interval type-2 fuzzy logic in the XAI field, however, is limited: the improved performances of interval type-2 fuzzy systems and their ability to handle a higher degree of uncertainty comes at the cost of increased complexity that makes the semantic mapping between the input and outputs harder to understand intuitively. The presence of type-reduction, in some contexts fail to preserve the semantic value of the fuzzy sets and rules involved in the decision process. By semantic value, we specifically refer to the capacity of interpreting the output of the fuzzy system in respect to the pre-defined and thus understood linguistic variables used for the antecedents and consequents of the system. An attempt at increasing the explainability of interval type-2 fuzzy logic was first established by Garibaldi and Guadarrama in 2011, with the introduction of constrained type-2 fuzzy sets. However, extensive work needs to be carried out to develop the algorithms necessary for their practical use in fuzzy systems. The aim of this thesis is to extend the initial work on constrained interval type-2 fuzzy sets to develop a framework that preserves the semantic value throughout the modelling and decision process. Achieving this goal would allow the creation of a new class of fuzzy systems that show additional interpretable properties, and could further encourage the use of interval type-2 fuzzy logic in XAI. After the formal definition of the required components and theorems, different approaches are explored to develop inference algorithms that preserve the semantic value of the sets during the input-output mapping, while keeping reasonable run-times on modern computer hardware. The novel frameworks are then tested in a series of practical applications from the real world, in order to assess both their prediction performances and show the quality of the explanations these models can generate. Finally, the original definitions of constrained intervals type-2 fuzzy sets are refined to produce a novel approach which combines uncertain data and represents them using intuitive constrained interval type-2 fuzzy sets.

Overall, as a result of the work presented here, it is now possible to design constrained interval type-2 fuzzy systems that preserve the enhanced semantic value provided by constrained interval-type-2 fuzzy sets throughout the inference, type-reduction and defuzzification stages. This characteristic is then used to improve the semantic interpretability of the system outputs, making constrained interval type-2 fuzzy systems a valuable alternative to interval type-2 fuzzy systems in XAI. The research presented here has resulted in three journal articles, two of which have already been published in *IEEE Transactions*

on Fuzzy Systems, and four papers presented at the *FUZZ-IEEE* international conference between 2018 and 2020.

List of Publications

1. P. D'Alterio, J. M. Garibaldi and A. Pourabdollah, "Exploring Constrained Type-2 Fuzzy Sets," *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Rio de Janeiro, 2018, pp. 1-7 [Chapter 3]

2. P. D'Alterio, J. M. Garibaldi and R. John, "On the Concept of Meaningfulness in Constrained Type-2 Fuzzy Sets," *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, New Orleans, LA, USA, 2019, pp. 1-6 [Chapter 6]

3. P. D'Alterio, J. M. Garibaldi, R. John and A. Pourabdollah, "Constrained Interval Type-2 Fuzzy Sets," *IEEE Transactions on Fuzzy Systems*, 2020 [Chapter 3]

4. P. D'Alterio, J. M. Garibaldi and R. I. John, "Constrained Interval Type-2 Fuzzy Classification Systems for Explainable AI (XAI)," *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Glasgow, United Kingdom, 2020, pp. 1-8 [Chapter 5]

5. P. D'Alterio, J. M. Garibaldi, R. I. John and C. Wagner, "Juzzy Constrained: Software for Constrained Interval Type-2 Fuzzy Sets and Systems in Java," *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Glasgow, United Kingdom, 2020, pp. 1-8 [Appendix, Chapter A]

6. P. D'Alterio, J. M. Garibaldi, R. John and C. Wagner, "A Fast Inference and Type-Reduction Process for Constrained Interval Type-2 Fuzzy System," *IEEE Transactions on Fuzzy Systems*, 2020 [Chapter 4]

7. P. D'Alterio, J. M. Garibaldi and C. Wagner, "A Constrained Parametric Approach for Modelling Uncertain Data", *IEEE Transactions on Fuzzy Systems (under review)*, 2020 [Chapter 7]

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Explainable Artificial Intelligence

Intelligent systems have been widely adopted in recent years to tackle problems in a variety of fields, ranging from image classification to medical data analysis. Although state of the art models are able to produce reliable predictions, understanding their decision process may be very challenging. Many popular artificial intelligence (AI) tools like neural networks and deep learning, in fact, behave as *black boxes* [4]: when they receive the input values (i.e. the pixels of an image or the medical data of a patient), they extract and combine features non-linearly in order to identify patterns that determine the output; once the predictions are made, however, analysing how the input features have been combined in the decision process will provide information with a limited level of meaningfulness to a human user. In other words, although such models are capable of making reliable predictions, it is hard to understand *why* a specific prediction was made. Although in many contexts this may not represent a significant issue, in other situations it causes serious ethical and practical problems.

Specifically, in scenarios that significantly affect users, understanding the reasoning behind the choices of an AI model is required to ensure fair, non-discriminatory treatment, to validate the output of the system against experts'

knowledge and to detect any inconsistencies in the classification process [5, 6].

The medical domain is a typical example of a situation in which understanding the motivations that led to the final recommendation or prediction is vital to ensure that the patients receive the right treatment and that physicians can validate the choice made by the AI model to prevent harmful therapies being mistakenly advised.

The need for AI tools that can provide explanations for their decisions generated a new AI research field named *explainable artificial intelligence* (XAI) [7].

Its ambitious goal is to build a new generation of intelligent models that not only are reliable in their predictions but can also be intuitively interpreted by their end-users so that they can be deploy in contexts in which the transparency of the AI model is crucial.

So far in the literature, there are mainly two research areas focused on building explainable AI: one tries to "open" the black box models, extracting and analysing information from their decision process to make it more understandable by humans; the other research area makes use of AI techniques that have an inherently explainable design, based on meaningful concepts and structures (e.g words and rules) that mimic the thought process of humans in their everyday life.

1.2 Fuzzy Logic and Explainable Artificial Intelligence

Fuzzy logic is one of the tools with which it is possible to build AI models that are inherently explainable. It was introduced by Zadeh in 1965 [8] to represent classes that do not have clear boundaries, e.g. "the class of tall men" [8]. Fuzzy sets (FSs) are a key component of fuzzy logic and are based on the concept of *degree of truth*: they model classes "of continuum grade" [8]

by the use of membership functions that assign to each element a real number *between* 0 and 1. As Zadeh pointed out in his paper, this approach is useful as classes in the real world do not always have "precisely defined criteria of membership" [8]. Therefore fuzzy logic allows to directly and naturally model vague concepts used by humans when they speak or write, such as words. An example of fuzzy set modelling the words *medium height* is shown in Fig. 1.1. Having membership degrees as numbers between 0 and 1 makes it easy to describe a concept like this, for which there are no clear boundaries but a smooth transition from one class to the other (in this case, from *low height* to *medium height* to *tall height*).

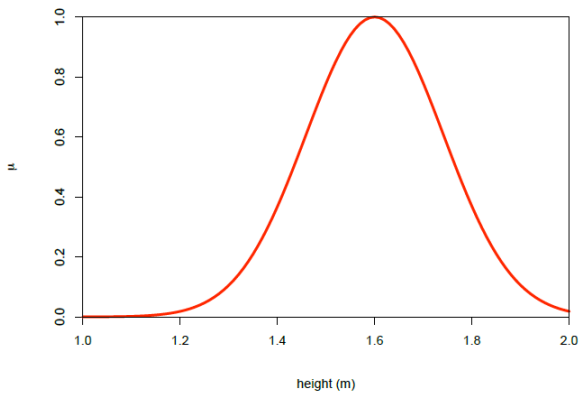


Figure 1.1: *A Gaussian FS modeling medium height. Height in m on the x-axis; membership degree on the y-axis.*

In addition to their modelling capabilities, FSs can be put together to form *fuzzy rules*. They are if-then statements, just like first-order-logic rules, used to model an inference process that creates an input-output mapping between the antecedents and the consequents. For example, the fuzzy rule "if the temperature is high then the speed of the fan is high", is used to establish a relation between the current temperature and the motor of the fan. One of the key advantages of the use of fuzzy logic is that it can be easily understood by humans, even non-experts in the field, since it makes use of the same concepts and structure of human reasoning. For these reasons, Zadeh himself defined the use of fuzzy logic as *computing with words* [9].

The modeling power of a single rule, however, is very limited. To tackle

many of the real-world problems, multiple rules are designed and organized in rule-bases.

The process of feeding the input values to the rule-base (fuzzification), performing the inference and then converting the fuzzy result into a number (defuzzification) is usually called fuzzy logic system (FLS).

Since their introduction, FLSs have been widely used in the literature, particularly in the area of controllers and intelligent systems.

One of the key advantages of the use of fuzzy logic is that it can be used to build interpretable systems. Thanks to the rule-based structure and the use of linguistic labels, FLSs inherently have all the characteristics to tackle the new challenge of XAI [10]. Understanding the decision process followed by the systems and how the inputs are combined to produce the outputs is intuitively easy to understand as it can be explained in human-understandable terms and even in natural language [11, 12].

However, the kind of fuzzy sets described so far, called type-1 (T1) fuzzy sets, have some limitations, specifically when it comes to the level of uncertainty that they can handle. In fact "it may seem problematical, if not paradoxical" [13] that the membership degrees of FSs are exact numbers, when the goal of fuzzy logic is to model vague or uncertain contexts. To overcome this issue, Zadeh himself introduced a new class of FSs, named type-2 fuzzy sets.

1.2.1 Type-2 Fuzzy Logic

There are many sources of uncertainty that need to be taken into account when modelling a FLS [14]. One of them regards the fact that words mean different things to different people and it is something that must be reflected in the design of fuzzy sets [15]. Type-2 (T2) fuzzy sets [16], thanks to the ability to express uncertainty around their membership function, are able to model this scenario.

The membership degree of each object is no longer a single value, but rather

an interval, in which each point can have a different weight in [0, 1].

Intuitively, T2 FSs can be seen as three-dimensional objects obtained by "blurring" a type-1 (T1) fuzzy set and giving a different weight to each of the blurred points. An example of a T2 FS is shown in Fig. 1.2, where the difference in the weighting is made clearer by the use of different colours (with blue being 0 and red being 1).

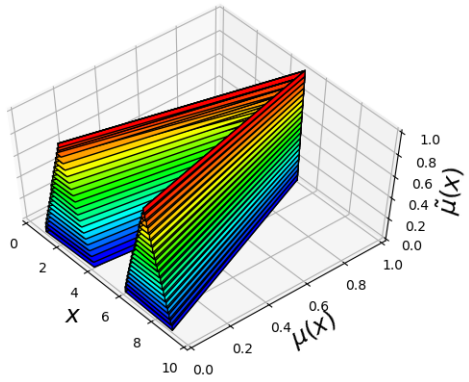


Figure 1.2: *A type-2 fuzzy set.*

The higher modeling capabilities of T2 FSs and FLSs, however, come with a significantly increased computational cost. Carrying out many of the fundamental operations, requires significantly longer run-times, which makes them less suitable for practical applications. Specifically, widely adopted defuzzification techniques require a preliminary step called type-reduction with a high computational complexity.

For this reason, a more efficient special case of T2 FSs, called *interval* type-2 (IT2) fuzzy sets, has recently become widely adopted [17]. In IT2 FSs, the membership degree is an interval where each point has weight *either* 0 or 1. Some popular research works [18, 19] have shown that this limitation is sufficient to make IT2 FLSs and the defuzzification more efficient than their T2 counterpart, making them good candidates for real-world application, especially in contexts with strict time constraints.

An example of an IT2 FS is shown in Fig. 1.3.

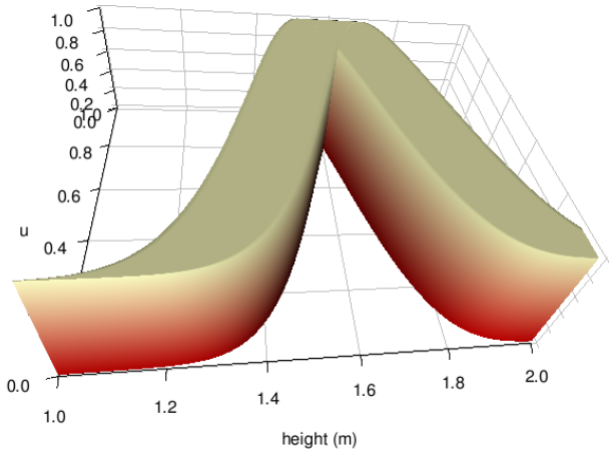


Figure 1.3: *An interval type-2 fuzzy set. Picture from [1].*

1.2.2 Interval Type-2 Fuzzy Sets and Semantic Difficulties

As the need for interpretable intelligent systems increased, some research works [1, 20, 21] started to analyse the semantic relation between T2 and IT2 FSs and the concepts they model. They have shown how the mathematical freedom given by the standard definition of T2 and IT2 FSs, may lead to results that are hard to interpret intuitively *in certain contexts*. Many T2 FSs in the literature are produced through a process of knowledge elicitation with a number of participants, asked to provide a T1 FSs for a specific word [1]. Some other times, T2 FSs are build starting from a T1 FS modelling the same concept, by the addition of uncertainty around the membership function through different "blurring" processes [22–30]. However, when the T2 FS is generated, there are no guarantees that it will preserve some properties, e.g. continuity, convexity and monotonicity, that may be crucial to preserve a semantic connection between the FS and the concept it models [1]. Additionally, during the type-reduction of T2 and IT2 FSs, *embedded sets* are taken into account: intuitively, they are T1 FSs that lie *within* a T2 FS. Popular type-reduction procedures (e.g. [18, 19] process all the embedded sets regardless of their shape, even the ones that are not meaningful in that specific context [1, 20, 21]. This phenomenon can make the decision process of a FLS harder to understand

intuitively, making T2 and IT2 FLSs less appealing for the use in XAI.

However, T2 and IT2 FLSs have been shown to outperform T1 FLSs in many tasks (e.g. [27, 31, 32]); therefore being able to increase the semantic value of their input-output mapping would represent a significant progress for the use of fuzzy logic in XAI.

1.2.3 Constrained Interval Type-2 Fuzzy Sets

Constrained type-2 (CT2) and constrained interval type-2 (CIT2) [1] fuzzy sets present a possible solution to mitigate the decreased interpretability of T2 FSs in certain contexts. Their goal is to preserve semantic meaning in the generation of a T2 FS when it is obtained from an already existing T1 FS modelling the same concept.

CT2 FSs represent a special case of T2 FSs as they impose additional constraints in order to restrict the possible shape of both the generated FS and its embedded sets. Keeping a *shape coherency* [1] throughout the modelling process is important for the interpretability of the model, as the shape is one of the key features from which humans intuitively understand the semantic meaning of a fuzzy set.

However, at the moment there is no systematic way to create CIT2 FSs for practical use and there has not yet been developed an inference framework that is able to preserve the semantic value guaranteed by the constrained approach.

1.3 Aims and Objectives

Recent research works [1, 20, 21] have shown that, in *some* contexts, the "mathematical freedom" [1] of the original definition of T2 FSs can cause some issues in the interpretability of T2 models: the practical and semantic difficulties in the determination of the shape of T2 FSs and the presence of embedded sets that are not in line with the intuitive interpretation of the modelled concept can cause a loss of the meaningful mapping between the input and outputs of

a T2 FLS.

The aim of this thesis is to extend the recently established foundational work on CIT2 FSs, specifically focusing on CIT2 FSs, to develop a framework that preserves the semantic value of CIT2 FSs throughout the inference, type-reduction and defuzzification in order to create a new class of CIT2 FLSs more intuitively explainable[1]. These FLSs will make the semantic mapping from the inputs to the output more intuitively interpretable, making them a valuable alternative to T2 and IT2 FLSs in XAI.

To achieve the aims stated above, this thesis pursues the following objectives:

1. *Formally define the generation of CIT2 FSs for practical use:* although the concept of CIT2 FSs has already been formulated, for them to be used in real-world applications it is necessary to formalize the definition of some key components to make the generation of CIT2 FSs in practice more systematic.

2. *From interpretable models to explainable systems:* CIT2 FSs guarantee a semantic connection between the sets and the words they model; to build interpretable CIT2 FLSs, however, it is important to develop an inference and defuzzification framework that preserves the semantic value of CIT2 FSs throughout the process. This property would ensure a semantic mapping between the inputs and the output of the FLSs and the ability to produce human-understandable explanations for the model predictions.

3. *Make CIT2 FLSs usable in practical applications:* for CIT2 FLSs to be usable in the real-world, it is necessary to design algorithms that

[1]Here and in the rest of the thesis, *interpretability* refers to the capacity of giving a semantic interpretation to the different components of the system (e.g. sets and rules representing respectively human concepts and relations between them), while *explainability* refers to the ability to explain in human-understandable terms the connection between the system components and the outputs or predictions produced (i.e. the decision process followed by the system).

carry out both the modelling and the inference process with reasonable run-times on modern computer hardware. It is therefore important to analyse the mathematical properties of the sets to produce efficient procedures and approximation algorithms that do not make the increased interpretability too expensive in terms of computational complexity. Furthermore, producing software libraries that implement these algorithms would be beneficial to facilitate the use of CIT2 FLSs in the research community, while possibly encouraging the use and discussion of CIT2 FSs in XAI.

4. *Validate the theory with real-world applications:* once all the necessary theory and algorithms have been laid out, it is necessary to test the novel CIT2 FLSs in a series of practical applications from the real world, in order to assess both their prediction performance and the perceived interpretability of the explanations provided when dealing with non-synthetic data.

1.4 Outline of the thesis

This thesis has the following layout: Chapter 2 presents a literature review on T1, IT2 and T2 fuzzy logic with a particular emphasis on their use in XAI. Some interpretability issues that may arise with the use of T2 fuzzy logic are also discussed, together with possible solutions proposed in the literature.

Chapter 3 is focused on the theoretical foundations of CIT2 FSs and FLSs. It introduces the core definitions and proofs that are necessary to build a first inference framework. Two defuzzification algorithms are designed and applied to a first case study, where CIT2 and IT2 FLSs are compared and contrasted with a focus on the interpretability of both models.

The two defuzzification algorithms proposed in Chapter 3, however, have the downside of being significantly slower than the most popular defuzzification approaches for IT2 FLSs. In Chapter 4 this issue is tackled: a new,

faster inference and defuzzification procedure for CIT2 FLSs is proposed and compared with the algorithms introduced in Chapter 3, showing its significant run-time improvement; additionally, it is discussed how the steps of the algorithm can be analyzed to generate natural-language explanations for each of the system predictions.

In Chapter 5, the algorithm introduced in Chapter 4 is then used in two real-world case classification from the medical domain, where it is shown how the explanations produced by CIT2 FLSs can be beneficial in the contexts in which transparent AI models are needed and it is discussed when CIT2 FLSs can represent a valid alternative to IT2 FLSs.

The concept of *meaningfulness*, as described in the first paper that introduced CT2 and CIT2 FSs [1], is refined in Chapter 6. The meaningfulness is decoupled from a specific shape and is described in terms of a set of contextual mathematical constraints that define what kind of fuzzy set can give an "acceptable" representation of a given concept. This new characterization of the concept of meaningfulness allows to create more flexible CIT2 FSs in which multiple shapes can coexist. Chapter 7 uses this new concept of meaningfulness in a data-modelling problem, in which opinions gathered from surveys are modelled using CIT2 FSs in a way that keeps a high interpretability while still showing the effects of the inter-variation on the FSs produced.

Chapter 6, instead, focuses on the relation between the concept of *meaningfulness* and the original CIT2 definitions. In the chapter, the meaningfulness is decoupled from the use of a specific shape. By analyzing some case studies it is shown how the traits underpinning the meaningfulness of a concept can be more naturally encoded through a set of mathematical constraints that define an "acceptable" representation is. These mathematical constraints, can then be used to restrict the shape of a CIT2 fuzzy set. This new characterization of the concept of meaningfulness allows to create more flexible CIT2 models in which multiple shapes can coexist. Chapter 7 uses this new definition of meaningfulness in a data-modelling problem, in which opinions gathered from

surveys are modelled through CIT2 fuzzy sets in an intuitive way, preserving the original structure of the data while showing the effects of the inter-expert variation on the models produced.

Chapter 2

Literature Review

2.1 Fuzzy Logic

In his famous paper [8] in 1965, Zadeh introduced the concept of fuzzy sets to model classes that do not have clear boundaries. Their peculiarity is that their membership functions assign to each object a value in the interval [0, 1] rather than either 0 or 1, as would happen in standard set theory. Formally, given a universe (also called *universe of discourse*) $X \subseteq \mathbb{R}$, a FS A can be expressed as:

$$A = \{(x, \mu_A(x)) | x \in X\} \tag{2.1}$$

where $\mu_A : X \mapsto [0, 1]$ identifies the membership function of the fuzzy set A.

All the points in the universe of discourse with a membership value greater than 0 constitute the *support set* of A:

Definition 2.1. *Given a FS A, its support set, here named $SUPP_A$ is the set of all the $x \in X$ for which $\mu_A(x) > 0$:*

$$SUPP_A = \{x \,|\, x \in X \wedge \mu_A(x) > 0\} \tag{2.2}$$

Since every FS is a set of pairs, they can easily be represented on the Cartesian plane, with the universe of discourse on the x-axis and the membership degree on the y-axis. Fig. 2.1 shows the difference between a fuzzy set (in red)

and a classical set (in purple). From the picture it is possible to see that in the fuzzy case the boundaries have a smooth transition compared to the sharp edges of the classical set.

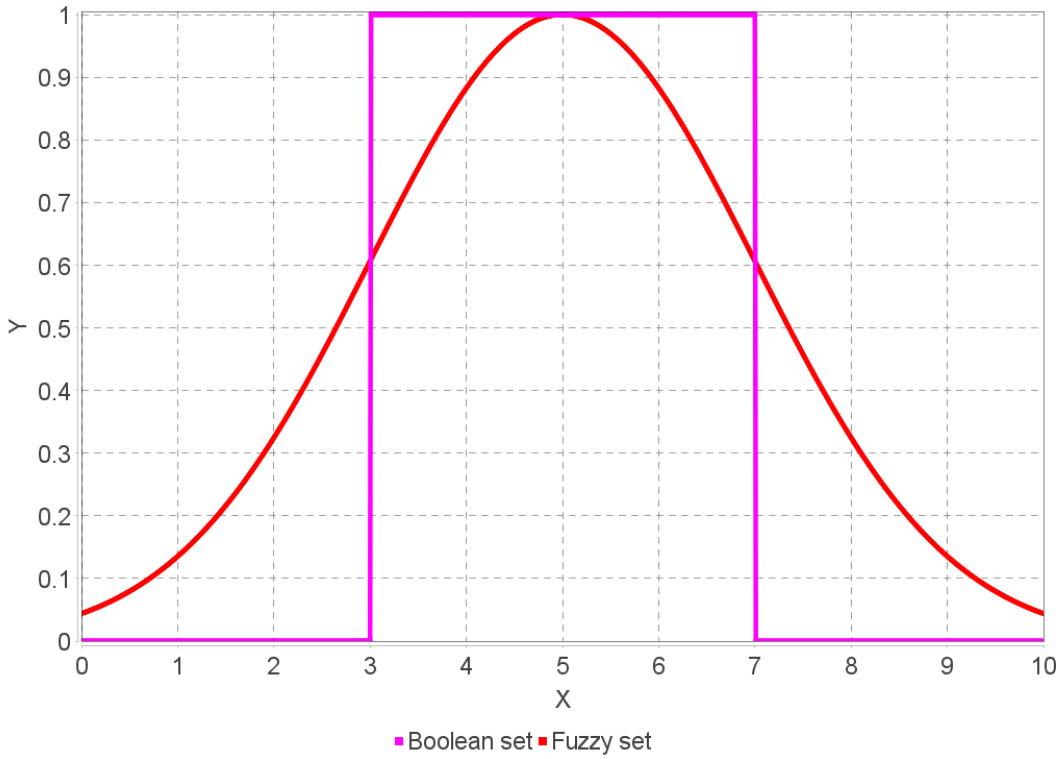


Figure 2.1: *Example of a fuzzy set (in red) and a crisp, i.e. boolean, set (in purple)*

The union, intersection and complement operations from classic set theory have been extended to be used for fuzzy sets [8, 14] as shown below:

$$\mu_{A\cup B}(x) = \mu_A(x) \oplus \mu_B(x),\ \forall x \in X \tag{2.3}$$

$$\mu_{A\cap B}(x) = \mu_A(x) \star \mu_B(x),\ \forall x \in X \tag{2.4}$$

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x),\ \forall x \in X \tag{2.5}$$

with A and B fuzzy sets, X the universe and $\oplus$ and $\star$ being respectively a t-conorm and a t-norm operator.

Definition 2.2. *A t-norm $\star$ is a binary function $[0,1] \times [0,1] \mapsto [0,1]$ that satisfies the following properties:*

- *Commutativity:* $a \star b = b \star a$

- *Monotonicity:* $a \star b \leq c \star d$ *if* $a \leq c \wedge b \leq d$

- *Associativity:* $a \star (b \star c) = (a \star b) \star c$

- *1 is the identity element:* $a \star 1 = a$

Two of the most widely used t-norm operators in fuzzy logic to implement the intersection ($\cap$) are the *minimum t-norm* and the *product t-norm*, i.e.:

$$\textbf{Minimum t-norm: } \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)), \forall x \in X \tag{2.6}$$

$$\textbf{Product t-norm: } \mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x), \forall x \in X \tag{2.7}$$

Definition 2.3. *A t-conorm* $\oplus$ *is a function* $[0,1] \times [0,1] \mapsto [0,1]$ *that satisfies the following properties:*

- *Commutativity:* $a \oplus b = b \oplus a$

- *Monotonicity:* $a \oplus b \leq c \oplus d$ *if* $a \leq c \wedge b \leq d$

- *Associativity:* $a \oplus (b \oplus c) = (a \oplus b) \oplus c$

- *0 is the identity element:* $a \oplus 0 = a$

One of the most widely used t-conorm operators in fuzzy logic to implement the union ($\cup$) is the *maximum t-conorm*, i.e.:

$$\textbf{Maximum t-conorm: } \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)), \forall x \in X \tag{2.8}$$

The choice of the specific t-norm and t-conorm operators can vary and depends, among other things, on the specific problem modelled or addressed.

2.2 Rules and rule-bases

With fuzzy logic, it is possible to create relations between fuzzy sets through rules. They are if-then statements like the first-order logic ones, for example:

$$\text{IF } x \text{ is } A \text{ THEN } y \text{ is } B \tag{2.9}$$

where A and B are FSs with universe X and Y, $x \in X$, $y \in Y$. The antecedent block can contain multiple fuzzy sets in conjunction or disjunction, for example "IF x is A AND y is B" or "IF x is A OR y is B". The "AND" and "OR" operators are implemented respectively with the fuzzy set intersection ($\cap$) and union ($\cup$).

Whenever the FSs and inputs involved model words, the rules can be expressed in a way that is very similar to human reasoning. For example, a controller that increases the speed of a fan when the temperature is high, could be implemented with fuzzy logic by the following rule: IF temperature is high THEN fan_speed is fast. This property has been defined by Zadeh as "computing with words" [9].

Since the operations that are necessary to carry out the inference, are implemented with functions that work on fuzzy sets, for the input values to be used they must be *fuzzified* (i.e.turned into fuzzy sets) by the *fuzzifier* [14]:

Definition 2.4. *Given a universe X, the fuzzifier maps a crisp point $x \in X$ into a fuzzy set A_x.*

When the input is a crisp number, the fuzzifier is called *singleton* fuzzifier [1] [14]:

Definition 2.5. *A singleton fuzzifier is one for which $\mu_{A_x}(x) = 1 \wedge \mu_{A_x}(x') = 0$; $x, x' \in X$, $x \neq x'$.*

Rules that produce fuzzy sets as outputs are known as Mamdani rules [33]:

[1] Only *singleton* fuzzification is described, as *non-singleton* fuzzification is beyond the scope of this thesis

once the antecedents have been evaluated, the *minimum t-norm* is used to carry out the implication between the fuzzy set obtained from the antecedent evaluation and the consequent set(s). Since the modelling capability of a single rule is limited, for practical applications multiple rules are designed and grouped in rule-bases. The fuzzy sets resulting from the evaluation of each Mamdani rule are grouped through the *union* (i.e. a t-conorm, usually implemented through the *maximum t-conorm*). Therefore, the global result is still a fuzzy set.

Since Mamdani rule-bases produce fuzzy sets as outputs, they must be turned into crisp numbers to be usable in practice: this process is called *defuzzification* [14]:

Definition 2.6. *A* defuzzifier *maps one or more fuzzy sets into a real number.*

The defuzzification of the result of Mamdani rule-bases can be carried out through different approaches.

One of the most popular in the literature is the centroid defuzzification:

Definition 2.7. *Given a fuzzy set A with a discrete universe of discourse X with n points, its centroid is computed as:*

$$Centroid(A) = \frac{\sum_{i=1}^{n} \mu_A(x_i) x_i}{\sum_{i=1}^{n} \mu_A(x_i)}, \tag{2.10}$$

Although it is a very popular defuzzification method and one of the first to be used in real-world applications [14], its computation can be time consuming.

The fuzzification, rule-base, inference and defuzzification processes are usually referred to as a *fuzzy logic system* (FLS). A scheme summarizing all the components of a FLSs and how they communicate is reported in Fig. 2.2.

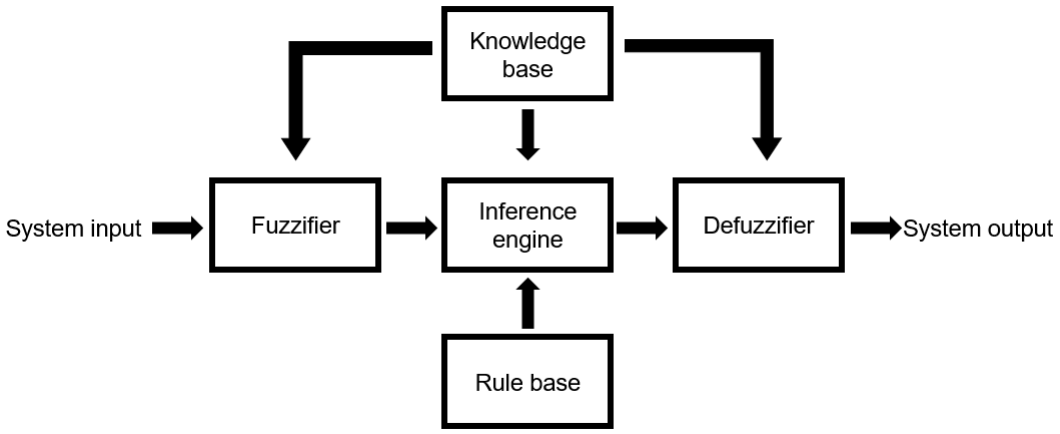


Figure 2.2: *Overview of the architecture of a fuzzy system.*

2.3 Fuzzy Logic in Explainable Artificial Intelligence

Fuzzy logic has some characteristics that make FLSs inherently understandable by humans. In fact, FSs can be designed to have a clear semantic meaning (e.g. when they model a word) while rules create an input-output mapping with a structure similar to human reasoning. Additionally, after a FLS produces an output, it is possible to analyse the rules of the rule-base that fired together with their antecedents, making it possible to gain valuable insight into the decision process of the system.

These properties have recently made FLSs a valuable tool for XAI applications in different areas, like the medical domain. Some fuzzy models have been shown to be usable to detect many diseases [34], including life threatening illnesses like lung [35] and breast [36] cancer. Other works analyzed the ability to produce textual explanations in natural language from FLSs [11, 12, 37].

Fuzzy sets have also been used to model the variation that naturally occurs in human decisions. In fact, if queried repeatedly, humans may provide different answers over time, as they "may have new or better information about the problem domain, may have forgotten specific details, may be in a different mood, etc." [38].

This non-deterministic behaviour has been modelled by non-stationary fuzzy sets [39]. They capture minor variations through random alterations of the membership function, according to a given perturbation function. Non-stationary fuzzy sets have been successfully applied to model expert-knowledge variation in intelligent systems [40, 41].

To improve the performance of interpretable FLSs [10], they are sometimes optimized with genetic algorithms [42] to tune a single component of a FLS (such as the FSs involved, the rulebase, the rule structure) or all of them.

Although FLSs can be used to build interpretable models, some concerns have been raised about the inherent interpretability of fuzzy logic [5, 10]. A FLS, in fact, is not intuitively understandable *per se*: the use of inappropriate shapes for the FSs, a rule-base with hundreds of rules or individual rules with a long chain of antecedent and consequents are all issues that may significantly decrease the interpretability of a fuzzy model. These aspects must be taken into account when designing a system, without assuming that the use of fuzzy logic automatically makes the system easy to understand.

Additionally, the kind of fuzzy sets described so far, i.e. type-1 (T1) fuzzy sets, can only handle a limited amount of uncertainty. In fact, their membership functions are very "crisp", as they assign a single number to every element in the universe of discourse. However, sometimes it is challenging to define the membership degree with such precision. For example, words usually have a slightly different meaning for different people. Therefore, for each element in the universe, different membership degrees could be chosen, depending on the background and taste of a specific person; it would be desirable, in this context, to model this uncertainty when designing the membership function.

2.4 Type-2 Fuzzy Sets

Type-2 (T2) fuzzy sets [16] were introduced by Zadeh to overcome the "crispness" of T1 memberships. Intuitively, they are obtained by "blurring" the

boundaries of T1 fuzzy sets. As a result, the membership degree is no longer a number, but an interval in which each point can have a different weight between 0 and 1. Formally, a type-2 fuzzy set (T2 FS) $\tilde{A}$ is denoted as:

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) \,|\, x \in X, u \in [0, 1]\} \tag{2.11}$$

where $\mu_{\tilde{A}} : X \times [0, 1] \mapsto [0, 1]$ is the membership function of $\tilde{A}$.

T2 FSs can be represented as three dimensional objectives (Fig. 2.3). The "third dimension" is used to represent the weight of the points given by the membership function to each pair (x, u).

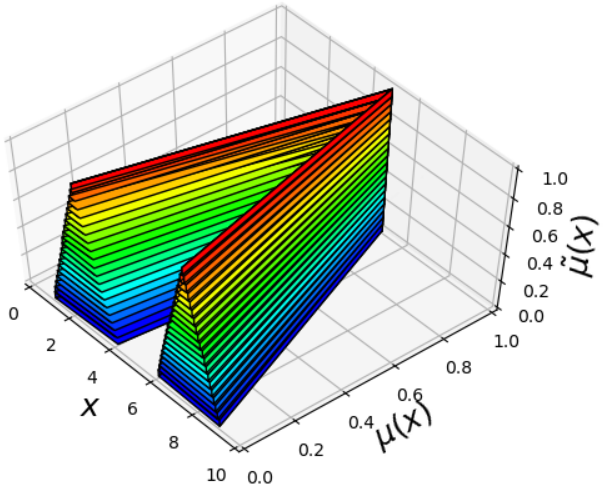


Figure 2.3: *A type-2 fuzzy set. The colours help identify the values in the third dimensions, going form blue to red, being respectively 0 and 1.*

Because of their more complex structure, they have additional characteristics compared to T1 sets. Two of them are the *primary* and *secondary membership* [14, 43, 44]:

Definition 2.8. *Given a T2 FS $\tilde{A}$ with universe X, the primary membership of x, J_x, is:*

$$J_x = \{(x, u) | u \in [0, 1],\ \mu_{\tilde{A}}(x, u) > 0\} \tag{2.12}$$

The primary membership J_x can be intuitively seen as the T1 FS that is obtained by slicing the T2 FS vertically at the value x.

Definition 2.9. *For every $x \in X$, and $u \in [0,1]$, the value of $\mu_{\tilde{A}}(x,u)$, also written as $\mu_{\tilde{A}(x)}(u)$, is called the* secondary grade *of x. For every $x \in X$, $\mu_{\tilde{A}(x)}$ is the secondary membership of $\tilde{A}$ with respect to x.*

Another key concept of T2 FSs is the *footprint of uncertainty*:

Definition 2.10. *Given a T2 FS $\tilde{A}$, its footprint of uncertainty (FOU) is the set of points (x,u) for which $\mu_{\tilde{A}}(x,u) > 0$:*

$$FOU(\tilde{A}) = \{(x,u) \,|\, (x,u) \in X \times [0,1], \mu_{\tilde{A}}(x,u) > 0\} \tag{2.13}$$

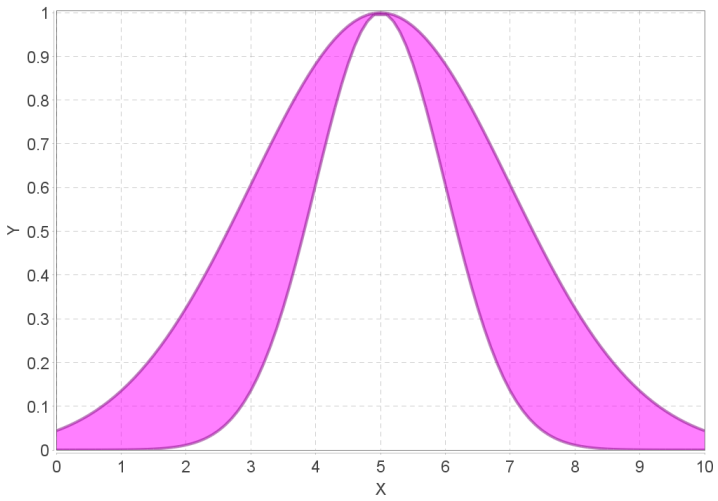


Figure 2.4: *Example of FOU (purple region).*

Intuitively, the FOU is the area representing the uncertainty in the membership function (e.g. Fig. 2.4), without taking into account the weight of each specific point. Intuitively, it is a 2D projection of the T2 FS. The boundaries of this area are called *upper* ($\overline{\mu}_{\tilde{A}}$) and *lower* ($\underline{\mu}_{\tilde{A}}$) membership function of the FOU [14] (Fig. 2.5):

$$\overline{\mu}_{\tilde{A}}(x) = \sup\{u | u \in [0,1],\ \mu_{\tilde{A}}(x,u) > 0\} \tag{2.14}$$

$$\underline{\mu}_{\tilde{A}}(x) = \inf\{u | u \in [0,1],\ \mu_{\tilde{A}}(x,u) > 0\} \tag{2.15}$$

Embedded sets (ESs) are another key concept related to T2 FS, since they

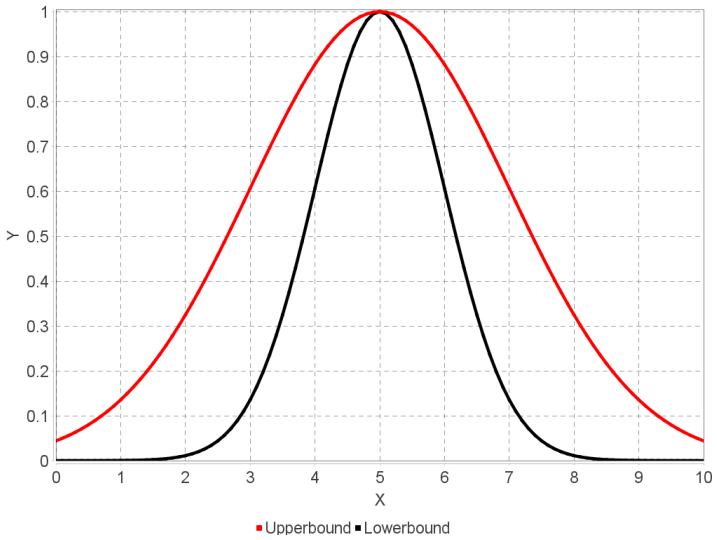


Figure 2.5: *Upper (in red) and lower (in black) bounds of the FOU in Fig. 2.4.*

have been widely used to obtain theoretical results for T2 FSs and are used in many fuzzy measures [44].

Definition 2.11. *[1] A type-2 embedded set (T2 ES), denoted $\tilde{A}_E$, is a path along the T2 set it belongs to. It contains only one primary degree u_x for each x, with its associated secondary grade v_x:*

$$\mu_{\tilde{A}_E}(x, u_x) = v_x \quad x \in X, u_x \in J_x \tag{2.16}$$

Definition 2.12. *[44] A type-1 embedded set (T1 ES), denoted A_E represents a projection of a T2 ES, i.e. its secondary degree has been dropped. Therefore it contains one primary degree u_x for each x:*

$$\mu_{A_E}(x) = u_x;\ \mu_{A_E}(x, u_x) = v_x \quad x \in X, u_x \in J_x \tag{2.17}$$

The defuzzification of a T2 FS (i.e. its conversion into a number) usually involves a *type-reduction* step. Its goal is to map a T2 FS into a T1 FS which can then be defuzzified using T1 defuzzification procedures.

The type-reduction is carried out following the procedure shown in Algorithm 1 (rephrased from [45]). The algorithm requires the evaluation of all the embedded sets of the discretized T2 FS and is for this reason sometimes

called *exhaustive method*. As the number of embedded sets increases exponentially with the number of discretization points used, this procedure is very computationally expensive and limits the applicability of T2 FLSs in practice.

For this reason, type-reduction has become a very active research area [14]. Recently, some approximation procedures have been proposed, that significantly reduce the run time of T2 FLS.

Algorithm 1 Type-reduction for type-2 fuzzy sets (rephrased from [45])

Given a type-2 fuzzy set $\tilde{A}$

for each embedded set E of the type-2 fuzzy set $\tilde{A}$ **do**

Find the minimum secondary membership grade, z

Calculate the primary domain value x of the centroid of E

Pair z with x (some x values may be paired with more than 1 z value)

end for

for each primary domain value x **do**

Keep only the pair (x, z) with the maximum secondary grade z

end for

The final (x, z) pairs represent the type-reduced set of $\tilde{A}$

Greenfield et al.[46] presented the sampling approach. The authors claim that a single embedded set has a very little impact on the final T2 centroid output. Therefore, they developed an approximation algorithm that only considers a random subset (of fixed cardinality) to carry out the type-reduction. They show how for a random sample of at least 1000 embedded sets the difference with the exhaustive method is negligible while the running time is drastically reduced (up to 10 000 times faster).

A different approach is represented by the use of $\alpha-$ planes [47, 48]. Conceptually, the idea is to discretize ("cut") the T2 FS along the third dimension, generating a set of planes. The centroids of the planes, can then be used to compute the centroid of the T2 FS they belong to. An equivalent approach (zSlices) has been proposed by Wagner and Hagras [49].

A conceptually similar idea has been used by John [50] with the vertical slice centroid type-reduction. A given T2 FS is cut into a number of vertical slices and each one of them is defuzzified as a T1 set. Then the domain (x) value of the slice is paired with this defuzzified value to build the type-reduced set. This method, however, is highly intuitive since no mathematical proof has been given for its validity.

2.5 Interval Type-2 Fuzzy Sets

Although some approximation procedures have been introduced to reduce the time complexity of T2 FLSs, they still remain significantly slower than their T1 counterpart. For this reason, a more efficient special case of T2 FSs is usually used in the literature: *interval* type-2 (IT2) FSs.

Intuitively, IT2 membership functions assign a weight of either 0 or 1 to every point generated by the "blurring". Therefore, the third dimension can be dropped as they can be fully represented by the FOU with its upper and lower membership functions. Formally, an IT2 FS is defined as follows:

Definition 2.13. *[17] An interval type-2 fuzzy set (IT2 FS), denoted $\tilde{A}$, is characterized by an IT2 MF $\mu_{\tilde{A}} : X \times [0,1] \mapsto \{0,1\}$ (i.e. $\mu_{\tilde{A}}(x,u)$ is either 0 or 1):*

$$\tilde{A} = \{((x,u), \mu_{\tilde{A}}(x,u)) \,|\, x \in X, u \in [0,1]\} \tag{2.18}$$

The inference process and set operations such as the union and intersection, can be carried out using T1 mathematics working with the upper and lower memberships of the FOU only [14, 17], which simplifies the execution of IT2 FLSs compared to general T2 FLSs:

Definition 2.14. *Given two IT2 FS $\tilde{A}$ and $\tilde{B}$ the lower memberships of the IT2 FSs resulting from their union and intersection can be computed as follows:*

$$\underline{\mu}_{\tilde{A} \cup \tilde{B}}(x) = \underline{\mu}_{\tilde{A}}(x) \oplus \underline{\mu}_{\tilde{B}}(x) \tag{2.19}$$

$$\underline{\mu}_{\tilde{A}\cap\tilde{B}}(x) = \underline{\mu}_{\tilde{A}}(x) \star \underline{\mu}_{\tilde{B}}(x) \tag{2.20}$$

with $\star$ and $\oplus$ being respectively a t-norm and a t-conorm. The upper membership of $\tilde{A} \cup \tilde{B}$ and $\tilde{A} \cap \tilde{B}$ can be computed analogously.

Also the type-reduction step for IT2 is simpler to compute, compared to general T2 FSs. Specifically, when type-reducing an IT2 FSs with a continuous FOU, the resulting T1 FS R is fully identified by the interval $[l, r]$, as its membership function is the following:

$$\mu_R(x) = \begin{cases} 1, & x \in [l, r] \\ 0, & \text{otherwise} \end{cases} \tag{2.21}$$

The values l and r are respectively the lowest and highest centroid among all the embedded sets of the IT2 FS to defuzzify. The most widely used algorithms to compute l and r are the Karnik-Mendel (KM) [18] and the enhanced Karnik-Mendel (EKM) [19] procedures. They mathematically model the type-reduction problem in an efficient iterative procedure that converges in a finite number of iterations.

Once the type-reduced set has been obtained, the centroid defuzzification is used to generate the final crisp value. Since the type-reduced set is identified by the interval $[l, r]$, its centroid is easily computed as its midpoint $\frac{l+r}{2}$.

2.6 Interpretability Issues of Type-2 Fuzzy Logic

As already mentioned in the Introduction, some research papers analyzed how the use of T2 and IT2 FSs and FLSs can impact the overall interpretability of fuzzy models. One of the issues that can arise in some contexts, concerns the loss of semantic value when embedded sets are taken into account [1, 20, 21].

Thanks to the representation theorem [44], a T2 (or IT2) FS can be expressed as the union of its T2 embedded sets. However, it is often the case that the embedded sets do not plausibly model the concept represented by the

T2 FS they belong to.

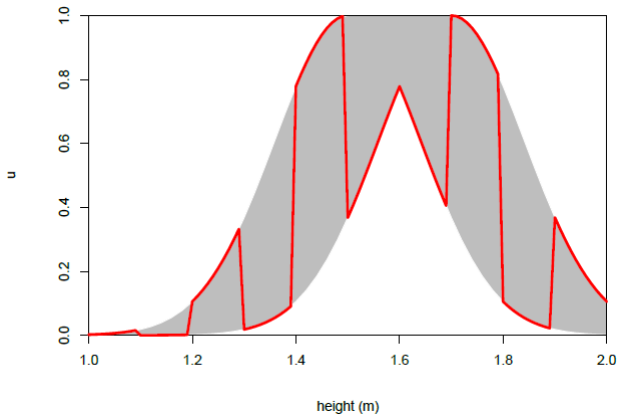


Figure 2.6: *One of the embedded set of the FOU shown (picture from [1])*

For example, the IT2 FS in Fig. 2.6 models the concept of *medium height*. The T1 embedded set in red, although mathematically acceptable, could hardly carry the same semantic meaning due to its shape. It seems reasonable to expect an IT2 FS modelling *medium height* to be a collection of different but plausible T1 embedded sets modelling the same concept. However, in the original mathematical definition of a T2 FS, this semantic connection cannot be preserved. This is a significant semantic issue in some contexts, since the shape plays an important role in the interpretation that humans give to a set.

This phenomenon becomes increasingly problematic when one takes into account fuzzy operators that make extensive use of the embedded set. For example, the exhaustive method for type-reduction processes all the embedded sets. Although other procedures (e.g. KM and EKM) do not process individual embedded sets, they directly compute l and r. These two values, however, represent the centroids of two of the embedded sets of the IT2 FS to type-reduce. Therefore, when l and r come from two embedded sets that do not carry a clear semantic meaning, like the one shown in Fig. 2.6, providing a human-understandable explanation for the type-reduction and defuzzification becomes very challenging. In other words, understanding intuitively *why* and *how* two specific l and r values have been produced is not straightforward. This problem also affects FLSs, as the type-reduction and defuzzification are

the last two steps for the computation of the final output. Consequently, even if the FSs used have a clear semantic meaning and the rule-base model is similar to the structure used in human reasoning, providing an explanation for the final system prediction is very challenging due to the presence of the type-reduction and defuzzification that decrease the overall interpretability.

The explainable T2 FLSs in the literature mainly perform classification tasks by solely identifying the single-consequent rule with the highest firing to make their predictions [11, 12, 35]. Although this system bypasses the issues that may arise from the type-reduction and defuzzification it also has some limitations. Specifically, having a model in which a single rule (the one with the highest firing strength) contributes to the final result may be not be sufficient to model complex problems.

A recent paper [51] proposed a novel method that shows which rules contributed to the final prediction of an IT2 FLS after the type-reduction and defuzzification. Although this approach improves the global interpretability of IT2 FLSs, a gap remains between the rules that fired and the way in which the crisp output was generated. In other words, although this approach provides a better insight on the decision process, it does not fully solve the issue generated by the lack of interpretability of the type-reduction. In fact, although knowing which rules fired helps to understand how the system is reasoning, it still does not fully explain in a meaningful manner how the l and r values are obtained from the type-reduction: the same set of firing rules does not always correspond to the same type-reduced set, as other components need to be taken into account, such as the firing strength and the impact on the final output for each rule.

2.6.1 Well-Shaped Interval Type-2 Fuzzy Sets

Some research work [1, 21] tackled the lack of interpretability of the type-reduction by imposing additional restrictions on the original definition of T2

or IT2 FSs. Specifically, the main goal was to completely remove the embedded sets that could hardly carry a semantic meaning while keeping the ones with a shape that could plausibly represent the concept modelled by the T2 FS.

Well-shaped IT2 FS [21] achieve this aim by imposing two additional mathematical properties, convexity and normality, to all the embedded sets of IT2 FSs.

Definition 2.15. *An T1 FS A is normal if and only if* $\exists x \in X : \overline{\mu}_A(x) = 1$

Definition 2.16. *A T1 FS A is convex if and only if :*

$$\mu_A(\delta x_1 + (1-\delta)x_2) \geq \min\left(\mu_A(x_1), \mu_A(x_2)\right), \forall x_1, x_2 \in X \text{ and } \delta \in [0,1] \quad (2.22)$$

The reason behind this choice is that in the vast majority of the practical applications of fuzzy logic, all the fuzzy sets used present these two characteristics, especially when they model meaningful concepts such as words.

By limiting the shapes of the embedded sets, this approach also limits the possible shapes of the FOU to the ones that can be fully covered by only convex and normal embedded sets (from here, the name *well-shaped* sets).

Definition 2.17. *(Adapted from [52]) Let $\tilde{A}$ be an IT2 fuzzy set; let $[b, c]$ be the top base of its upper membership $\overline{\mu}_{\tilde{A}}$ and $[f, g]$ the top base of its lower membership $\underline{\mu}_{\tilde{A}}$.*

$\tilde{A}$ is well shaped if and only if:

1. *$\tilde{A}$ is normal and convex*
2. *$f \geq b \wedge g \leq c$ i.e., the top base of the lower membership is completely within the top base of the upper membership.*

In a recent paper [52], the authors created a new well-shaped representation theorem from which they extend some of the most popular fuzzy uncertainty measures to their well-shaped IT2 FSs.

Although this novel approach represents a first step in tackling the interpretability issues caused by the unrestricted shape of the embedded sets, it

creates a very strong connection between convexity, normality and the meaningfulness of a set. However, as discussed later in this thesis (Chapter 3 and 6), the meaningfulness of a set is a property that is heavily context dependant. There are many cases in which convexity and normality alone fail to guarantee that the shape of a set will be meaningful. For example, to model the word *medium* additional properties might be required. A plausible representation, would involve a fuzzy set with a membership function that monotonically increases up to a point or a plateau, before monotonically decreasing. However, in well-shaped FSs it is not possible to impose these additional properties.

Furthermore, in other contexts non-normal sets still have a semantic connotation. For example, the fuzzy outputs (i.e. before the defuzzification step) of T1 Mamdani system are rarely normal. However, by analysing them it is possible to extract valuable information about the decision process of the FLS (e.g. the firing of the rules and which consequent sets contributed to the final result).

2.6.2 Constrained Type-2 Fuzzy Sets

Constrained type-2 (CT2) fuzzy sets [1] represent an alternative approach to address the problem caused by the unrestricted shape of the embedded sets.

Garibaldi et al. [1] proposed a systematic way to create T2 FSs starting from meaningful T1 FSs while preserving the semantic value throughout the process. In contrast with the well-shaped approach, in the CT2 case, the shape of the starting T1 FS, called *generator set*, is assumed to be meaningful. Therefore, only the embedded sets that have the same shape as the generator set are considered as *acceptable* and processed by the fuzzy operators, including the type-reduction. A similarity relation is used to determine the weight of each pair (x, u), $x \in X, u \in [0, 1]$: the closer the point (x, u) is to the generator set, the higher the weight. Intuitively, CT2 FSs can be seen as a way to model a T1 FS with uncertainty around its location on the x-axis. Similarly to the

well-shape shaped approach, also CT2 indirectly limit the possible shape of the FOU to the ones that are fully representable by the *acceptable* embedded sets.

Formally, a CT2 FS is defined as follows (rephrased from [1]):

Definition 2.18. *A CT2 FS $\tilde{A}$ is a T2 FS denoted by $\mu^{\delta}_{\tilde{A}} : X \times [0,1] \mapsto [0,1]$ that is built from a generator T1 FS A with $\mu_A : X \mapsto [0,1]$ and a similarity relation $\delta : X \times X \mapsto [0,1]$ (that captures the imprecision related the value of X). And it is built as follows:*

$$\mu^{\delta}_{\tilde{A}}(x,u) = \sup_{u=\mu_A(y)} \delta(x,y),\ x,y \in X \tag{2.23}$$

When the similarity function returns either 0 or 1, the set generated is called constrained *interval* type-2 (CIT2) fuzzy set. A CIT2 obtained from a Gaussian generator set is shown in Fig. 2.7. Some of the acceptable embedded sets are shown in black within the footprint of uncertainty. All the acceptable embedded sets, are translated version of the generator set along the x-axis, within the FOU.

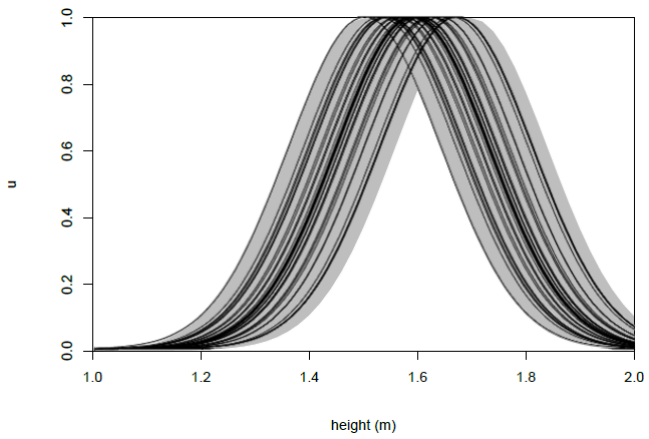


Figure 2.7: *Example of a CIT2 FS (picture adapted from [1])*

The constrained approach, compared to the well-shaped one, focuses more on the importance of a specific shape to model a given concept, rather than convexity and normality. On one hand, this gives the constrained approach the flexibility to use, for each concept, the shape that better represents its semantic

meaning. On the other hand, choosing a single shape may be limiting in some situations, as some concepts may be plausibly modelled by multiple shapes. For example, the word *medium* could be modelled by both Gaussians and triangles but this scenario is not allowed in the CT2 representation.

2.7 Reflection on the state of the art

Being able to design interpretable IT2 intelligent systems would represent an important step forward for the use of fuzzy logic in XAI, as IT2 logic has been shown to outperform T1 in many scenarios while also offering higher modelling capabilities and handling of uncertainties.

CIT2 FSs represent a promising tool to increase the interpretability of IT2 models as they focus on preserving the semantic connection between the sets and the concept they model through the use of meaningful shapes. Although the idea of CIT2 FSs has already been introduced, there are some limitations in the state of the art that do not make the CIT2 representation usable in practice. At the moment, there is no systematic way to create CIT2 FSs once the generator set has been determined. It would be beneficial, for the design of CIT2 FSs, to have a formal analysis of how the generator set, the FOU and the acceptable embedded sets are mathematically related, in order to make CIT2 FSs implementable in software. Furthermore, there has not yet been developed an inference framework that is able to preserve the semantic value guaranteed by the constrained approach. As a result, the additional interpretable properties of CIT2 FLSs would be lost if used with standard IT2 inference and defuzzification methods. A particular effort should be made in the design of a new type-reduction step, as it is the component that breaks the semantic connection in the input-output mapping of IT2 FLSs.

Lastly, there are no studies that involve real-world applications of CIT2 FLSs. Comparing and contrasting this new class of sets with other approaches is crucial to assess both the perceived interpretability and performance of this

new modelling method. This is a necessary step to understand in which contexts CIT2 FLSs represent a valuable alternative to the standard IT2 modelling approach.

Chapter 3

An Inference Framework for Constrained Interval Type-2 Fuzzy Sets

3.1 Introduction

Whilst the current T2 and IT2 framework has shown to have many advantages over T1 approaches, particularly in their ability to exhibit greater performance in most situations, there are drawbacks. Two properties which may decrease the overall interpretability of T2 systems are: (i) there is currently no agreed mechanism to derive the FOU, particularly in the situation in which a concept being modelled by a T1 set has uncertainty added to form a T2 set representing the same concept; and (ii) embedded sets (ESs) may have any shape, including ones which bear no relationship to the concept being modelled.

To overcome these issues, Constrained Type-2 (CT2) fuzzy sets have been proposed [1]. The idea behind them is to address the two limitations above by: (i) providing an explicit method for generating the boundaries of the footprint of uncertainty that keeps a *shape coherency* [1] throughout the generation of the type-2 set, based on an underlying concept modelled by a type-1 set; and (ii) restricting the acceptable embedded sets that may be used to only a

subset of all the ESs, in order to process only shapes that may be considered meaningful in that specific context. Even though the concept of CT2 FS has already been formulated [1], some key components are currently lacking formal definitions such as the acceptable embedded sets, constrained inference and centroid defuzzification.

This chapter, will provide some theoretical underpinning for this new constrained representation, focusing specifically on constrained interval type-2 (CIT2) fuzzy sets and contributing towards the objectives 1 and 2 of this thesis. In addition to formal definitions, a full inferencing and defuzzification framework is then proposed for the creation of CIT2 Mamdani-style fuzzy inference systems. Next, the CIT2 approach is compared with the recent framework introduced by Wu et al [21, 52] for creating 'well-shaped' type-2 sets. Finally, a practical application will be shown and compared with the conventional IT2 representation in terms of interpretability and explainability of the outputs, performance and run-times. Specifically, a genetic architecture will be described for the automatic generation of CIT2 fuzzy systems which is tested on two real world data-sets. Whilst interpretability is itself a difficult and complex concept to define, and is somewhat subjective in nature, nevertheless worked examples and the practical applications are shown to illustrate ways in which interpretability is enhanced. Throughout, it is stressed that the proposed CIT2 approach, which may be used in contexts in which explainability and interpretability are considered important, is an *alternative* to other approaches including the conventional type-2 approach.

3.2 Motivation

In the literature, there are three main approaches to determine the upper and lower bounds of the FOUs of T2 FSs when starting from already existing T1 MFs modeling the same concept. The first one identifies the two boundary MFs by taking the parameters of the existing T1 MFs and adding some uncertainty to them [25–30]. For example, in the case of a T1 Gaussian with mean m and variance v, the upper and lower bounds of the FOU could be the Gaussians with mean m and variances $v-k$ and $v+k$ respectively, with k being a positive real number.

A different method defines the FOU as the area covered by the translation along the x-axis of the starting T1 MF by a factor c and $-c$, $c \in \mathbb{R}$ [22–24]. The result is a symmetrical blurring around the starting T1 MF. An example of an FOU obtained with this approach with a T1 Gaussian can be seen in Fig. 3.1.

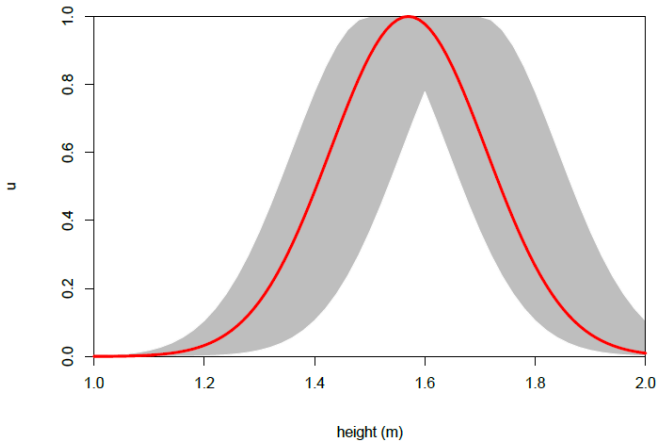


Figure 3.1: *In red, one of the embedded sets of the interval type-2 fuzzy set in grey (picture from [1])*

Another approach has also been proposed. It models the FOU so that it embeds all the T1 MFs obtainable from observations [20] or from the modeling of the same concept under different circumstances [53].

All those methods have in common the fact that they identify some T1 shapes as "meaningful" in their context and then use them to build the FOUs. However, when some fuzzy operators such as the Karnik-Mendel (KM) type-

reduction algorithm [18] are used, all the ESs are processed, regardless of their shape. As a consequence of that, ESs that could hardly represent the concept they are modelling, will likely determine the end-point of the defuzzified centroid. Since those ESs have a low interpretability due to their shape, the explainability of the output and, consequently, of the fuzzy system or set that generates it, decreases. However, in the recent years building explainable intelligent systems has become increasingly important [7, 54]. The following examples support these claims. Suppose that one decides to model the concept of *medium height* using a T1 Gaussian MF, as shown in Fig. 3.2. This set is named T1 generator set (GS). If one wants to build an IT2 FS from that, a possible approach would be to ask different people to place the mean of the Gaussian on the x-axis, after its variance value had been previously determined (similar approaches can be found in [55, 56]).

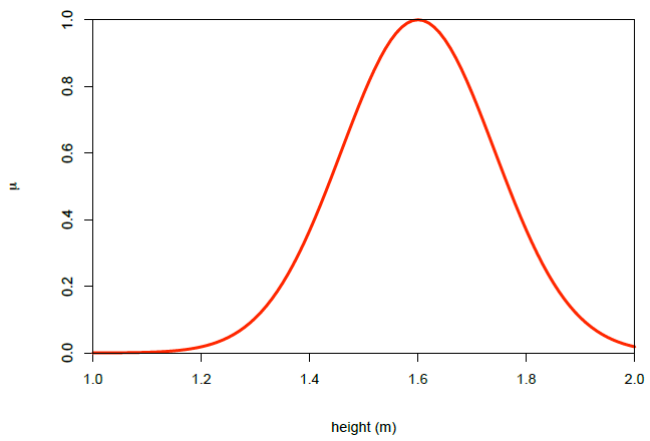


Figure 3.2: *T1 Gaussian MF (picture from [1])*

It is likely that something similar to what is shown in Fig. 3.3 would be obtained, since the concept of *medium height* would vary slightly from person to person. Now this collection of T1 MFs can be used to determine the FOU of the IT2 FS.

As in [53], those sets will be embedded in the FOU. To do so, the translation method mentioned above will be used, i.e. the FOU will be defined as the area covered by the shifting of the GS from the leftmost to the rightmost Gaussian to embed. The result of this operation is shown in Fig. 3.4.

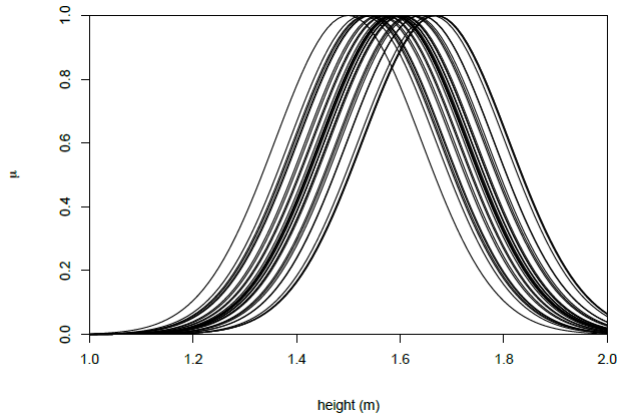


Figure 3.3: *Possible result of the thought experiment described above (picture from [1])*

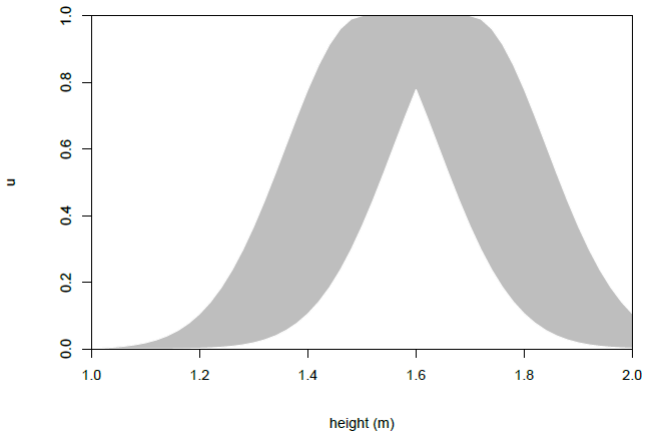


Figure 3.4: *FOU of a possible IT2 FS modelling medium height (picture from [1])*

If the standard IT2 representation is used, the ESs within the FOU can have arbitrary shapes. That makes even the ES shown in Fig. 3.5 acceptable. In this particular context, it is clear that a T1 ES like that has very little relation with the concept of *medium height*. In fact, no observation of the participants' opinion during the experiment led to such shape. Furthermore, this representation affects the centroid value and its explainability. The set shown in Fig. 3.6 has been obtained with the process described in the thought experiment above.

If the KM procedure [18] is used to type-reduce it, the algorithm will find the two ESs that give us the left and right endpoints of the centroid. For the IT2 FS in Fig. 3.6, the results are show in in Fig. 3.7.

These sets do not seem to fit this case very well. That is because, to obtain

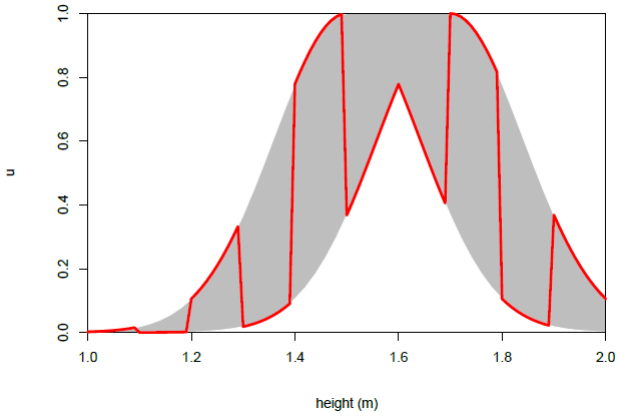


Figure 3.5: *One of the embedded set of the FOU shown (picture from [1])*

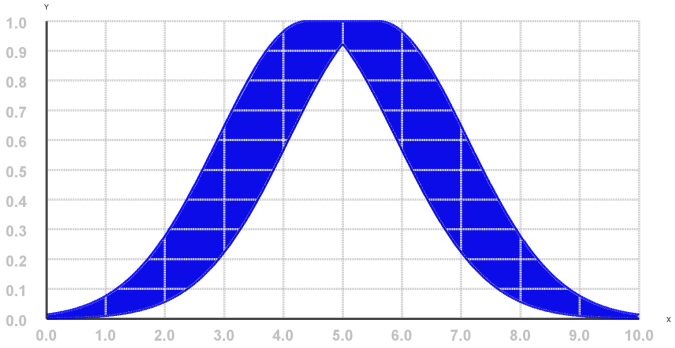


Figure 3.6: *Possible FOU generated from a Gaussian T1 MF*

the type-reduced value, the algorithm chose two ESs that did not represent any of the observations made during the experiment; additionally, those shapes could hardly represent the concept of *medium height* that is being considered.

System output defuzzification represents another useful example to see how the standard IT2 representation affects the interpretability and explainability of fuzzy systems. Consider, for example, the fuzzy output set shown in Fig. 3.8, and its associated left and right endpoints shown in Figs. 3.9 and 3.10, respectively. In Fig. 3.9, the embedded sets of the left endpoint derived using the constrained centroid (Fig. 3.9(a)), and the KM procedure (Fig. 3.9(b)) are compared. Similarly, Fig. 3.10 compares those of the right endpoint. The ES used for the constrained centroid preserve the same level of interpretability of T1 system outputs in that the shapes of the generator sets are clearly identifiable and so are the firing strengths that generated them. As a consequence of this, it is possible to get an intuitive idea of the sets that lead to the end-

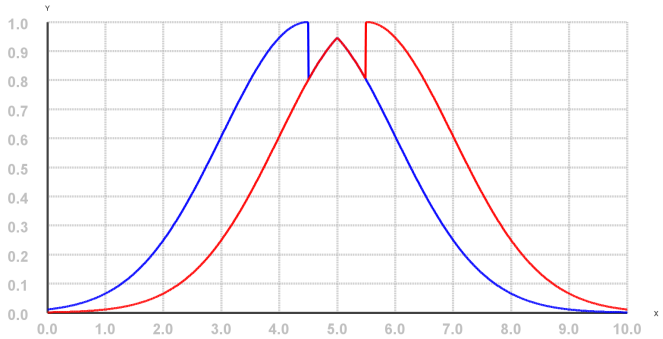


Figure 3.7: *ESs used by the KM procedure to obtain the centroid of the IT2 FS in Fig. 3.6*

points. In addition to that, knowing which rules (and therefore which inputs and antecedents) generated the ES from which the endpoints are obtained, gives an explanation to how and why the final output of the system has been obtained. In the KM case, on the other hand, the shape coherency with the original shape is partly lost and the firing strengths are not as clear as in the CIT2 case.

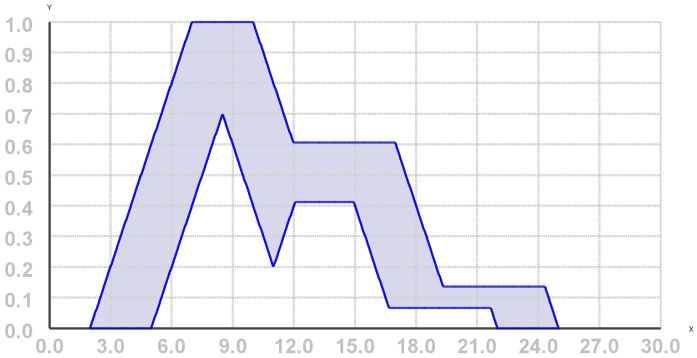


Figure 3.8: *Fuzzy output of a CIT2 fuzzy system*

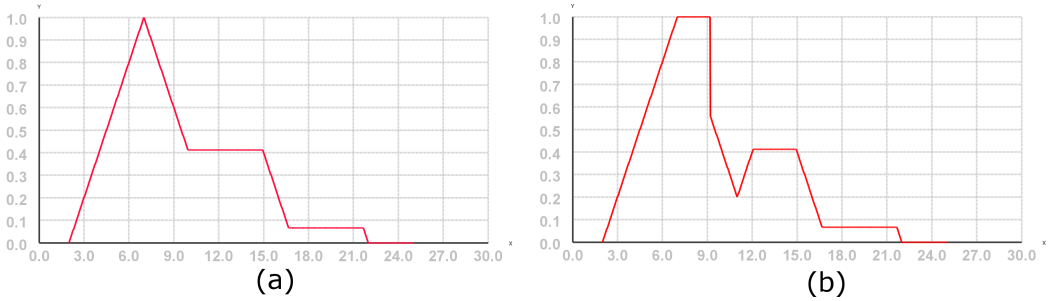


Figure 3.9: *ESs that determine the left value of the CIT2 (a) and KM (b) centroid of the set in Fig. 3.8.*

Intuitively, the standard T2 definition gives too much "mathematical free-

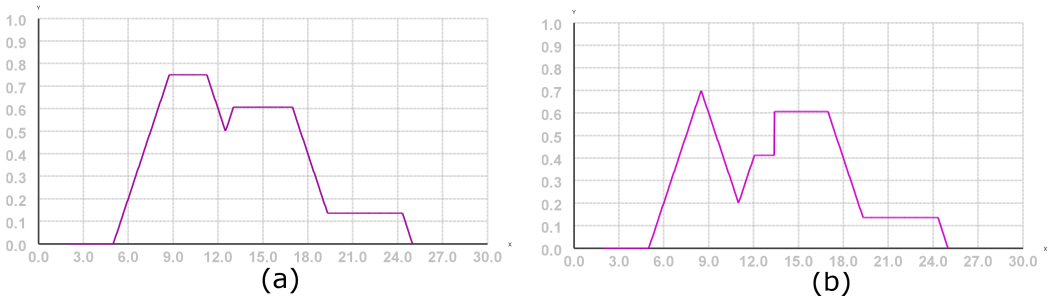


Figure 3.10: *ESs that determine the right value of the CIT2 (a) and KM (b) centroid of the set in Fig. 3.8.*

dom" *in some contexts*, posing no restrictions on the shape of the FOU and of the ESs, especially when modeling T2 MFs from an underlying concept represented as a T1 FS with uncertainty. For these reasons, CIT2 FSs were proposed, in which both the FOU and the ESs considered as acceptable have a shape that is "meaningful" for the context in which they are used.

The specific sense of "meaningfulness" can vary. The intuitive idea is that the shape of the MFs should be reasonable for the semantic meaning they carry. For example, in the case of the concept of medium height, only a MF that monotonically increases up to a plateau and then monotonically decreases would be "meaningful". That is simply because any MF without these properties would result in a counter-intuitive set for the representation of the medium height concept.

In other contexts, meaningful shapes can be obtained as a result of experimental observations, data analysis or experts' knowledge.

3.3 Constrained Interval Type-2 Fuzzy Sets

Although the main concepts of CT2 FSs can be extended to all T2 FSs, the rest of the thesis will only focus on *interval* type-2 fuzzy sets and their constrained representation (CIT2). The motivations behind this decision will be discussed later in the chapter. Also, it is assumed that the universe of discourse (UOD) considered is a connected subset of $\mathbb{R}$.

The idea behind CIT2 FSs is to generate a T2 FS starting from a T1 FS

modeling the same semantic concept. This T1 FS is called type-1 generator set (T1 GS) (e.g. the T1 FS in Fig. 3.2 is the T1 GS for the thought experiment in Sec. 3.2). To obtain the CIT2 FS, uncertainty is added on the location of the T1 GS on the x-axis. That is done by using a set of offsets, that intuitively represent all the possible valid locations of the T1 GS. This set of offsets is called *displacement set*:

Definition 3.1. *A displacement set (DS), denoted D, is a closed set of real numbers such that:*

$$D \subseteq \mathbb{R}, 0 \in D \tag{3.1}$$

When the DS is a continuous interval, it can be expressed as D=[a,b], where $a, b \in \mathbb{R}, a \leq 0 \leq b$.

With a DS plus a T1 GS, it is possible to define the T1 FSs that will represent the acceptable embedded sets (AES) of the CIT2 FS modelled.

Definition 3.2. *A collection of T1 acceptable embedded sets (CAES), is a set of T1 FSs obtained from the shifting of a T1 GS G. Formally, each of the acceptable embedded sets (AES) S in a CAES can be expressed as:*

$$S = \{(x, \mu_S(x)) \,|\, x \in X\} \tag{3.2}$$

where

$$\mu_S : X \mapsto [0, 1], \exists c \in D : \mu_S(x) = \mu_G(x - c), \forall x \in X \tag{3.3}$$

given a UOD X, a DS D, a T1 GS G.

Given a CAES, it is possible to generate a CIT2 FS:

Definition 3.3. *A constrained interval type-2 fuzzy set (CIT2 FS)* $\breve{A}$*, is defined as follows:*

$$\breve{A} = \{((x, u),\, 1) \,|\, x \in X,\, u \in \bigcup_{S \in CAES_{\breve{A}}} \mu_S(x)\} \tag{3.4}$$

with $CAES_{\breve{A}}$ *being the CAES from which* $\breve{A}$ *is obtained. In this case,* J_x *can be rewritten as follows:*

$$J_x = \bigcup_{S \in CAES_{\breve{A}}} (x, \mu_S(x)), \quad \mu_S(x) > 0 \tag{3.5}$$

$\breve{A}$ *can also be written as:*

$$\begin{aligned} \breve{A} &= \int_{x \in X} \int_{u:(x,u) \in J_x} 1 \Big/ (x, u) \\ &= \int_{x \in X} \int_{u \in \bigcup\limits_{S \in CAES_{\breve{A}}} \mu_S(x)} 1 \Big/ (x, u) \end{aligned} \tag{3.6}$$

It is important to note that CIT2 FSs represent a subset of IT2 FSs since they impose additional constraints on their mathematical definition, just like IT2 FSs represent a subset of the more general T2 FSs.

In order to prove an important property, it is necessary to build a three-dimensional version of the sets in the CAES. Since they are T1 FSs, building their three-dimensional representation is straightforward. Given a T1 set A, its three-dimensional representation $\tilde{A}$ (i.e. its representation as a T2 FS) is defined as follows:

$$\tilde{A} = \{(x, \mu_A(x), 1) \,|\, x \in X\} \tag{3.7}$$

By applying (3.7) to all the sets in a given CAES, a collection of IT2 acceptable embedded sets is obtained.

Definition 3.4. *A collection of acceptable IT2 embedded sets* $(\widetilde{CAES})$ *of a CIT2 set* $\breve{A}$*, denoted* $\widetilde{CAES}_{\breve{A}}$*, is a set of CIT2 embedded sets described as follows:*

$$\widetilde{CAES}_{\breve{A}} = \{\tilde{S} \,|\, S \in CAES_{\breve{A}}\} \tag{3.8}$$

with

$$\tilde{S} = \{((x, \mu_S(x)), 1 \,|\, x \in X\} \tag{3.9}$$

Each of the sets $\tilde{S}$, can also be described as:

$$\tilde{S} = \int_{x \in X} \int_{\mu_S(x)} 1 \Big/ x = \int_{x \in X} (\mu_S(x), 1) \,/x \tag{3.10}$$

The sets in the $\widetilde{\text{CAES}}_{\breve{A}}$ are actual T2 ESs of $\breve{A}$, since they satisfy Definition 2.11.

While all the definitions up to this point could be easily extended to the general CT2 case, the conversion of T1 MFs to AESs of a general T2 FS would not be so trivial. That is because the membership degree of each of the pairs $((x, \mu_S(x))$ could not be easily determined since it could be any value between 0 and 1. The conversion to AES of IT2 FS, instead, is straightforward and shown in Def. 3.4. A possible solution to this has been proposed in [1], in which a similarity function is used on each AES S and the GS to determine $\mu_{\tilde{S}}(x, \mu_S(x))$, $\forall x$. However, the use of this and other possible approaches, together with the interpretability of three-dimensional embedded sets will be analyzed in future work. Definition 3.4 is very important since it allows us to introduce the CIT2 representation theorem:

Theorem 3.1. *Given a CIT2 set $\breve{A}$ and its $\widetilde{CAES}_{\breve{A}}$, $\breve{A}$ can be expressed as the crisp set union of all the IT2 sets $\tilde{S}$ in $\widetilde{CAES}_{\breve{A}}$:*

Proof. To do that, it will be shown that it is possible to write the union of all the $\tilde{S} \in \widetilde{\text{CAES}}_{\breve{A}}$ as (3.6), by rewriting $\widetilde{S}$ as in (3.10):

$$\begin{aligned} \int_{\tilde{S} \in \widetilde{\text{CAES}}_{\breve{A}}} \tilde{S} &= \int_{\tilde{S} \in \widetilde{\text{CAES}}_{\breve{A}}} \left(\int_{x \in X} \int_{u = \mu_S(x)} 1 \Big/ (x, u) \right) \\ &= \int_{x \in X} \int_{u \in \bigcup\limits_{S \in \text{CAES}_{\breve{A}}} \mu_S(x)} 1 \Big/ (x, u) \end{aligned} \tag{3.11}$$

□

Theorem 3.1 allows us to define CIT2 operations by only working with AESs. For example, the union of two sets $\breve{A}$ and $\breve{B}$ is defined as follows.

Corollary 1. *Given two CIT2 sets $\breve{A}$ and $\breve{B}$, their union is the union of the T2 embedded sets $\tilde{S}$ in $\widetilde{CAES}_{\breve{A}}$ and $\widetilde{CAES}_{\breve{B}}$*[1]*:*

$$\begin{aligned} \breve{A} \cup \breve{B} &= \int_{\tilde{A}' \in \widetilde{CAES}_{\breve{A}}} \tilde{A}' \cup \int_{\tilde{B}' \in \widetilde{CAES}_{\breve{B}}} \tilde{B}' \\ \breve{A} \cup \breve{B} &= \int_{\tilde{A}' \in \widetilde{CAES}_{\breve{A}}} \int_{\tilde{B}' \in \widetilde{CAES}_{\breve{B}}} \tilde{A}' \cup \tilde{B}' \end{aligned} \tag{3.12}$$

Intuitively, all the combinations of all the AES of the two CIT2 sets involved in the operation are considered. The unions between the AESs of $\breve{A}$ and $\breve{B}$ generate the AESs of the FS generated from the union of $\breve{A}$ and $\breve{B}$.

Analogously,the CIT2 intersection and complement can be derived:

$$\breve{A} \cap \breve{B} = \int_{\tilde{A}' \in \widetilde{\mathrm{CAES}}_{\breve{A}}} \int_{\tilde{B}' \in \widetilde{\mathrm{CAES}}_{\breve{B}}} \tilde{A}' \cap \tilde{B}' \tag{3.13}$$

$$\overline{\breve{A}} = \int_{\tilde{A}' \in \widetilde{\mathrm{CAES}}_{\breve{A}}} \overline{\tilde{A}'} \tag{3.14}$$

Also the upper and lower MFs of the FOU of a CT2 FS can be expressed in terms of the AES:

Definition 3.5. *Given a CIT2 FS $\breve{A}$, its upper MF $\overline{\mu}_{\breve{A}}$ and lower MF $\underline{\mu}_{\breve{A}}$ are defined as follows:*

$$\overline{\mu}_{\breve{A}}(x) = \sup_{S \in CAES_{\breve{A}}} \mu_S(x) \tag{3.15}$$

$$\underline{\mu}_{\breve{A}}(x) = \inf_{S \in CAES_{\breve{A}}} \mu_S(x) \tag{3.16}$$

Even though IT2 and CIT2 operations may seem similar, they are conceptually different. In the IT2 case, the only goal of operations such as the union and intersection is to generate the new upper and lower-bound MFs and therefore the FOU. In the CIT2 case that is not enough. In fact, the key point of CIT2 operators is the generation of a new CAES, that determines which ESs are considered acceptable and therefore which ESs will be considered by

[1] (3.12) involves integral and union signs, where the integral sign is shorthand for lots of union signs. The union sign indicates the union between members of a set, whereas the integral sign represents the union of the sets themselves.

other CIT2 fuzzy operators (such as the centroid). This property is necessary to maintain the concept of interpretability (as semantic relation) described so far in the chapter.

Since every CIT2 set can be expressed as the union of the AES in its $\widetilde{\text{CAES}}$, this property can be used to define the constrained centroid, denoted as $Cen(\breve{A})$:

$$Cen(\breve{A}) = \int_{\tilde{A}' \in \widetilde{\text{CAES}}_{\breve{A}}} Cen(\tilde{A}') \tag{3.17}$$

That is, the union of all the centroids of the sets in $\widetilde{\text{CAES}}_{\breve{A}}$. The constrained centroid is analogous to the IT2 one, in which the centroid is the union of the centroids of all its embedded sets [17]. The difference is that in the CIT2 case only the collection of AESs is taken into account. They represent a subset of all the ESs examined in the standard IT2 approach. In addition, since the CAES is a subset of all the ESs embedded in a given FOU, the constrained centroid will always be contained (or will be equal to) the standard IT2 centroid.

When a CIT2 FS is not the result of a CIT2 fuzzy operator but is generated from a T1 GS with a continuous DS, the CIT2 centroid has an interesting mathematical property. In fact, in that case, the centroid can be rewritten as the following interval:

$$Cen(\breve{A}) = [Cen(\tilde{A}_L), Cen(\tilde{A}_R)], \quad \tilde{A}_L, \tilde{A}_R \in \widetilde{\text{CAES}}_{\breve{A}} \tag{3.18}$$

with $\tilde{A}_L, \tilde{A}_R$ being the left-most and right-most AES of $\breve{A}$. The proof for that equation is straightforward: since all the AES of a CIT2 generated from a GS share the same shape, the AES obtained from the leftmost shift will trivially have the lowest centroid value and will therefore determine the left endpoint of the centroid; analogously, the right endpoint is generated by the rightmost AES.

However, (3.18) may not hold anymore after the application of a set theory operation. Intuitively, that is because (3.18) can be used when all the sets in

$\widetilde{CAES}$ have the same shape. An example of a case in which (3.18) can not be used is given by the CIT2 FS in Fig. 3.11. Its AES (e.g. Fig. 3.9 (a), 3.10 (a)) are obtained as the aggregation of three triangular MFs "truncated" (i.e. inferred) at different heights. In that case, determining which "truncation values" generate the AES with the lowest and highest centroid value is non-trivial, as will be also discussed in Sec. 3.4.1.

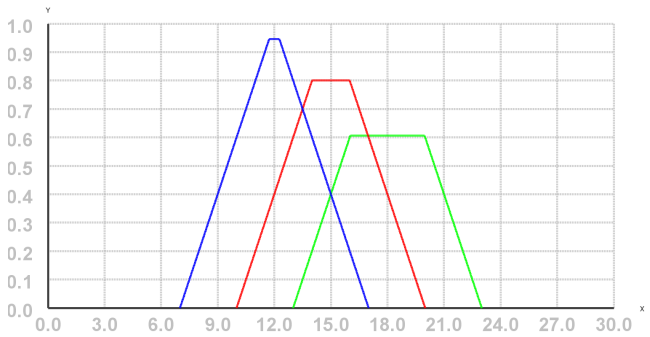


Figure 3.11: *Some AES of the CIT2 output from the inference of a CIT2 rule in which all the sets involved are CIT2 sets*

Lastly, the FOU (see (2.10)) of a CIT2 FS $\breve{A}$ can be rewritten using only the AESs:

Definition 3.6. *The FOU of a CIT2 FS $\breve{A}$ can be defined as:*

$$FOU(\breve{A}) = \{(x,u) \,|\, ((x,u),1) \in \int_{\tilde{S} \in \widetilde{CAES}_{\breve{A}}} \tilde{S}\} \tag{3.19}$$

3.4 Inferencing with CIT2 sets

Now that a formal definition of CIT2 FSs and all their components has been presented, they can be used to build fuzzy rules and fuzzy systems. For CIT2 fuzzy systems to be usable, however, the procedure to carry out the Mamdani inference with singleton fuzzification needs to be defined.

Consider the following constrained interval-type-2 fuzzy rule (CIT2 fuzzy rule), i.e. a fuzzy rule in which all the sets involved are CIT2 FSs:

$$\text{IF } x_1 \text{ IS } \breve{A} \text{ AND } x_2 \text{ IS } \breve{B} \text{ THEN } y \text{ IS } \breve{C} \tag{3.20}$$

Using Theorem 3.1, the latter can be rewritten as:

$$\begin{aligned}&\text{IF } x_1 \text{ IS } \int_{\tilde{A}' \in \widetilde{\text{CAES}}_{\breve{A}}} \tilde{A}' \text{ AND } x_2 \text{ IS } \int_{\tilde{B}' \in \widetilde{\text{CAES}}_{\breve{B}}} \tilde{B}' \text{ THEN} \\ &y \text{ IS } \int_{\tilde{C}' \in \widetilde{\text{CAES}}_{\breve{C}}} \tilde{C}'\end{aligned} \tag{3.21}$$

Since all the sets in the $\widetilde{\text{CAES}}$ are a three-dimensional representation of T1 sets (see (3.7)), T1 mathematics can be used to operate with them.

After the singleton fuzzification of the input, the antecedent operation is straightforward. For example, for the fuzzified input x_1 in the rule mentioned above, it is:

$$\int_{A' \in \text{CAES}_{\breve{A}}} \mu_{A'}(x_1') \tag{3.22}$$

where x_1' is a specific value of x_1.

The antecedent composition is therefore given by the following formula:

$$\begin{aligned}&\int_{A' \in \text{CAES}_{\breve{A}}} \mu_{A'}(x_1') \quad \star \int_{B' \in \text{CAES}_{\breve{B}}} \mu_{B'}(x_2') = \\ &\int_{A' \in \text{CAES}_{\breve{A}}} \int_{B' \in \text{CAES}_{\breve{B}}} \mu_{A'}(x_1') \star \mu_{B'}(x_2')\end{aligned} \tag{3.23}$$

with $\star$ being a T-norm. The antecedent composition as described so far, returns a set of real numbers. Each of these values can be then used to apply the implication method (i.e. any T-norm) to each of the AES $C' \in \text{CAES}_{\breve{C}}$, producing the $\text{CAES}_{\breve{C}^*}$ of the fuzzy CIT2 output $\breve{C}^*$. In the rest of the chapter, it is assumed that the minimum operator is used for the implication method and informally refer to this operation as *truncation*. To defuzzify $\breve{C}^*$, a procedure that is based on the result shown in (3.17) is implemented. The CIT2 centroid is a pair (l, u), where:

$$l = \inf\left(Cen(\breve{C}^*)\right) \tag{3.24}$$

$$u = \sup\left(Cen(\breve{C}^*)\right) \tag{3.25}$$

remembering from (3.17) that:

$$Cen(\breve{C}^*) = \int_{\tilde{C}' \in \widetilde{\mathrm{CAES}}_{\breve{C}^*}} Cen(\tilde{C}') \tag{3.26}$$

Since each of the IT2 sets in the $\widetilde{\mathrm{CAES}}_{\breve{C}^*}$ is just a three-dimensional representation of a T1 set, the equivalent T1 sets in $\mathrm{CAES}_{\breve{C}^*}$ can be defuzzified instead, by using the standard T1 centroid defuzzification method. Therefore, the pair (l, u) provides us a lower (l) and an upper (u) bound for the set of centroids in (3.17). This approach is conceptually similar to the Karnik-Mendel (KM) [18] procedure, in the sense that both return a pair composed of the upper and the lower bound of a set of centroids (that in the case of the KM approach, is the set of the centroids of all the ES of the IT2 FS).

The whole inference process where the CIT2 FSs involved have a finite number of AES, is described in pseudo-code in Algorithm 2.

Algorithm 2 Inference and Type-Reduction Algorithm

procedure CIT2 MAMDANI INFERENCE AND TYPE-REDUCTION (CIT2-FUZZY SYSTEM S, INPUT $[x_1, ..., x_n]$)
  for each rule $R_i \in$ S **do**
    for each permutation P of the AES of the CIT2 antecedents in R_i **do**
      firing_strengths.add(P.evaluateAntecedents()); ▷ the FSs in P are T1 AES
    end for
    for each consequent $C \in R_i$ **do**
      for each AES $E \in C$ **do**
        for each $c \in$ firing_strengths **do**
          CIT2_result_R_i.add_AES(implicate(E, c)); ▷ add a new AES to the rule i output
        end for
      end for
    end for
  end for
  CIT2_output=$\{\emptyset\}$ ▷ CIT2 FS representing the output of the system
  for each rule $R_i \in$ fuzzy system S **do**
    CIT2_output=CIT2_union(CIT2_output, CIT2_result_R_i); ▷ union of rule outputs
  end for
  left_value=$\inf_{E \in \text{CIT2\_rb\_output}}$(centroid($E$)); ▷ lowest AES centroid value
  right_value=$\sup_{E \in \text{CIT2\_rb\_output}}$(centroid($E$)); ▷ highest AES centroid value
  return (left_value, right_value);
end procedure

3.4.1 Result of CIT2 operators

It is interesting to see how the result of CIT2 operators on CIT2 FSs may result in a FS in which it may not be possible to identify a T1 GS G in the CAES from which the remaining AESs can be obtained by shifting G. That is because there is no guarantee that all the sets obtained as the result of the implication operator, for example, will have the same shape.

However, the shape of the T1 GS is not totally lost after the application of CIT2 fuzzy operators. Fig. 3.11 shows some of the AES of the inference output of a CIT2 fuzzy rule of the form IF x_1 IS $\breve{A}$ THEN y IS $\breve{C}$ where all the CIT2 FSs involved have a discrete DS (i.e. a finite number of AES). It is possible to see that even though the sets forming the CAES of the output do not share exactly the same shape, they all come from the same generator set (i.e. a triangular T1 FS) truncated at different heights during the inference process (the consequent CIT2 FS $\breve{C}$ before the inference can be found in Fig. 3.12).

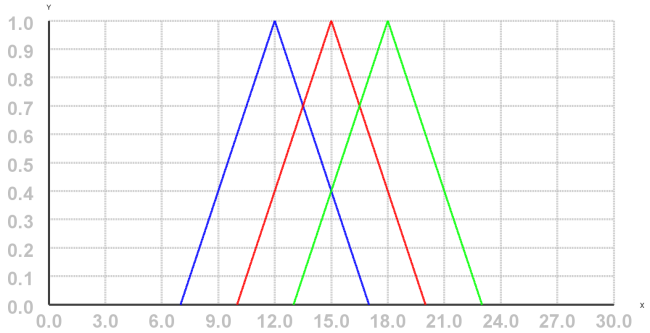


Figure 3.12: *Consequent CIT2 in the rule generating the output set shown in Fig. 3.11*

Intuitively, these AESs are meaningful even if they have different shapes because they represent actual T1 inference results that are obtainable from T1 inference by picking one of the AES from each of the antecedent and consequent CIT2 FSs in the fuzzy rule. The fact that each of the AESs is obtained from a shifted GS truncated at a given height is extremely important to build interpretable and explainable CIT2 systems. In fact, when one of those acceptable embedded sets is selected, its interpretability is guaranteed by the

semantic connection with the concept it is modeling, since it has the same shape as the GS, while its truncation height is directly related to the firing strength of the rule(s) that generated it. Therefore, it is possible to give an explanation for how this AES has been generated by showing the rules (and therefore, the inputs) that contributed to its creation.

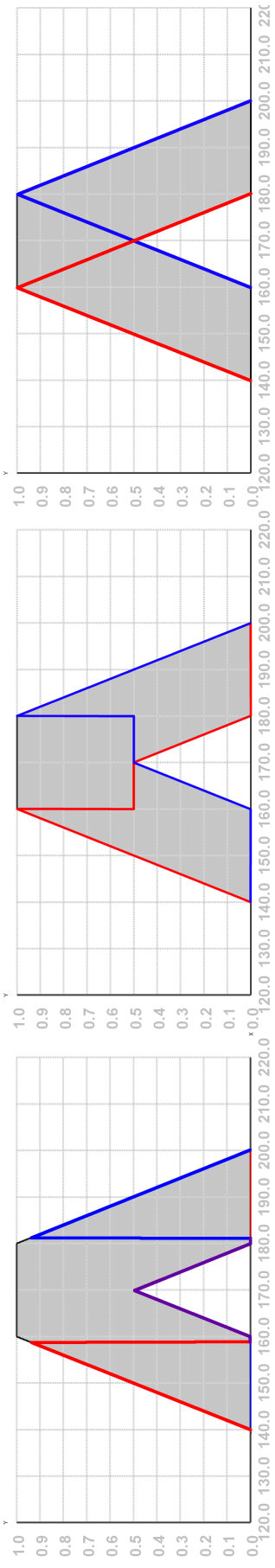


Figure 3.13: *ESs that determine the end-point values of the KM (a), W-CIT2 (b) and CIT2 centroid (c). In (a), the area where the 2 ESs overlap is coloured in purple.*

3.4.2 On the interpretability of CIT2 sets and systems

As shown In Sec. 3.3, the CIT2 FOU is a set of points, exactly like the FOU of a standard IT2 FS. If one considers the shape of a CIT2 FS alone, it is clear that its interpretability depends only on the shape of its FOU (and its boundaries) and not on the specific set of ES that are embedded into it. However, some T2 uncertainty measures do make use of these embedded sets and it is in these cases that CIT2 are able to provide a clear advantage over IT2 FS, allowing for the creation of explainable CIT2 FS and systems. Specifically, each of the AES that can be selected by the above mentioned fuzzy operators in the CIT2 case, has been created so that it is able to carry meaningful information. This is done both by keeping a semantic relation with the concept it is modeling (i.e. by keeping the same shape as the generator set) and by conveying, in the case of rule-base systems, information on the rule that generated it and its firing strength. In other words, it is possible to build CIT2 fuzzy systems that not only are able to solve, for example, classification problems, but that are also able to explain, in terms of the input space, how each endpoint of the interval centroid has been obtained. With a standard IT2 system this property is lost simply because in the defuzzification process, the ESs that produce the endpoints do not carry any meaningful information on which rules played a role in their generation and why. Therefore, in IT2 systems an explanation in terms of the input space can not be provided for the centroid but only for the boundaries of the FOU of the fuzzy output of the system. The ability of CIT2 fuzzy systems to explain also the endpoints of the centroid, on the other hand, clearly represents a novelty and a progress for T2 FSs in the increasingly popular XAI field.

3.4.3 Efficiency

The main goal of Algorithm 2, is to provide a procedure to compute the inferencing and defuzzification processes described in this section. For now, the

optimization of computational complexity has not been the main focus. It is clear that the proposed algorithm is slower than the current IT2 inferencing and defuzzification methods. That is because after the evaluation of the whole rule-base, the output is a set of AESs (line 16, in Algorithm 2) that can be quite big in size: each rule can produce (line 9) a number of implication sets that, in the worst case, is equal to the size of the permutations of the AESs of the antecedents, multiplied by the cardinality of the DS of the consequent. In addition, on line 16 the unions of all the possible permutations of the AESs of the CIT2 resulting from the single rules is generated. This union, generates a number of AES that grows as a double exponential, being $O(k^{n+1})^m$ where m is the number of rules, n the number of antecedents per rule and k the number of AES of each of the CIT2 involved.

Since this approach enumerates all the AESs to find the final defuzzified output, it is the analogous of the exhaustive defuzzification method rather than the KM one. In fact, the strength of the KM procedure is that it quickly identifies the ESs to be used for the left and right centroid values. On the contrary, in Algorithm 2 the AESs that give the left and right centroid value are found using a brute force approach, first building all the AESs of the total rule-base evaluation (line 16) and then finding among them the two that will give us the left and right centroid values (lines 18 and 19). For use in real-world problems, this approach is impractical because of its prohibitive computational complexity. For this reason, the alternative, much faster and practical defuzzification Algorithm 2 is proposed in Sec. 3.6. This algorithm is then used within the genetic framework described in Sec. 3.7, in which it is applied on two well known real-world datasets and compared to the KM procedure.

3.5 Comparison with a different constrained approach

In this section, the constrained representation presented in this chapter will be compared to a different approach (that here will be called W-CIT2) proposed by Wu et al. in [21, 52]. They start from the observation that ESs have been used to obtain theoretical results such as the definition of uncertainty measures and are processed regardless of their shape.

However, the authors point out that in many fuzzy logic applications the MFs that are used are convex and normal. Consequently, they propose a constrained representation theorem that allows the definition of the FOU of well-shaped (see [52] for details) IT2 FSs by using only convex and normal ESs. They claim that this definition is more general than the one that only considers ES with the same shape and doesn't require any expert knowledge or data analysis to determine which shapes are meaningful in a given context. Using this new theorem, many constrained uncertainty measures (such as centroid, entropy and cardinality) are defined mathematically. In addition to that, the authors show how the convexity and normality constraints can be simply added to the KM algorithm to find the constrained centroid value of a well-shaped IT2 FS. Finally, the authors also state that this approach can't be used in Mamdani systems since their outputs can be non-well-shaped.

The main difference between the representation theorem proposed in this chapter and the W-CIT2 one is in the definition of the ESs that are considered acceptable. Even though it is true that the W-CIT2 theorem allows the presence of multiple shapes among the ESs, normality and convexity can be not sufficient and not necessary to obtain shapes that are meaningful. Those two properties alone, still do not guarantee there will be a meaningful connection between an ES and the concept it models. To support this claim, a comparison is provided between the ESs that determine the end-points of the W-CIT2, CIT2 and IT2 centroid with the KM procedure (Fig. 3.13). The set

to defuzzify has been obtained starting from a triangular T1 MF as a generator set, using the approach described in this chapter to build the FOU around it. The comparison shows how the ESs used by the KM approach (Fig. 3.13 (a)) are both non-normal and non-convex. In addition to that, they could hardly represent any word or label. As a result, the meaningfulness and interpretability of the centroid value returned as an output decreases. On the other hand though, the KM algorithm can be applied to any IT2 FS, regardless of the approach used to obtain its FOU. The ESs used by the W-CIT2 approach, instead, are both normal and convex. However, also in this case the relation between the original T1 triangular shape (i.e. the one that has been used as a generator set) and the ESs is lost. Again, these sets would hardly model the same concept (e.g. medium height) from which the generator set is obtained. The ESs used by the CIT2 approach, instead, keep the same level of the interpretability as the generator set as they share its shape. The only difference between them is their location on the x-axis. From this experiment, it can be concluded that normality and convexity alone may not be sufficient to guarantee the meaningfulness of a FS. In addition to that, the fact that W-CIT2 FSs are not usable in Mamdani systems represents a significant limitation that can be overcome by the CIT2 definition provided in this chapter, as shown in Sec. 3.4.

Algorithm 3 CIT2 Sampling Algorithm

```
1: procedure CIT2 Sampling Algorithm (CIT2-fuzzy system S, int total_samples, Input [x_1, ..., x_n])
2:    Set centroids=new Set();                          ▷ This set will contain the centroids of the sampled AES
3:    HashMap<CIT2, T1MF> cit2_to_aes;                  ▷ Mapping each CIT2 into one of its AES
4:    for int index=0;index<total_samples; index++ do   ▷ Each iteration generates 1 sample
5:       T1_Rulebase t1_rulebase=new T1_Rulebase();
6:       cit2_to_aes=new HashMap<>();                   ▷ For each sample, new AESs are chosen
7:       for each CIT2_Rule R_i ∈ S do
8:          T1_Antecedents antecedents=new T1_Antecedents();
9:          T1_Consequents consequents=new T1_Consequents();
10:         for each CIT2_Antecedent curr_ant ∈ R_i do
11:            T1MF current_random_aes;
12:            ▷ If a random AES for this set has never been generated before in this iteration...
13:            if cit2_to_aes.get(curr_ant)==null then
14:               ▷ ...generate it and add it to the Map
15:               cit2_to_aes.put(curr_ant, curr_ant.randomAES();
16:               ▷ By doing this, if a set appears multiple times as an antecedent in different rules, the same AES
17:               will be used to replace it in the current iteration
18:            end if
19:            antecedents.add(cit2_to_aes.get(curr_ant);   ▷ Add the AES to the rule
20:         end for
21:         for each CIT2_Consequent curr_cons ∈ R_i do
22:            T1MF current_random_aes;
```

```
23:                ▷ If a random AES for this set has never been generated before in this iteration...
24:                if cit2_to_aes.get(curr_cons)==null then
25:                    ▷ ...generate it and add it to the Map
26:                    cit2_to_aes.put(curr_cons, curr_cons.randomAES();
27:                end if
28:                consequents.add(curr_cons.randomAES());
29:            end for
30:            t1_rulebase.addRule(new T1_Rule(T1_Antecedents, T1_Consequents));
31:        end for
32:        for each Input input_i ∈ t1_rulebase do
33:            input_i.setValue(x_i);
34:        end for
35:        ▷The rulebase produces a sampled AES the centroid of which must be computed
36:        centroids.add(t1_rulebase.run().getCentroid());
37:    end for
38:    return (centroids.minValue(), centroids.maxValue());
39: end procedure
```

3.6 Sampling approach for the CIT2 centroid

As already discussed in Sec. 3.4, the evaluation of the CIT2 centroid as described in Algorithm 2 is prohibitive due to the astronomical number of AESs that are examined to determine the defuzzified value. Therefore, even though the algorithm proposed earlier in this chapter is theoretically correct for the computation of the CIT2 centroid, it is not usable in practice for real world problems. Conceptually, the problem is very similar to the one that is faced when exhaustive defuzzification is applied to T2 FSs. In that context, many approximation algorithms have been proposed to overcome the computational complexity of the exhaustive defuzzification. One of them is the sampling method [57]. The intuitive idea is that each of the ESs in a T2 FS only has a minimal contribution to the final result, therefore generating a random sample of the ESs is a good and efficient way to obtain an approximation of the actual centroid value, as shown in [45]. In this case, the same concept is applied to sample a fixed number of AESs to determine the constrained centroid. A sample, is obtained by replacing each CIT2 FS in the rulebase with one of its AES chosen at random (rather than replacing each set with all its AES, as in Algorithm 2).

The fuzzy output of the T1 system obtained by carrying out all the substitutions will produce a single sampled AES. As a consequence of that, only a subset of all the AES is generated, making this approach an approximation algorithm. Once the number of desired samples has been obtained, the AESs are defuzzified and the lowest and highest centroid values among them will determine respectively the left and right end-point of the constrained centroid.

Conceptually, the following steps are used to produce a single sampled AES of the CIT2 fuzzy output:

- For each set $\breve{A}$ involved in the FLS:
  - Generate a random number k within its DS
  - Use k to shift the GS of $\breve{A}$ along the x-axis, obtaining E, an AES

of $\breve{A}$; remembering that given a function $f(x)$ its translated version by a factor k along the x-axis can be written as $f'(x) = f(x - k)$, this step can be done in constant time without the need to store all the AES to choose one randomly

- Loop through all the rules and replace $\breve{A}$ with E

- Once all the CIT2 FS have been replaced with a random AES, a T1 rulebase is generated

- The fuzzy inferenced result of the rule-base represents a sampled AES

The output interpretability offered by CIT2 FLSs is given by the process used to produce the AES. In fact, each of them represents a T1 fuzzy output and as such keeps all the interpretability properties that belong to the outputs of T1 FLSs: the shapes of the consequent set involved in the rules are clearly identifiable together with the firing strengths used for the inference operator (e.g. see Fig. 3.9(a), 3.10(a), 3.13(a)). These properties, also make possible a direct connection between the endpoints of the interval centroid and the rules that were used in its generation.

The pseudo-code (mainly written following OOP conventions) of the sampling method is described in Algorithm 3.

Other than the reduction in the computational cost, the other main advantage of this approach is its applicability to systems in which the CIT2 FSs involved have a continuous DS, i.e. the number of AESs per CIT2 FS is infinite. In fact, Algorithm 2 only works with a discrete number of AESs and may therefore require an additional discretization step. With the sampling approach each CIT2 FS involved in the rule can be easily substituted with one of its AESs by shifting its generator set by a random value in the DS during the conversion step (mentioned above) of the CIT2 rule into a T1 one.

3.7 CIT2 Fuzzy Systems in Practice

In this section, a framework for the automatic learning of CIT2 fuzzy systems will be described and applied to two real-world classification problems. The aim is not to compare this learning method to other approaches proposed in literature in terms of performance, but rather to present a possible way of generating CIT2 fuzzy systems and show a practical application of these new fuzzy sets and their inference framework described so far.

Classification problems have been chosen because they represent one of the contexts in which interpretability and especially explainability play a crucial role. In many applications, in fact, knowing both the output (the interval centroid) and how it has been obtained (i.e. which rules and which inputs determined the ES that produced the endpoints) is of great value and it is the main reason for the emergence of the new XAI field.

3.7.1 Learning CIT2 fuzzy systems Through Genetic Algorithms

Genetic algorithms have been widely used for the automatic generation and optimization of fuzzy systems [58] since they allow for the creation of both the rule-base and the MFs without the need of any expert knowledge. Even though these systems are obtained through machine learning techniques, they can maintain the typical interpretability of fuzzy logic systems as long as they contain a reasonably small number of rules and it is possible to give a linguistic label to the MFs involved [59]. The genetic approach proposed for the generation of CIT2 fuzzy systems, is based on the architecture described in [2]. Each of the input variables of the system is partitioned in 3 triangular MFs. The center of each triangular generator set for the antecedent CIT2 FSs is determined using the well known fuzzy C-Means clustering algorithm (FCM) [60] on each input variable. The end-points of the triangles are the center of the previous and next clusters, if they exist, or the closest end-point

of the UOD increased by 10% of the UOD size, so that every point in the UOD belongs to at least one of the MFs with a membership value greater than 0. The continuous DS is an interval $[-c, c], c > 0$ with $2c = 5\%$ of the distance between the starting and end point of each triangular generator set. The output variable is partitioned with a number of CIT2 FS equal of the number of classes in the problem. Each of them is given an integer index from 0 to the number of classes involved. The index represents the peak of their triangular generator set while the start and end point of the triangles are obtained respectively subtracting and adding 1 to their peak points. The DS for all the CIT2 MF partitioning the output is an interval $[-c, c], c > 0$ with $2c = 10\%$ of the UOD. Once the MFs are determined, there is a first evolutionary stage to generate the rule-base of the system. During this process, the MFs are not changed. The number of rules is fixed (as shown in [2], redundant rules can be eliminated with an additional stage) and each chromosome codes an entire rule-base. With n input variables, each rule is coded with a set of $n+1$ integers. Each gene p_i represents the index of the MF to use for the $i-th$ antecedent or for the consequent, if $i = n+1$. A value of -1 for $p_i, i \leq n$, indicates that the $i-th$ input must not be included in the rule p_i belongs to.

A sequence of encoded rules represents a rule-base.

Table 3.1: Parameters used for the learning architecture

Parameters	**Values**
Population size	100
Iteration limit	100 per stage
Crossover	Single-Point
Crossover rate	0.7
Elitism	5%
Fuzziness in FCM	2.0
Mutation rate	1/chromosome_size
Fitness function	Accuracy value
Memberships per variable	3
Fuzzy Rules per chromosome	Fixed, 10
Number of samples in CIT2 centroid	50
Random distribution for the random sampling	Uniform
Discretization points for AES defuzzification	100

At the end of the first stage, the fittest chromosome is returned. The rule-base encoded by this chromosome is passed to the second stage of the learning process, with the goal of optimizing the MFs involved in the system. Each triangular CIT2 MF is encoded with 4 real numbers: 3 modelling the generator set (starting point, center and ending point of the triangle) and one representing the size of the DS as a percentage of the UOD. Thanks to the way CIT2 MFs are built starting from a T1 generator set, the encoding of CIT2 MFs only requires 1 additional parameter with respect to their T1 counterpart. That is because the upper and lower MFs bounding the FOU of the set, are determined from the T1 generator shape and the DS. Standard IT2 representations, instead, may require up to twice the number of parameters of their T1 counterpart to fully represent the FOU and its bounds. The optimized rule-base obtained at the end of the second stage is then returned as the final

output of the learning process. The architecture is summarised in Fig. 3.14. For more information on the tuning and learning process, please refer to [2].

Table 3.2: Results of the genetic CIT2 fuzzy system with two different defuzzification approaches

	CIT2	IT2(KM)	CIT2 Time	IT2 Time
Iris	96.0%	94.667%	4h5m	1h11m
New-Thyroid	91.167%	91.667%	6h19m	1h31m

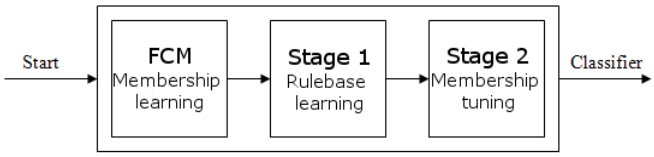


Figure 3.14: *The learning architecture used in this section. Adapted from [2]*

3.7.2 Application on real data-sets

The genetic architecture described above has been tested on two real world classification problems using two well known data-sets: iris [61] and new-thyroid [62]. The 10-fold cross validation method has been used to evaluate the CIT2 fuzzy systems; both data-sets, including the train and test partitions of each cross validation iteration, are publicly available on the KEEL website [63]. In both stages a single-point crossover has been used and the fitness function has been defined as the accuracy value of the rule-base encoded in the chromosome. A more detailed list of the parameters used in the optimization can be found in Table 3.1. The optimization has been carried out twice, once using the CIT2 sampling method with 50 samples to defuzzify the output and once using the implementation of the KM iterative procedure implemented in the Java library Juzzy[64]. The architecture has been implemented in Java using multi-thread computation on an i7-7600U CPU. The average results of both approaches and their running times for the 10 runs are reported in Table 3.2. It can be seen that the execution time of the CIT2 systems, featuring Algorithm 2, are higher than the IT2 systems. However, these execution times

represent approximately 10^7 individual defuzzification operations throughout the optimization process — i.e. each individual CIT2 defuzzification using Algorithm 2 takes around 1.5 millisecs using multi-threading to generate the samples. Whilst not as efficient as current IT2 defuzzification algorithms, this is clearly usable in real world applications, particularly decision problems.

As it is possible to see, the two approaches give similar results and perform well on both the data-sets analyzed. Therefore, to determine if and under what conditions one of the two defuzzification methods gives superior results more experiments are required, with a bigger number of data-sets and a statistical evaluation of their performances. To demonstrate the superior interpretability and explainability of the CIT2 approach, in Fig. 3.15 are shown the ES used to determine the right end-point of the constrained (a) and "standard IT2" (b) centroid generated by the KM procedure. Those ES have been obtained as the result of the defuzzification of the output of a CIT2 FLS generated through the learning framework described in this section. As discussed in Sec. 3.2, the AES selected by the CIT2 approach, provides a clearer understanding of the final system output, giving an intuitive idea of how the centroid value is obtained since, just like any T1 fuzzy output, it is still possible to identify the shapes of the consequent MFs and see the respective firing levels. Additionally, the firing strengths can be traced back to the rules and the inputs that generated the endpoints. The ability to produce explanations for each of the system outputs, *together* with the interpretable rule-based structure (characteristic of any FLS) make CIT2 FLS a valid alternative to IT2 for the development of FLS in the XAI field.

Currently running times seem to be the main drawback of this approach. In fact, in both the tests the IT2 approach with the KM procedure has proven to be roughly 3.5-4 times faster than the CIT2 one. In future works, a new and faster defuzzification methods to address this issue will be developed.

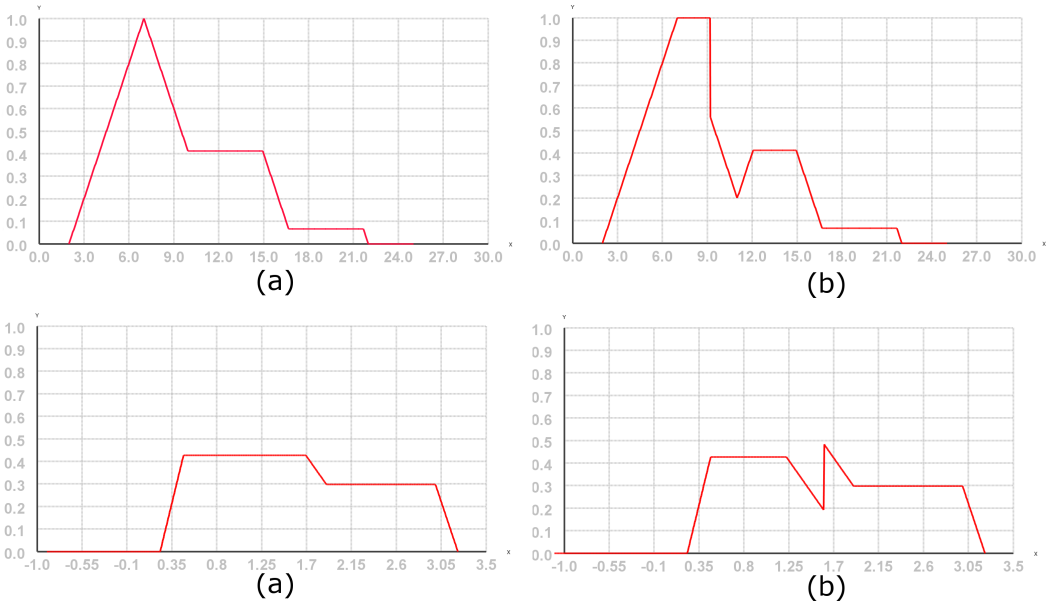


Figure 3.15: *ESs that determine the right end-point value of the CIT2 (a) and KM (b) centroid in a CIT2 system obtained through the genetic architecture described in this section.*

3.8 Summary

In this chapter, constrained interval type-2 (CIT2) have been fully formalized, showing how they can be obtained starting from a T1 FS with uncertainty on its exact location on the x-axis. The main idea behind CIT2 FSs is to produce a representation that considers only the ESs that have meaningful shape for a given concept; these embedded sets, called *acceptable* (AES), can then be used to define the FOU of the CIT2 FS and CIT2 fuzzy operators. The use of AESs rather than their unconstrained version, guarantees that CIT2 operators will only process embedded sets with a meaningful shape, increasing the interpretability of their output (as discussed in Sec. 3.2, 3.7.B).

Formal definitions of CIT2 FSs and AESs have been provided, together with the formulation of a new constrained representation theorem (Theorem 3.1). This allowed us to define all the main CIT2 operators, including the centroid defuzzification, by working only with "meaningful" ESs. Finally, a full inference framework has been presented for a CIT2 fuzzy system together with a defuzzification procedure. As a test case, a genetic architecture for the generation of CIT2 fuzzy systems has been described and applied to two real world

datasets. The preliminary results, presented here, show how the performances of the CIT2 approach are comparable to the ones obtained from the IT2 one, with the CIT2 system outputs presenting a higher level of interpretability. On the other hand, CIT2 FLSs have been shown to be slower, requiring approximately 4 times more time than their IT2 counterpart to complete the learning process. The next Chapter will tackle this issue, designing a faster approximation algorithm for to carry out the inference and type-reduction in CIT2 FLSs.

Chapter 4

A Faster Defuzzification approach

4.1 Introduction

Chapter 3 has shown how the higher level of interpretability of CIT2 FLSs comes at the cost of higher computational complexity in the type-reduction processes. In fact, while type-reducing a CIT2 set is trivial, the same operation becomes very computationally expensive for the inference. Specifically, the exhaustive procedure to type-reduce the output of a CIT2 Mamdani inference system has been shown to be impractical for real world applications due to its prohibitive computational cost. Even the approximation procedure (termed the CIT2 sampling method [65]), introduced for faster computation, has been shown to be significantly slower than the well-known Karnik-Mendel (KM) [18] algorithm for IT2 FSs.

The contribution of this chapter is a refined inferencing mechanism with an associated novel type-reduction approach which enables the much faster computation of CIT2 Mamdani inference systems. The novel procedure efficiently and deterministically selects a small number of appropriate embedded sets from which it produces the final type-reduced set. This reduction in the search space makes the approach presented in this chapter significantly faster

than the exhaustive and sampling CIT2 type-reduction algorithms [65] while maintaining comparable outputs and keeping the high level of interpretability that characterizes CIT2 fuzzy sets.

The rest of the chapter is organized as follows: after the novel inference and type-reduction technique is described and then formalized (Section 4.2), multiple experiments are carried out to compare this new algorithm with KM, its enhanced version (EKM, [19]) and the CIT2 sampling method to show the significant run time improvements (Section 4.3). Finally, the approach is applied to a real world classification problem, in which the explainability and accuracy of its classifications is discussed with respect to the KM approach (Section 4.3.3).

4.2 A Novel Constrained Centroid Defuzzification Method for Mamdani CIT2 fuzzy systems

Type-reducing a CIT2 fuzzy set is a trivial task: since all the acceptable embedded sets (AES) share the same shape, the left-most and right-most ones will produce the two end-points of the type-reduced set.

However, the same operation for the inference in Mamdani systems is non-trivial and computationally expensive. In fact, the AESs of the fired output of the system do not necessarily share the same shape anymore, as a consequence of the inference process. This phenomenon is shown in Fig. 4.1 where the AES (which before the implication had a triangular shape) have been 'truncated' at different heights as a result of the implication (min) operator. In this situation, determining the endpoint of the type-reduced set is no longer trivial.

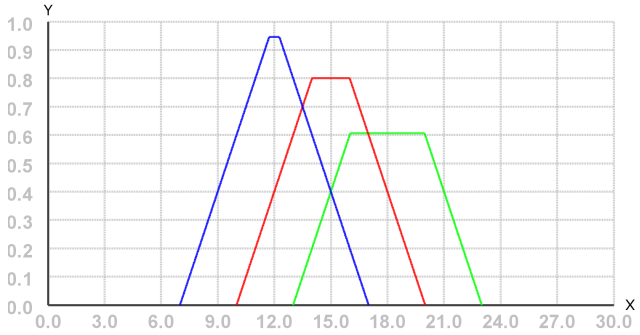


Figure 4.1: *Some of the AES generated from a CIT2 rule where the consequent has a triangular GS*

The exhaustive approach [65] and its approximation procedure named the sampling method [65] presented in the previous chapter, have been shown to be significantly slower than current type-reduction algorithms for IT2 Mamdani systems, making the use of CIT2 FLS in real world scenarios impractical.

In this chapter, a novel algorithm is proposed that selects a subset of AESs with specific criteria to compute the endpoints of the type-reduced set. In the experiments carried out and illustrated in Sec. 4.3, this new approach has been shown to be at least 7.5 times faster than both the exhaustive and sam-

pling algorithms while providing comparable results in terms of the endpoints produced. The lower computational times, make this novel approach usable in practical applications, fulfilling the third objective of this thesis.

4.2.1 Informal description

To type-reduce an IT2 FS, the KM algorithm finds the ESs with respectively the lowest and highest centroid. These two values, determine the endpoints of the interval that represents the type-reduced set. The MFs of these two ESs can be written using the lower and upper-bound MFs of the FOU they are embedded into:

$$\mu_L(x) = \begin{cases} \overline{\mu}_{\tilde{A}}(x), & x \leq S_L \\ \underline{\mu}_{\tilde{A}}(x), & x > S_L \end{cases}$$

$$\mu_R(x) = \begin{cases} \underline{\mu}_{\tilde{A}}(x), & x \leq S_R \\ \overline{\mu}_{\tilde{A}}(x), & x > S_R \end{cases}$$

where μ_L and μ_R are the MF of the ESs determining the endpoints, $\tilde{A}$ is the IT2 FS they belong to, S_L and S_R are two values in the universe of discourse (UOD), called respectively left and right *switch point*. Informally these two ESs 'coincide' with one of the two boundaries up to the switch point and then *switch* to the other boundary of the FOU. In the general case, if this approach is used to defuzzify a CIT2 FS, the ESs found by the KM algorithm would not be one of the AES (i.e. they would not have a *meaningful* shape).

As a result of the Mamdani CIT2 inferencing process (for more details, see [65]), the implication operator (min) is repeatedly applied to all the T1 AES of each consequent using all the values in the firing interval of the rule. An example of this operation is shown in Fig. 4.1, where the consequent before the implication had a triangular GS (i.e. the AES were triangles before they were "truncated").

The idea in this novel approach is to use a binary choice for the implication

operator for each of the AES. Rather than "truncating" them at all the possible values in the firing interval, each of the AES can be truncated either at the minimum or at the maximum value of the firing interval. For example, for a rule with a firing strength $[a, b]$, each AES of the consequent set can be truncated only at the values a or b rather than any of the values in $[a, b]$. As a consequence of that, the red set in Fig. 4.1, for example, would not be considered. In the determination of the left endpoint of the type-reduced set, only the leftmost AES of each MF is considered while only the rightmost AES is used when computing the right endpoint.

Each of the consequent MF is given an ordinal index based on its position. The aim is to choose an integer value i such that for all the consequent MF with an index value smaller than i, a specific endpoint of the firing interval is used during the inference (e.g the lower value); for all the MFs with an index value greater or equal to i, instead, the firing value used *switches*, so the other endpoint of the firing interval is used (e.g. if the lower value was used for the indices smaller than i, now the upper value is used). Hence, i is called *switch index*.

The goal of the new algorithm is to find the *switch indices* that produce the two sets with the maximum and minimum centroid value. Just like the switch points, the two switch indices can differ, respectively for the generation of the right and left endpoints of the type-reduced set.

4.2.2 Speeding up CIT2 Mamdani inference

As described above, for the exhaustive or sampling type-reduction methods to be used, each CIT2 rule has to produce not just a firing interval but rather a set of firing values that are then used to carry out the implication on the AESs of a consequent set of each rule. For example, in Fig. 4.1, three distinct firing values (i.e. the three different heights at which the sets have been 'truncated') are used. With the novel approach introduced in this chapter, however, only

the endpoints of the firing strength of each CIT2 rule are needed.

This subsection introduces a theorem that allows the firing strengths to be quickly determined in a way that is analogous to that used for IT2 rules. Specifically, to compute the endpoints of the firing strength of a CIT2 rule, it is sufficient to work with the boundary functions of all the CIT2 sets involved in the rule.

Theorem 4.1. *Given a CIT2 rule (i.e. a fuzzy rule in which all of the fuzzy sets involved are CIT2 FSs):*

$$IF\ x_1\ IS\ \breve{A}_1\ AND...\ AND\ x_i\ IS\ \breve{A}_i\ THEN\ y\ IS\ \breve{A}_{i+1} \tag{4.1}$$

the firing interval of the rule can be computed using only the upper-bound and lower-bound MFs $\overline{\mu}_{\breve{A}}$, $\underline{\mu}_{\breve{A}}$ *of the CIT2 FS* $\breve{A}_1, ..., \breve{A}_{i+1}$.

Proof. Each of the $\breve{A}_k$ CIT2 FS in the rule can be rewritten as the union of its IT2 AES, thanks to the *constrained representation theorem* [65].

$$\breve{A}_k = \int_{\tilde{S}_k \in \widetilde{\mathrm{CAES}}_{\breve{A}_k}} \tilde{S}_k \tag{4.2}$$

where $\widetilde{\mathrm{CAES}}_{\breve{A}_k}$ is the collection of IT2 AES of $\breve{A}_k$. Therefore, the rule in (4.1) becomes:

$$\begin{aligned} &\text{IF } x_1 \text{ IS } \int_{\tilde{S}_1 \in \widetilde{\mathrm{CAES}}_{\breve{A}_1}} \tilde{S}_1 \text{AND... AND } x_i \text{ IS } \int_{\tilde{S}_i \in \widetilde{\mathrm{CAES}}_{\breve{A}_i}} \tilde{S}_i \\ &\text{THEN } y \text{ IS } \int_{\tilde{S}_{i+1} \in \widetilde{\mathrm{CAES}}_{\breve{A}_{i+1}}} \tilde{S}_{i+1} \end{aligned} \tag{4.3}$$

Similarly to what has been done for the CIT2 antecedent and consequent FSs in (4.2), also the CIT2 fired output $\breve{B}$ generated by the rule can be expressed as the union of its IT2 AES:

$$\breve{B} = \int_{\tilde{S}_{\breve{B}} \in \widetilde{\mathrm{CAES}}_{\breve{B}}} \tilde{S}_{\breve{B}} \tag{4.4}$$

By definition, the FOU of $\breve{B}$ is:

$$FOU(\breve{B}) = \{(x, y)|(x, y, 1) \in \breve{B}\} \tag{4.5}$$

Since $\breve{B}$ can be expressed as the union of its IT2 AES (4.4) and since the FOU is obtained by dropping the third dimension of $\breve{B}$, (4.5) can be rewritten using the T1 AES of $\breve{B}$. In other words, since the third dimension (i.e. the secondary degree) is not needed for the definition of the FOU, it can be defined using the T1 "equivalents" (i.e. obtained by dropping the third dimension) of the sets $\tilde{S}_{\breve{B}}$ in (4.4).

$$FOU(\breve{B}) = \int_{S_{\breve{B}} \in \text{CAES}_{\breve{B}}} S_{\breve{B}} \tag{4.6}$$

The upper bound of the FOU can therefore be expressed as:

$$\overline{\mu}_{\breve{B}}(y) = \sup_{S_{\breve{B}} \in \text{CAES}_{\breve{B}}} \mu_{S_{\breve{B}}}(y) \tag{4.7}$$

As discussed in [65], each of the sets $S_{\breve{B}}$ is generated by replacing each of the CIT2 FS in the rule (4.1) with one of its AES and carrying out the standard Mamdani inference. Therefore the MF of one specific set $S_{\breve{B}}^{c,...,d} \in \text{CAES}_{\breve{B}}$ can be expressed as:

$$\begin{gathered} \mu_{S_{\breve{B}}^{c,...,d}}(y) = \mu_{S_1^c}(x_1') \star ... \star \mu_{S_{i+1}^d}(y) \\ \text{with } S_1^c \in \text{CAES}_{\breve{A}_1}, ..., S_{i+1}^d \in \text{CAES}_{\breve{A}_i+1}, \forall y \in Y \end{gathered} \tag{4.8}$$

with x_l' being the input value for the l^{th} input variable; y represents the output variable and Y its universe of discourse. Specifically, in (4.8) $\breve{A}_1$ (i.e. the first antecedent of the rule (4.1)) has been replaced with its c^{th} AES S_1^c, ..., $\breve{A}_{i+1}$ has been replaced with its d^{th} AES S_{i+1}^d.

Using (4.8) in (4.7), the upperbound of $\breve{B}$ becomes[1]:

$$\overline{\mu}_{\breve{B}}(y) = \sup_{S_1 \in \text{CAES}_{\breve{A}_1}, \ldots, S_{i+1} \in \text{CAES}_{\breve{A}_{i+1}}} \mu_{S_1}(x'_1) \star \ldots \star \mu_{S_{i+1}}(y) \tag{4.9}$$

Since the $\star$ operators are T-norms, the maximum value of $\overline{\mu}_{\breve{B}}(y)$ is obtained by maximizing each of the terms:

$$\overline{\mu}_{\breve{B}}(y) = \sup_{S_1 \in \text{CAES}_{\breve{A}_1}} \mu_{S_1}(x'_1) \star \ldots \star \sup_{S_{i+1} \in \text{CAES}_{\breve{A}_{i+1}}} \mu_{S_{i+1}}(y) \tag{4.10}$$

Remembering that the upperbound of a CIT2 FS $\breve{A}$ is:

$$\overline{\mu}_{\breve{A}}(x) = \sup_{S \in \text{CAES}_{\breve{A}}} \mu_S(x) \tag{4.11}$$

Using (4.11) in (4.10) the following is obtained:

$$\overline{\mu}_{\breve{B}}(y) = \overline{\mu}_{\breve{A}_1}(x'_1) \star \ldots \star \overline{\mu}_{\breve{A}_i+1}(y) \tag{4.12}$$

□

This proves that, in a CIT2 rule, the upperbound of the FOU of the CIT2 fired output is determined only by the upperbound MFs of the CIT2 FSs involved in the rule. Analogously, it is possible to show that the lowerbound of the FOU is determined by the lowerbound of the FSs in the rule.

Corollary 2. *Given a CIT2 fuzzy rule*

$$\textit{IF } x_1 \textit{ IS } \breve{A}_1 \textit{ AND... AND } x_i \textit{ IS } \breve{A}_i \textit{ THEN } y \textit{ IS } \breve{A}_{i+1} \tag{4.13}$$

and the IT2 rule obtained by replacing each CIT2 FS with an equivalent (i.e.

[1]Note how the superscripts of the sets S_k have been dropped since *all* the possible combinations of the AES of the sets $\breve{A}_k$ that generate all the $S_{\breve{B}} \in \text{CAES}_{\breve{B}}$, $1 \leq k \leq i+1$ are now considered.

with the same FOU) IT2 FS

$$IF\ x_1\ IS\ \tilde{A}_1\ AND...\ AND\ x_i\ IS\ \tilde{A}_i\ THEN\ y\ IS\ \tilde{A}_{i+1} \tag{4.14}$$

the two rules produce a fuzzy fired output with the same FOU

Proof. This is a straightforward consequence of the Theorem above. Since in both the IT2 and CIT2 rules, the FOU of the fuzzy output is determined only by the boundary functions of the sets involved in the rule, if each of $\breve{A}$ and $\tilde{A}$ have the same boundary functions they will produce the same FOU, since the inference in both cases is carried out in the same way. □

The boundary functions $\overline{\mu}_{\breve{A}}$, $\underline{\mu}_{\breve{A}}$ of a CIT2 fuzzy set $\breve{A}$ are defined in the same way as the boundary functions of an IT2 fuzzy set [65], i.e. they represent the boundaries of the FOU. Therefore Theorem 4.1 leads to the same results that are obtained when one uses IT2 fuzzy sets [17]. The reason why Theorem 4.1 has to be proved is in the different representation between CIT2 and IT2 fuzzy sets. In the IT2 case, the *representation theorem* holds [44], i.e. each IT2 fuzzy set can be represented as the union of its type-2 embedded sets; for CIT2 fuzzy sets, instead, the *constrained* representation theorem holds, i.e a CIT2 fuzzy set can be represented as the union of its *acceptable* embedded sets. Since the collection of acceptable embedded sets is a subset of all the embedded sets, all the theorems for IT2 sets that make use of the embedded sets need to be proven again for CIT2 fuzzy sets showing that the same results hold when only acceptable embedded sets are considered.

Although Theorem 4.1 is *one* of the reasons behind the improved run-times of the novel algorithm, this way of computing the firing interval of CIT2 rules cannot be used by the exhaustive or sampling method. In fact, as discussed in the first paragraph of this Subsection, these algorithms require a discrete set of firing values (and not just the endpoints of the firing interval) to determine the type-reduced set. Additionally, the analysis of the computational complexity carried out in Sec. 4.2.5 does not include the computation of the firing of the

rules in order to make a fair comparison between the three CIT2 type-reduction approaches.

Algorithm 4 Switch Index Type-Reduction Algorithm

Sort the CIT2 sets partitioning the output variable (i.e. all the sets used as consequents in the rulebase) in ascending order based on the min value of their support set
Give each sorted set $\breve{C}$ an ordinal index, obtaining the list $(\breve{C}_1, ..., \breve{C}_n)$
$S = \emptyset$ ▷ This set, will contain the centroid values of all the AES generated by the switch index approach
for each $\breve{C}_i \in (\breve{C}_1, ..., \breve{C}_n)$ **do**
    $F_i^L = 0$, $F_i^U = 0$ ▷ They will store the maximum lower and upper firing strength for C_i
    for each rule R in which $\breve{C}_i$ appears as a consequent **do**
        Compute $R_F.lower$ and $R_F.upper$, respectively the lower and upper bounds of the firing interval of R
        $F_i^L = \max(F_i^L, R_F.lower)$, $F_i^U = \max(F_i^U, R_F.upper)$
    end for
end for
for *index*=1 to n **do**
    for $\breve{C}_i \in (\breve{C}_1, ..., \breve{C}_n)$ **do**
        if $i <$ *index* **then**
            FS-1 $= F_i^L$
            FS-2 $= F_i^U$
        else
            FS-1 $= F_i^U$
            FS-2 $= F_i^L$
        end if
        Apply the implication operator on the rightmost AES of $\breve{C}_i$ using *FS-1*, obtaining $\overline{C}_i$
        Apply the implication operator on the leftmost AES of $\breve{C}_i$ using *FS-2*, obtaining $\underline{C}_i$
    end for

23: $\overline{C} = \bigcup_{1 \le i \le n} \overline{C}_i$ ▷ Do the union of the sets $\overline{C}_i, ..., \overline{C}_n$

24: $\underline{C} = \bigcup_{1 \le i \le n} \underline{C}_i$

25: $S = S \cup \{\text{centroid}(\overline{C})\} \cup \{\text{centroid}(\underline{C})\}$

26: **end for**

27: **return** the minimum and maximum centroid values $x_L, x_U \in S$

4.2.3 The algorithm

In this Subsection, a formal description of the algorithm is provided (Algorithm 4). As already mentioned, the idea is to find the *switch indices* that produce the AESs with the highest and lowest T1 centroid values. The algorithm described here, works with a single output variable at a time. In other words, it must be executed once for each output generated by the system. For simplicity, the analysis carried out in this chapter assumes that the CIT2 FLS only produces one output (i.e. it has only one output variable).

In the *for-loop* starting at line 11, different AESs are generated, testing all the possible switch index values. At the end of the procedure, the highest and lowest T1 centroid among all the AESs that have been generated, are used as the endpoint of the type-reduced set returned as an output. The identification of the switch indices uses a brute force approach. This method has been chosen for its simplicity and as a first strategy to compute the novel concept of switch indices introduced in this chapter. In future work, the mathematical properties of the AESs and the switch indices themselves will be analyzed to establish a criterion or a mathematical formula that could directly determine the right switch indices, similarly to what happens in the KM algorithm with the switch points.

Conceptually, the algorithm can be summarized in the following steps:

1. Give each CIT2 consequent MF an ordinal index by sorting them in ascending order of the minimum value of their support set, obtaining the list $(\breve{C}_1, ..., \breve{C}_n)$.

2. For each consequent $\breve{C}_i$, compute its lower firing value F_i^L as the maximum lower firing strength of all the rules where it appears as a consequent; analogously, compute its upper firing value F_i^U as the maximum upper firing strength of all the rules where it appears as a consequent.

3. If computing the right endpoint of the type-reduced set (i.e. to generate

the AES with the maximum centroid value), replace each consequent MF with its rightmost AES; if computing the left endpoint, take the leftmost AES instead.

4. Test all the possible switch index values, between 0 and the maximum index given to the consequent MFs:

   i. If computing the left endpoint, apply the inference operator on each replaced consequent MF C_i using F_i^U if the MFs has an index smaller than the switch index, use F_i^L otherwise; for the right endpoint instead, use F_i^L before the switch index and the F_i^U after it.

   ii. Do the union of the AES resulting from the inference and defuzzify the set obtained, computing its centroid.

5. Return, as the final type-reduced set, the lowest and highest centroid values obtained from the defuzzification at the previous step.

A representation of the intermediate results of these steps can be found in Sec. 4.3.2 and in Fig. 4.2: Fig. 4.2.a, shows the partitioning of an output variable by three sets (e.g. low, medium and high temperature); Fig. 4.2.b shows a possible FOU that is obtainable from a CIT2 FLSs that uses the MFs in Fig. 4.2.a as consequent sets. Fig. 4.2.c, instead shows graphically the effect of the step 4).i of the algorithm (Sec. 4.2.3) in which the implication operator is applied to the leftmost AES of all the consequent sets. Fig. 4.2.d shows the final AES produced as the union of the sets in Fig. 4.2.c.

4.2.4 Mathematical description

The exhaustive approach evaluates every combination of every *embedded set* at every firing strength that arises from each individual rule in combination in the output. Empirically, it has been observed that the combination of sets that produced the AES with the lowest (left-most) centroid, follow the AES obtained by carrying out the implication with the upper value in the firing

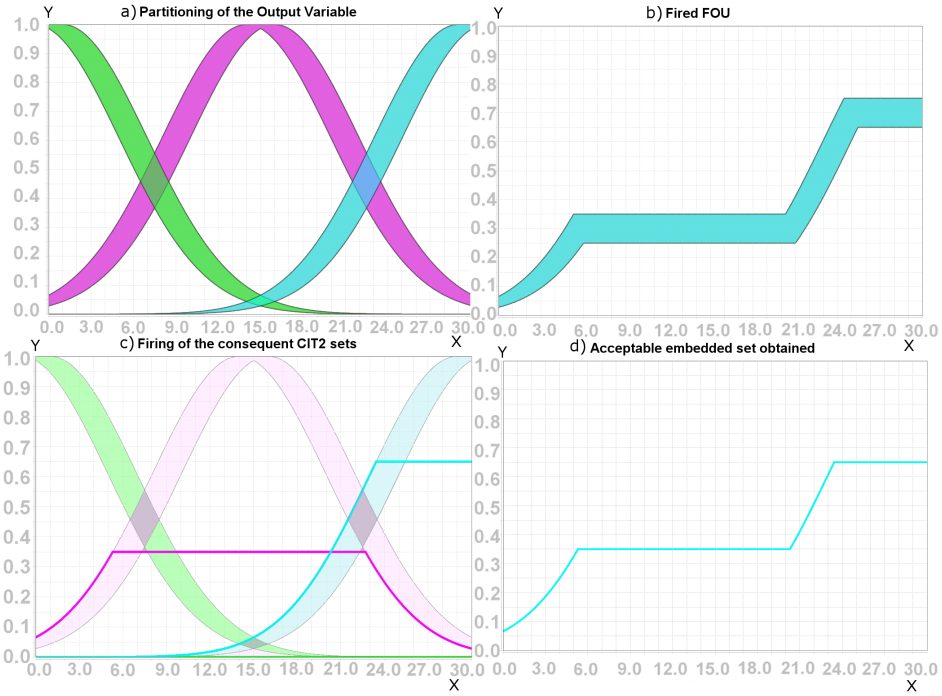


Figure 4.2: *Creation of the AES of the fired output (2.) that determines the left endpoint of the constrained centroid. First the partitioning of the output variable (1.) is shown, then for each consequent MF one AES is selected and the implication operator applied (3.). Finally, the inferenced sets are aggregated to produce the final AES (4.).*

interval on the leftmost AES of the consequent sets for some left-hand portion of the universe, before switching at some point to following the left AES with the lower value in the firing interval for the remainder of the universe (and vice versa for the highest centroid). This observed behaviour has inspired the current algorithm to determine this switch-point and use the acceptable embedded sets with these properties for the type-reduction. An example of this phenomenon is shown in Fig. 4.3, in which the fired FOU is obtained as described in Fig. 4.2. In this case, for the magenta section, the leftmost embedded set obtained with the higher firing value is used; the green section is where the switch happens and the left AES with the *lower* firing value is used instead.

Formally, the problem solved by the algorithm to compute the left endpoint of the constrained type-reduced set can be modelled mathematically as follows (the right endpoint can be expressed analogously):

Figure 4.3: *The fired FOU of a CIT2 FLS (shaded) and the AES with the lowest centroid value. The magenta section of the AES is obtained following the leftmost AES after the implication with the upper firing value in the firing interval, while the section in green is obtained following the leftmost AES after the implication with the lower firing value.*

$$\text{Left\_endpoint} = \min_{0 \leq SI \leq n} \left(\text{Centroid}\left(\bigcup_{1 \leq i \leq n} C_i^{L'} \right) \right) \tag{4.15}$$

$$\mu_{C_i^{L'}}(x) = \begin{cases} \min(\mu_{C_i^L}(x), F_i^U), & i < SI \\ \min(\mu_{C_i^L}(x), F_i^L), & i \geq SI \end{cases} \tag{4.16}$$

where SI is the switch index, C_i^L is the leftmost AES of the $i-th$ consequent set, $C_i^{L'}$ is the set obtained after the implication on C_i^L, F_i^U and F_i^L are the maximum and minimum firing strength among all the rules in which $\breve{C}_i$ appears as a consequent (computed as in Algorithm 4 at line 8).

Determining whether Algorithm 4 computes the same type-reduced set as the exhaustive approach is not straightforward. In fact, in the exhaustive version, *all* the AES of each consequent $\breve{C}_i$ are considered and the possible firing strength F_i in the min operator in (4.16) could be *any* value in $[F_i^U, F_i^L]$.

Additionally, the union of the AES before the centroid computation in (4.15) may produce a non-convex and non-normal set (such as that in Fig. 4.3) while the overlapping of the MFs of each AES also plays a role in the final result and makes the problem challenging to solve from a mathematical point of view. For these reasons, the formal relationship between Algorithm 4 and the exhaustive method needs to be studied in future work.

For now, the usefulness of the novel algorithm has been shown in the extensive tests reported in Sec. 4.3. Indeed, in all experiments undertaken so far, Algorithm 4 produces the same as the exhaustive method.

4.2.5 Analysis and computational complexity

The analysis carried out here does not include the computation needed to determine the firing strength of the rules (lines 4-9). In all the case studies examined in Sec. 4.3, the firing intervals of the rules are computed in the same way they are computed in IT2 inference, using Theorem 4.1.

Before the algorithm can build the AESs, it is necessary to sort the n consequent sets used in the CIT2 FLS in ascending order of the minimum value in their support set, which requires $\mathcal{O}(n \log n)$ operations. Once the consequents are sorted, for each of the n iteration of the *for-loop* at line 11, two AES are generated, with each generation requiring $\mathcal{O}(n)$ operations (because of the union at lines 23, 24). The defuzzification at line 25 requires $\mathcal{O}(kn)$ operations with k being the discretization level used and assuming that for each discretized point x its membership degree with respect to $\overline{C}$ is computed as:

$$\mu_{\overline{C}}(x) = \max_{1 \leq i \leq n} \mu_{\overline{C}_i}(x) \tag{4.17}$$

and the membership degree of x with respect to $\underline{C}$ is calculated in the same way. Therefore, the final computational complexity of the algorithm is $\mathcal{O}(2kn^2)$, where n is the number of MFs that partition the output variable. This represents a significant improvement when compared to the original exhaustive algorithm that had a computational complexity of $\mathcal{O}(k^{n+1})^m$ where m is the number of rules, n the number of antecedents per rule and k the number discrete number of AES that had to be selected for each of the CIT2 FSs in the CIT2 FLS [65]. A comparison of the run times of the novel procedure, the sampling method, and the exhaustive algorithm is presented in Section 4.3.

4.3 Practical Applications

This section is focused on the application of Algorithm 4 in three case studies for the comparison of this novel approach with other type-reduction methods. The first subsection shows a run time comparison between KM, EKM, CIT2 sampling, CIT2 exhaustive and Algorithm 4 in the type-reduction of a large number of CIT2 FSs. The second part of the section, instead, compares the different constrained approaches in terms of endpoints of the produced type-reduced set to analyze their differences. Lastly, the third subsection presents a qualitative comparison between Algorithm 4, the sampling method and EKM in a real-world case study. Specifically, the problem of the recommendation of post-operative breast cancer treatment is analyzed. The accuracy values of the different approaches are compared, together with the interpretability of the classifications that they produce.

4.3.1 Run time comparison

The experiments reported here, consist in the type-reduction of a number of FSs produced as the output of a CIT2 FLS. Since the computational complexity of Algorithm 4 is $\mathcal{O}(kn^2)$ with n being the number of MFs that partition a given output variable and k being the discretization level used to defuzzify the AES, the experiments involve output variables partitioned with a different number of MFs. By doing this, it is possible to see how the algorithm performs as the cardinality of the partitioning increases.

The experimental setup is the following: 4 FLS have been produced with the output variable partitioned respectively with 2, 3, 5 and 7 MFs. Each of these MFs is used as the consequent of a different fuzzy rule with a single antecedent MF and one input variable. Therefore, a FLS with a partitioning size of n has n rules. The generator sets used in this experiments are triangular MFs with parameters $(x-1,\ x,\ x+1)$, $x \in \mathbb{N}$, $1 \leq x \leq 7$. The displacement set used to generate the FOUs is the interval $[-0.5, 0.5]$ and the resulting sets

can be seen in Fig. 4.4. The minimum operator has been used to carry out the implication.

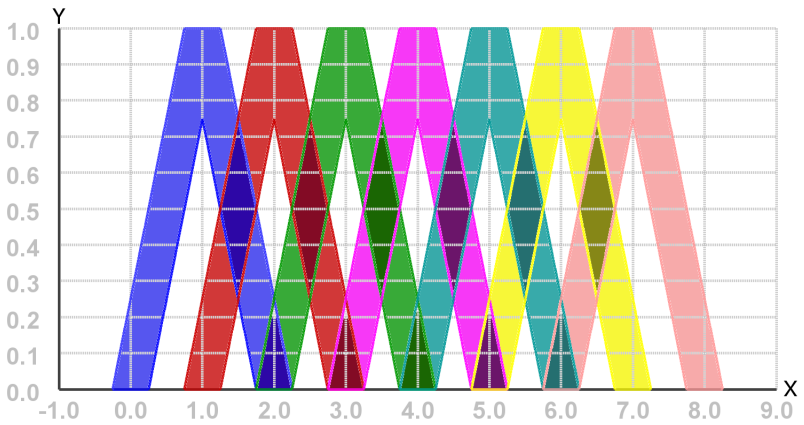


Figure 4.4: *Fuzzy sets used for the experiment in Sec. IV-A*

Each system has been run 5×10^6 times and its outputs type-reduced using different algorithms. The input values have been set randomly, whilst maintaining that each rule always fires with a minimum firing strength of 0.1.

The methods tested are KM [18], EKM [19], the sampling CIT2 method [65] with 50 samples with uniform random distribution (CIT2-S50) [65], the exhaustive method (CIT2-Exh.)[65] and the novel procedure introduced in this chapter (Algorithm 4) (CIT2-SI). Additionally, the generator sets of the CIT2 FLSs have been used to create a T1 version of the FLSs described above to compare the run times of these T2 FLSs with their T1 counterparts. For the exhaustive CIT2 approach, 5 AES have been considered for each CIT2 FS (the generator set plus 2 AES at its left and 2 at its right, uniformly distributed). The experiments have been run in Java on a Windows machine with an i7-7600U CPU. For the KM, EKM and T1 FLSs implementations, the Juzzy library [64] has been used. To defuzzify the T1 AES, T1 ES and the output of the T1 FLS, they are uniformly discretized in 1000 points and their centroid

is computed.

Table 4.1: Running times (in seconds) of the different approaches.

MFs	**T1**	**KM**	**EKM**	**CIT2-Exh.**	**CIT2-S50**	**CIT2-SI**
2	12.29s	196.57s	72.35s	635s	1077.21s	**19.09s**
3	26.00s	285.60s	107.58s	2655s	1424.17s	**62.94s**
5	41.15s	479.54s	198.56s	35600s (est.)	2323.88s	**191.76s**
7	53.77s	780.25s	**247.32s**	714170s (est.)	2979.54s	398.88s

The run times of the different approaches are reported in Table 4.1. The minimum value in each row **among the T2 approaches** is highlighted in bold. As it is possible to see, Algorithm 4 (CIT2-SI in the table) is at least 7.5 times faster in all the cases when compared to the sampling type-reduction technique. In addition to that, CIT2-SI performs overall better than all the other approaches, being slower than EKM (but still faster than KM) only when the output is partitioned with more than 5 MFs. For the exhaustive approach in the last two FLS (CIT2-Exh. with 5 and 7 MFs), only 1000 type-reductions have been performed and then their run time multiplied by 5000 to obtained an estimate of the total time it would be required to perform 5×10^6 type-reductions using that algorithm due to its impractical computational time.

Although it has been been shown that run times are heavily affected by the specific programming language used to implement the type-reduction algorithm [66], the significant difference of at least one order of magnitude between the presented approach (CIT2-SI) and the other CIT2 algorithms (CIT2-Exh. and CIT2-S50) can hardly depend on implementation details. The relationship between CIT2-SI, KM and EKM, however, may be different in other programming languages, since the specific timings of each depend on both the algorithm and the programming language used.

4.3.2 Comparison between the constrained approaches

To compare the type-reduction set produced by the three different approaches (exhaustive, sampling and switch index) a FLS for a simplified version of the iris problem [61] is analyzed. In the original version, 4 input variables are used (sepal and petal length and width) to identify the type of iris plant. In this version, only 2 of them are used: petal length and width. This choice has been made because the computational time for the exhaustive approach grows very quickly with the number of antecedents and rules of the FLS. Therefore, in order to be able to use it for this comparison, a compact rule-base and a small number of input variables are necessary. Each variable is partitioned with 3 labels (*low*, *medium* and *high*) used to create the following 5 rules:

1. If petal length is low and petal width is low then species is setosa.

2. If petal length is medium and petal width is medium then species is versicolor.

3. If petal length is high and petal width is high then species is virginica.

4. If petal length is medium and petal width is high then species is virginica.

5. If petal length is high and petal width is medium then species is virginica

To run the exhaustive algorithm, each CIT2 FS involved in the system has been discretized in 5 AES: the generator set plus 2 AES at its left and 2 at its right, evenly distributed. Additional details on the MFs used in this experiment can

be found in Table 4.2.

Table 4.2: Membership function used in the iris system

Name	**Shape**	**Parameters**	**Displacement Set**
Length Low	Gaussian	(std.dev.=1.2, mean=1)	[-a, a], a=5% UOD
Length Medium	Gaussian	(std.dev.=0.9, mean=3.8)	[-a, a], a=5% UOD
Length High	Gaussian	(std.dev.=1.2, mean=7)	[-a, a], a=5% UOD
Width Low	Gaussian	(std.dev.=0.6, mean=0	[-a, a], a=5% UOD
Width Medium	Gaussian	(std.dev.=0.35, mean=1.25)	[-a, a], a=5% UOD
Width High	Gaussian	(std.dev.=0.7, mean=2.5)	[-a, a], a=5% UOD
Setosa	Triangular	(A=0, B=1, C=2)	[-a, a], a=5% UOD
Versicolor	Triangular	(A=1, B=2, C=3)	[-a, a], a=5% UOD
Virginica	Triangular	(A=2, B=3, C=4)	[-a, a], a=5% UOD

For the sampling method, the results have been obtained as the average of 50 executions of the sampling method computed with 50 samples each time. The standard deviation for this approach is also reported. The T1 AES selected by the different approaches are discretized in 1000 points to be defuzzified. In Table 4.3, the interval representing the type-reduced set for the 3 approaches is reported for 3 different input values, one for each of the possible species.

Table 4.3: Comparison of the different constrained type-reduction methods

Inputs	**Exhaustive**	**Sampling**	**Switch Index**
(0.2, 1.4)	[0.83, 1.26]	[0.85±0.01, 1.22±0.01]	[0.83, 1.26]
(1.4, 4.7)	[2.08, 2.69]	[2.19±0.04, 2.57±0.03]	[2.08, 2.69]
(2.1, 6.6)	[2.77, 3.18]	[2.80±0.01, 3.16±0.01]	[2.77, 3.18]

In all the cases both the switch index and the exhaustive approach produce the same result while the sampling gives a slightly different value. Table 4.4,

shows the average absolute difference (for both the endpoints of the type-reduced set) between the sampling and switch index procedures with respect to the exhaustive method over the 150 entries of the iris dataset. In other words, each entry is a pair $[x, y]$ representing the average absolute difference between two approaches for the left (x) and right (y) endpoint of the interval representing the type-reduced set.

Also the standard deviation is reported. As can be seen, in the FLS analyzed here there is no difference between the switch index approach and the exhaustive one. At the moment, it can not be proven whether they always produce the same results or this only happens in a subset of situations, perhaps caused by the specific MFs, discretization or partitioning used. The relation between Algorithm 4 and the exhaustive approach will be further studied in future work with a formal analysis and additional case studies.

Table 4.4: Average absolute difference between the approaches

	Sampling	**Switch Index**
Exhaustive	[0.06±0.04, 0.06±0.04]	[0.0, 0.0]

Step-by-step application of the algorithm

The iris CIT2 FLS presented above, will be used to illustrate each step of Algorithm 4, in order to clarify how the procedure works. In this example the input value for the petal length is 1 while its width is 3. The three MFs modeling respectively the setosa, versicolor and virginica species are represented (shaded) in Fig. 4.5. Algorithm 4 sorts them using the leftmost value of their support set in order to give each one of them an ordinal index (line 2). In this case, the index of setosa (in blue) is 0, since it is the leftmost CIT2 FS partitioning the output variable, while the indices of versicolor and virginica (in red and green) are respectively 1 and 2.

Then, the firing interval for each rule is computed (for-loop at line 4) . The

firing strengths of the rules in the system are the following:

```
1: IF Length IS Low AND Width IS Low THEN Species IS
     Setosa: [0.18, 0.32]
2: IF Length IS Medium AND Width IS Medium THEN Species IS Versicolor:
    [0.51, 0.82]
3: IF Length IS High AND Width IS High THEN Species IS Virginica:
    [0.03, 0.06]
4: IF Length IS Medium AND Width IS High THEN Species IS Virginica:
    [0.07, 0.14]
5: IF Length IS High AND Width IS Medium THEN Species IS Virginica:
    [0.03, 0.06]
```

For each of the three classes, the firing interval is computed as the maximum lower and maximum upper values of the firing strength of the rules in which they appear as consequent. In this case, the firing interval of each class are:

```
Setosa: [0.18, 0.32]
Versicolor: [0.51, 0.82]
Virginica: [0.07, 0.14]
```

At line 20 of Algorithm 4, the implication operator (minimum) is then applied to the rightmost AESs of the three classes. The rightmost AES for each of the classes is represented with a solid line in Fig. 4.5. Line 21 carries out the implication on the leftmost AES. Since the two operations are very similar, only line 20 will be analyzed.

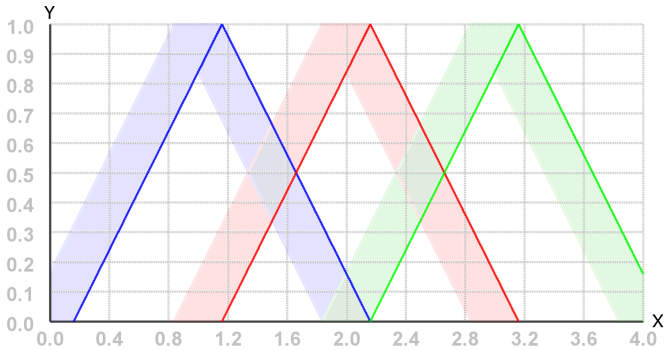


Figure 4.5: *CIT2 fuzzy sets modeling the three iris classes (shaded) and their rightmost AES*

Before doing the implication, the procedure selects a current switch-index value to try for the current iteration of the for-loop at line 11. The result of line 20 for all the possible switch-index values tested by the for-loop is shown

in picture Fig. 4.6.

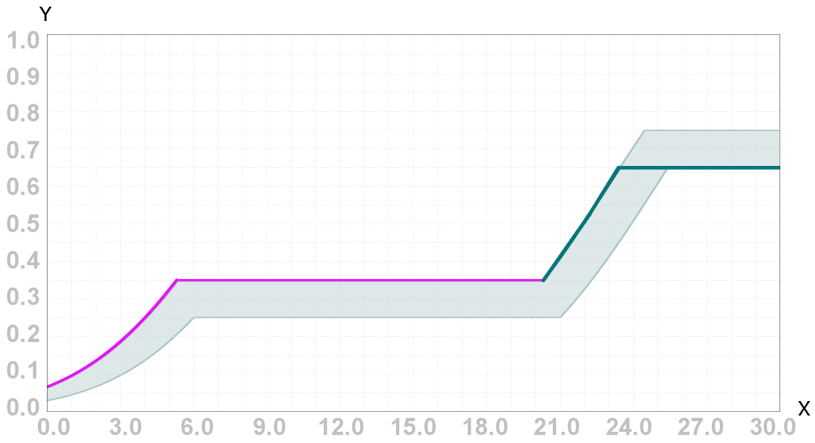


Figure 4.6: *Result of the implication with different switch-index values*

If the switch index value is smaller than the index of the class, then the lower firing value is used for the implication, otherwise the upper firing value is used. For each of the switch index values, a single set is produced by doing the union of the three classes after the implication. The set obtained at this stage is an AES of the fuzzy output of the FLS. These three AESs obtained from the union are then defuzzified and their centroid values stored in a list S (line 25). After also line 21 is computed and the centroid values produced by it are added to S, the interval $[\min(S), \max(S)]$ is returned as the value of the type-reduced set.

4.3.3 Real-world application

In this subsection, the novel algorithm is qualitatively compared to the EKM procedure and sampling method on a real-world classification task.

The problem analyzed in this chapter is the recommendation of post-operative therapy for breast cancer. In this case both the interpretability and the explainability of the system play a crucial role. An interpretable system is made of MFs with a clear semantic meaning (i.e. a linguistic label) and a rule-base composed of a limited number of rules [10]. This allows a non-expert audience, i.e. the physicians in this case, to get an intuitive understanding of the rules followed by the system to produce the final classification. Explainability,

instead, is defined as the ability to "explain the user the process it followed to make the output decision" [10]. In other words, the system must provide an explanation for each of the classifications produced. Therefore, in FLS for XAI it is important to use defuzzification algorithms with a type-reduction process that can produce explanations for the outputs of the FLS.

The goal of the system proposed here, is to determine whether a chemotherapy treatment may or may not be beneficial as a post-operative treatment. This decision problem was first described by Garibaldi et al. [41].

To provide a final recommendation to the patient, a multi-disciplinary group of physicians decide on the most effective therapy to recommend. In this case, the goal of the system is to replicate the decision of the group of doctors with respect to the recommendation of chemotherapy only.

The Nottingham University Hospitals NHS Trust clinical guidelines for adjuvant therapy following surgery (reproduced verbatim).

NPI < 3.0	None
NPI 3.1–3.4	
ER +ve	Recommend hormone therapy
ER –ve	Recommend chemotherapy if VI
NPI 3.4–4.4	
ER +ve	Recommend hormone therapy
ER –ve	Recommend chemotherapy
NPI > 4.4	
ER +ve	Discuss chemotherapy
	Consider
	Recommending chemotherapy
	Age <40
	VI
	HER-2 +ve
	Weak ER(< 100/300)
	Recommending Against Chemotherapy
	Age >60
	Only 1 LN positive
	Special type cancer
ER -ve	Recommend Chemotherapy

ER +ve: ER is positive.
ER –ve: ER is negative.
Age: in years.
HER-2: Human Epidermal growth factor Receptor 2.
VI (Vascular Invasion): presence of unequivocal tumor in vascular spaces.

Figure 4.7: *The protocol for the recommendation of chemotherapy*

To make the fuzzy system interpretable, it has been built starting from the clinical protocol used by the Nottingham University Hospitals NHS Trust (Fig. 4.7), generating the rule-base shown in Fig. 4.8.

The system has the following five inputs:

- NPI: Nottingham Prognostic Index, an index that indicates the prognosis

Rule	Antecedent	Consequent
1	IF (*NPI* is *Low*)	THEN (*Chemo* is *No*)
2	IF (*NPI* is *Medium low*) and (*ER* is not *Negative*)	THEN (*Chemo* is *No*)
3	IF (*NPI* is *Medium low*) and (*ER* is *Negative*)	THEN (*Chemo* is *Maybe*)
4	IF (*NPI* is *Medium high*) and (*ER* is not *Negative*)	THEN (*Chemo* is *No*)
5	IF (*NPI* is *Medium high*) and (*ER* is *Negative*)	THEN (*Chemo* is *Yes*)
6	IF (*NPI* is *High*) and (*ER* is not *Negative*)	THEN (*Chemo* is *Maybe*)
7	IF (*NPI* is *High*) and (*ER* is not *Negative*) and (*Age* is *Young*)	THEN (*Chemo* is *Yes*)
8	IF (*NPI* is *High*) and (*ER* is not *Negative*) and (*VI* is *Yes*)	THEN (*Chemo* is *Yes*)
9	IF (*NPI* is *High*) and (*ER* is *Weak*)	THEN (*Chemo* is *Yes*)
10	IF (*NPI* is *High*) and (*ER* is not *Negative*) and (*Age* is *Old*)	THEN (*Chemo* is *No*)
11	IF (*NPI* is *High*) and (*ER* is not *Negative*) and (*LN* is *Negative*)	THEN (*Chemo* is *No*)
12	IF (*NPI* is *High*) and (*ER* is *Negative*)	THEN (*Chemo* is *Yes*)

Figure 4.8: *Rule-base obtained from the protocol shown in Fig. 4.7*

after the surgery. It is calculated from three criteria: size of the lesion, number of involved lymph nodes and tumor grade. For this variable, 4 linguistic label (and therefore, 4 FSs) were identified from the recommendation protocol: *low*, *medium-low*, *medium-high* and *high*. The cut-off points between the labels are respectively 3.0, 3.4 and 4.4. The universe of discourse (UOD) is the interval [0,10].

- ER: Estrogen Receptor test result, it shows whether estrogen fuels the tumor. This can be used to decide if hormone-suppression treatment would be beneficial. The linguistic labels in this case are *negative*, *weak* and *positive*, with the cut-off points being 20 and 100. The UOD is the interval [0, 300].

- Age: the age of the patient. The labels are *young*, *middle age* and *old*, with their respective cut-off points being 40 and 60 while the UOD is [0, 90].

- VI: Vascular Invasion, represents the presence of unequivocal tumor in vascular spaces. It has three labels, *yes*, *maybe*, *no* with the cut-off points being 1.5 and 2.5. The UOD is [1, 3].

- LN: positive Lymph Node ratio, it's the ratio of lymph nodes that are positive to cancer change on the total sample of tested lymph nodes. The labels in this case are *negative* and *positive* with the cut-off point being 0.03.

The description of these input variables is based on material previously presented in the original paper [41]. The output variable, instead, is the *chemotherapy recommendation* that is partitioned in three labels, *yes*, *no* and *maybe*. The *yes* and *no* cases, represent respectively a recommendation in favour and against the chemotherapy. The *maybe* case, instead, represents a situation in which an agreement among the physician could not be reached and therefore a clear recommendation can not be provided; as a consequence of that, the administration of the chemotherapy is further discussed with the patient.

To build interpretable MFs that keep their semantic meaning and cut-off points but also obtain a FLS with good performance, the following optimization process has been implemented. The T1 MFs used for the input variables of the VI-F FLS in [41] are used as a starting point by a genetic algorithm. To carry out the optimization in a way that keeps the cut-off points intact, the intersection points of the MFs remain unchanged and only the slopes of the intersecting segments of the MFs are tuned. For example, consider the T1 MFs for the *age* variable, as shown in Fig. 4.9 [41].

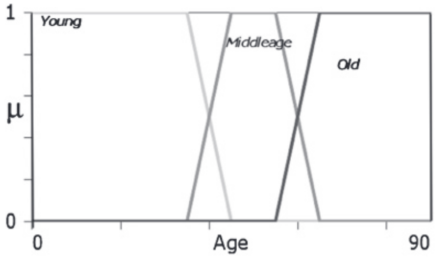


Figure 4.9: *Unoptimized T1 MFs for the age variable. From left to right, they model the words* young, middle age *and* old.

The goal of the genetic algorithm is to find the optimal slopes for the intersecting oblique lines of the *young*, *middle age* and *old* MFs. By doing that,

their intersection points and therefore the cut-off points between them remain unchanged. The same optimization process is used for all the MFs partitioning the input variables to ensure a high level of interpretability of the systems and the adherence to the protocol described in Fig. 4.7. The parameters of the genetic optimization are reported in Table 4.5.

Table 4.5: Parameters used for the genetic optimization in the breast cancer recommendation FLS

Parameters	**Values**
Population size	100
Iteration limit	100
Crossover	Single-Point
Crossover rate	0.9
Elitism	5%
Mutation rate	1/chromosome_size
Fitness function	Accuracy value

For the output variable instead, there are no indications in the protocol that can help build the three MFs (*yes, maybe, no*). For this reason, they have been designed as follows: the *maybe* MF is modeled as an isosceles triangles centered in 50 (the midpoint of the UOD) while its width is determined by the genetic algorithm. The *yes* and *no* MFs, instead, are shoulder MFs respectively ending and starting in the midpoint of the UOD. The cut-off points are the ones with a membership value of 0.5 in the *maybe* MF. An example of the partitioning generated by the genetic algorithm for the output variable *chemotherapy recommendation*, is shown in Fig. 4.10.

The process described so far, generates the T1 MFs that can be used as GSs of the CIT2 MFs. To obtain CIT2 MFs, however, also the displacement set (DS), i.e. the shifting values to generate the FOU, needs to be determined. The choice of the width of the DS for each CIT2 MF is made by the genetic algorithm. The FLS returned at the end of the optimization is the one with

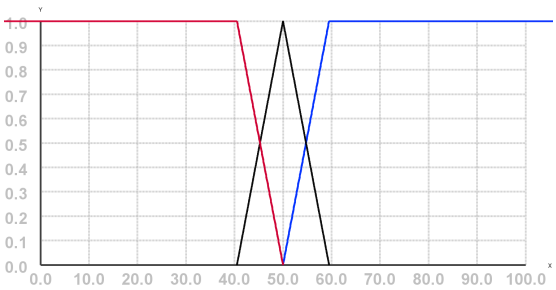


Figure 4.10: *A possible partitioning of the chemo recommendation output generated by the genetic algorithm, The MFs represent the following labels, from left to right: No, Maybe, Yes*

the highest accuracy value on the training set.

The real-world dataset used for the optimization of the system is the same one presented in the original paper [41]. However, due to its imbalanced nature, only some of its entries have been selected. Specifically, all the 191 *yes*, all the 52 *maybe* and 191 *no* cases have been chosen, for a total of 434 instances.

The optimization has been run four times to generate a T1 FLS, an IT2 FLS and 2 CIT2 FLS using respectively the sampling method and Algorithm 4 for the type-reduction step. The process to obtain the T1 FLS is the same one used to determine the GS of the IT2 and CIT2 FLS. The genetic optimization to obtain the FOUs of IT2 and CIT2 FLS is the same. To run the systems, the Mamdani inference is used, with the *min* function implementing the AND and implication operators while the EKM type-reduction procedure is used for the IT2 FLSs. The final output of the system is calculated as the mid-point (centroid) of the type-reduced set. This value is then converted into a class using the cut-off points between the chemo MFs *no*, *maybe* and *yes*. Although the endpoints of the type-reduced set are not *directly* used at this step of the classification in this example, they are very useful in the development of explainable systems. In fact, producing an interval as an output rather than a crisp value and being able to explain how the interval has been generated would provide additional information to the end user regarding the effects of the uncertainty on the final classification (i.e. the width of the interval), thereby clearly showing the decision process followed by the FLS.

The accuracy values of each of these systems have been computed as the average of a 5-fold cross validation approach repeated 5 times for a total of 25 executions per system. The results are reported in Table 4.6. All the FLSs have been designed in Java; the T1 and IT2 FLSs have been implemented with Juzzy [64] while Juzzy Constrained [67] has been used for the CIT2 FLSs.

Table 4.6: Results of the different genetic FLS

FLS	**Accuracy**
T1	70.762%
IT2 (EKM)	71.826%
CIT2 (Switch Index)	72.568%
CIT2 (Sampling, 50 samples)	72.845%

The data shows that the IT2 and the 2 CIT2 FLSs perform better than the T1 one; both the CIT2 also show a higher accuracy than the IT2 FLS, with the CIT2 FLS with the sampling method having the best performance (0.277% better than the switch index algorithm). Being this comparison only based on a single case study with a specific tuning algorithm, it is not sufficient to make any claims on which modeling approach, i.e. IT2 or CIT2, performs better and under which circumstances. The main goal of this case study is to provide a worked example of the novel algorithm proposed in this chapter, and show its potential in terms of its use in XAI applications, as discussed in the next subsection. However, a more formal comparison, using multiple datasets and a statistical analysis will be carried out in future work to get a better understanding of which approach is better in which situations.

4.3.4 Interpretability

With an IT2 fuzzy system, regardless of the type-reduction method used, it is possible to provide an explanation for the outputs of the system by analysing the rules that fired with a given set of inputs. Following a novel approach

proposed by Mendel [51], any input can be linked to its IT2 first-order rule partition from which it is possible to determine the firing rules. These can then be shown to the end-user as an explanation for the output produced.

As a further enhancement to this capability, CIT2 fuzzy systems have the ability to also explain the type-reduced set. When a designer wants to explicitly model the effects of the uncertainty on the decision process, the interval obtained from type-reduction can be provided as the system output. An application of this concept is shown in Sec. 4.3.2, where the firing of each class is reported as an interval; the same strategy can also be applied to the chemotherapy recommendation scenario, in which the system output is represented by an interval, e.g. [75, 90], showing how much the FLS is in favour of the chemotherapy treatment and how certain or uncertain its decision is. With a CIT2 FLS, the specific rules and inputs that determine *each of the endpoints of the interval*, i.e. 75 and 90 in this example, can be identified by analyzing the AESs that lead to those values during the type-reduction, as illustrated by the following analysis.

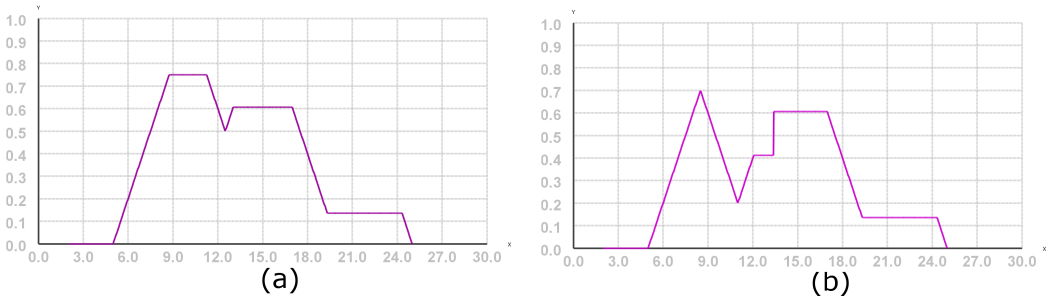


Figure 4.11: *ESs that determine the right value of the EKM (a) and CIT2 (b) centroid.*

Fig. 4.11 shows one of the ES selected by the EKM procedure and the AES chosen by Algorithm 4 to type-reduce an output of the system. In other words, these are the ESs chosen by the procedures to obtain the right endpoints of the type-reduced set. In the CIT2 case, by looking at the way those AESs are generated, it is possible to see the contribution of each of the consequent MFs to the final result as well as the firing strengths obtained from the input values.

The AES in Fig. 4.11.b has been obtained as the union of the sets shown in Fig. 4.12. The latter sets, represent all the $\overline{C}_i$ at line 23 of Algorithm 4, before the union. Through this analysis, it is possible to see that the *no* MF (the one in blue) was fired with a strength of 0.45, the *maybe* one (in the middle) with a value of 1 and *yes* with a value of 0.39. Additionally, it is possible to identify which rules generated the firing strengths (line 8 of Algorithm 4), making possible the generation of a textual explanation for each of the endpoints of the type-reduced set, similar to what can already be done for the outputs of T1 FLS (e.g. [11, 35]).

Linking each ESs identified by the KM procedure to rules or inputs of the systems, on the other hand, can be challenging. In fact, for the resolution of the well-defined mathematical problem carried out by the KM procedure, it makes no difference if the IT2 fuzzy set to type-reduce has been obtained as the output of an IT2 FLS or not. The procedure is, in fact, unaware of the existence of the rulebase.

The ability to use the algorithm proposed in this chapter in order to produce explanations has been further explored in Chapter 5.

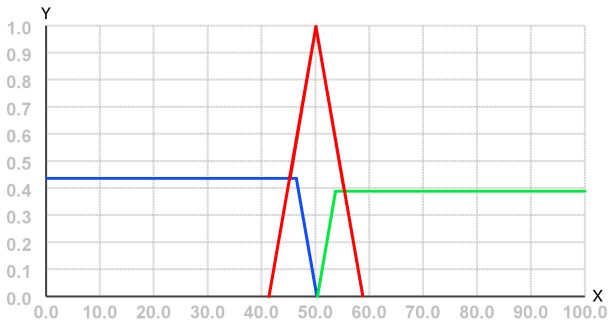


Figure 4.12: *The unions of these sets generates the AES shown in Fig. 4.11.b*

4.4 Summary

CT2 FSs have been proposed as a way to increase the interpretability and explainability of T2 FSs [1], being a specific way of generating T2 FSs when starting from a T1 MF modeling the same concept. Particularly, CIT2 FS

have been previously described and analysed, showing how they can be used to produce CIT2 FLS with a high level of explainability [65, 68]. However, the two original type-reduction procedures originally presented, had the drawback of being significantly slower than the widely used KM [18] procedure.

In this chapter, a novel inference and type-reduction algorithm for CIT2 FSs has been presented, based on the idea of *switch indices* rather than the *switch points* used in the KM procedure.

The running times of the novel algorithm presented in this chapter have been compared to different T2 type-reduction procedures (KM, EKM, CIT2-S50), showing better performance in three of the four tests carried out.

Finally, a real-world classification application has been used as a case study to have a qualitative comparison in terms of accuracy and interpretability between the algorithm produced in this chapter and the widely adopted EKM procedure. It has been shown that the CIT2 FLS with the novel algorithm keeps the same level of accuracy as its IT2 counterpart while producing outputs with a higher level of interpretability (for each of the AES it is possible to determine which rules and input values generated them).

In future work, it will be studied how to further decrease the run time of Algorithm 4. In fact, the identification of the switch indices, for now, has been carried out using a brute force approach. Determining a different stopping criterion or a direct way to identify the switch indices (similarly to what happens with the switch points in the KM procedure) would further improve the computational complexity of the novel procedure presented here. Finally, the possible advantages and differences in the use of the constrained modeling approaches in systems like Takagi-Sugeno [69] will be studied.

Chapter 5

Constrained Interval Type-2 Fuzzy Systems in Explainable Classification Tasks

5.1 Introduction

In this chapter, the CIT2 defuzzification algorithm proposed in Chapter 4 will be used to design CIT2 FLSs that provide explanations for each of their classifications. For both endpoints of the interval centroid, the AES, the rules and the input variables that contributed to their creation will be identified, adding valuable information for the understanding of the internal decision process of the system.

The rest of the chapter is organized as follows: after a brief introduction on CIT2 fuzzy sets and the reasons why they were introduced, the creations of the explanations for CIT2 FLSs will be discussed; this approach will then be applied to two case-studies in the medical domain, showing how the explanations can be obtained and the level of information they are able to provide while briefly discussing why the same level of explanation is harder to achieve with the standard IT2 representation.

5.2 Explainable Constrained Interval Type-2 Fuzzy Systems

This subsection shows how the mathematical restrictions of CIT2 fuzzy sets, together with the inference and defuzzification approach in Chapter 4, can be used to design CIT2 FLSs that are able to provide explanations for each of the output centroids they produce.

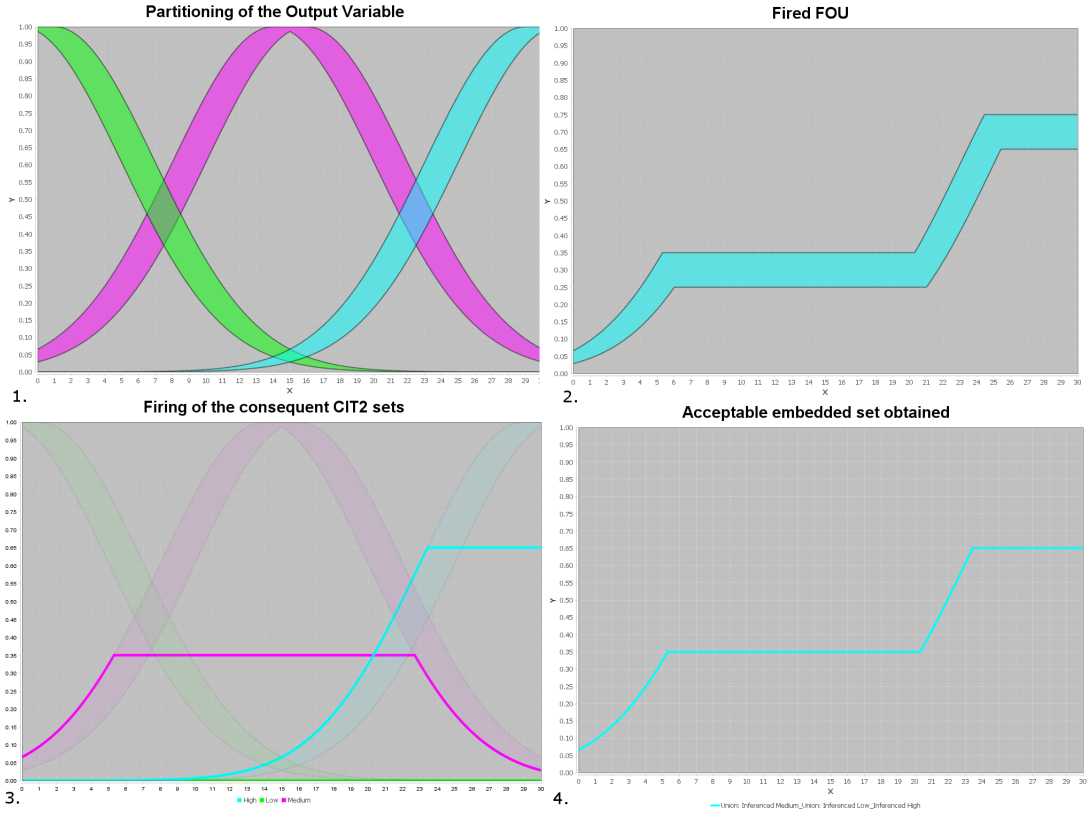


Figure 5.1: *Creation of the AES of the fired output (2.) that determines the left endpoint of the constrained centroid. First the partitioning of the output variable (1.) is shown, then for each consequent MF one AES is selected and inferenced (3.). Finally, the inferenced sets are aggregated to produce the final AES (4.).*

In summary, the algorithm selects the two AES to determine the endpoints of the interval centroid of a CIT2 FLS. Each of the AESs is generated as the aggregation (by the use of the *or* operator) of all the MFs that appear as consequents in the rule-base. Each CIT2 consequent is replaced with one of its AESs (more on this below) and then one of the endpoints of the firing interval of the rule they belong to is used to carry out the inference. The latter choice

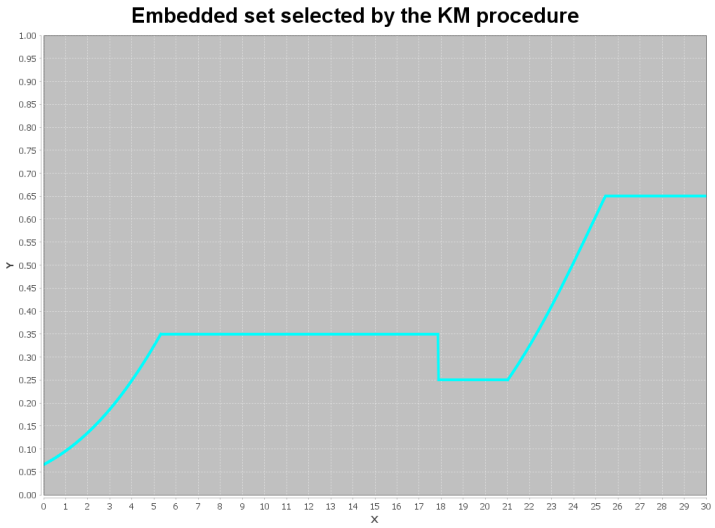


Figure 5.2: *The ES determining the left endpoint of the centroid of the same set as that shown in Fig. 5.1.2 using the KM procedure*

depends on the index value assigned to the consequent MF and on the *switch index* value that has been chosen by the algorithm. By noting the rules and the firing value used for the inference on each consequent MF, it is possible to build an explanation for the final output.

The algorithm can be briefly summarized in the following steps:

1. Give each CIT2 consequent MF an ordinal index by sorting them in ascending order of the minimum value of their support set.

2. For each CIT2 consequent set, compute its firing interval as the maximum lower and maximum upper values of the firing strengths of all the rules where it appears as a consequent.

3. If computing the right endpoint of the constrained centroid (i.e. to generate the AES with the maximum centroid value), replace each consequent MF with its rightmost AES; if computing the left endpoint, take the leftmost AES instead.

4. Test all the possible switch index values, between 0 and the maximum index given to the consequent MFs:

    i. If computing the left endpoint, use the upper value of the firing

interval to utilise the MFs with an index smaller than the switch index and *switch* to the lower value afterwards; for the right endpoint instead, use the lower value of the firing interval before the switch index and the upper one after it.

ii. Do the union of the AES resulting from the inference and defuzzify the set obtained.

5. Return, as the final constrained centroid, the lowest and highest centroid values obtained from the defuzzification at the previous step.

The process that leads to the creation of one of the acceptable embedded sets that determine the constrained centroid is also shown in Fig. 5.1. It is straight-forward to see that the AES has been obtained as the union of two MFs (*medium* and *high*); additionally, the respective firing strengths of the rules that were used are also identifiable (i.e. the 'truncation heights' in Fig. 5.1.2), producing an easily interpretable AES. Once each consequent MF is replaced with one of its AESs (the leftmost or rightmost one) and for each one of them an inference value is chosen (i.e. one of the endpoints of the firing interval), all the operations are carried out using T1 mathematics. For this reason, as can also been seen in the example in Fig. 5.1, the AESs that determine the endpoints of the constrained centroid keep the same level of interpretability as any fuzzy output of a T1 FLS. In other words, while CIT2 FLSs allow for the modeling of uncertainty around the membership function (making use of the FOU) they also keep the same level of interpretability as T1 FLSs. On the other hand, the IT2 modeling struggles to achieve the same properties. The lower ES chosen by the KM procedure to defuzzify the same output set as that shown in Fig. 5.1.2 is shown in Fig. 5.2. Compared to the one selected by the constrained approach (Fig. 5.1.4), it is harder to identify how the consequent MFs contributed to its creation and it can be challenging to link it to the rules of the system and their firings (see Sec. 4). This is because the KM procedure selects the two ES that solve a well-defined mathematical

problem but that do not necessarily carry a semantic meaning.

Furthermore, as will be demonstrated in the next subsection and in the case studies in Sec. 5.5, these properties of CIT2 FLSs can be used to produce a human-readable explanation for each output of the system.

5.3 Generation of the explanation

In the examples provided in this chapter, the explanations for the classification systems are divided into two parts: first the predicted class is presented, together with the interval centroid that generated it; then, for both endpoints of the centroid, the AESs, the rules and firing values that produced them are shown. Each rule has a different consequent MF, showing the firing strength for each of the possible classes.

The interpretable AESs provided give an intuitive idea of the firings of each class while the description with the rules that fired gives a more detailed and accurate description of the decision process followed by the FLS. The creation processes of the AESs themselves are illustrated: for each consequent MF in the rulebase one AES is chosen and inferenced using one of the endpoints of the firing interval; the union of all the inferenced sets gives the AES of the fired FOU of the rulebase.

While similar explanations have already been produced for T1 FLS before (e.g. [11, 12]), they represent a novelty in the T2 field. In fact, producing explanations for IT2 and T2 FLS outputs has been very challenging since to compute the left and right endpoints of the interval centroid, all the embedded sets are processed regardless of their shape. As a consequence of that, the embedded sets that determine the endpoints of the interval centroid in the standard IT2 approach do not carry any particular meaning (making them harder to interpret), nor do they have a direct link with any of the rules of the rulebase (making the generation of an explanation less straightforward).

At this stage, there is no data gathered from users (e.g. with surveys) that

determine the usefulness of the explanations of CIT2 FLS compared with IT2 ones. The superior explainability claimed in this chapter is therefore based on the *ability* of CIT2 FLS to produce explanations for their classifications rather than on the users' feedback. Future work will focus on validating these claims by the use of surveys in which both approaches are compared in order to understand if the additional information provided by CIT2 FLSs is perceived as useful by domain experts.

5.4 Juzzy Constrained: a CIT2 software library

To facilitate the use by the research community of CIT2 FSs and to make the use of CIT2 FLSs possible in practice, a CIT2 software library has been produced. It implements CIT2 FSs and FLSs with all the algorithms introduced in Chapter 3 and in Chapter 4. The library is called *Juzzy Constrained*, has been developed in Java and it is open-source and freely available on GitHub and Maven[1]. It has been designed as an extension of the popular T1 and T2 *Juzzy* [64] and follows its conventions to facilitate its use for developers. The new toolkit is capable of using the constrained representation to provide human-readable explanations for the constrained interval centroids produced by the systems. The library has been fully described in a paper presented at the *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* [67]; a detailed description of the software library and examples showing how to use it can be found in the Appendix of this thesis (Chapter A). All the practical applications in the rest of this chapter, have been implemented using *Juzzy Constrained*.

[1] https://github.com/PasqualeDAlterio/JuzzyConstrained

5.5 Case Studies

In this section, two case studies taken from the medical domain are analyzed. The goal is to demonstrate that the use of CIT2 FLS can be beneficial in situations in which it is important to understand the decision process behind the system classification to detect possible inconsistent decisions and/or to guarantee a fair treatment. At the same time it will be shown that, in these examples, both CIT2 and IT2 FLS achieve the same level of accuracy.

```
The predicted class is MAYBE, from the midpoint of the output [49.16, 52.2]
The leftmost centroid (49.16) is obtained from firing the following rules:
    1. Chemo_no: 0.6, obtained because NPI IS High [1, 1] AND ER IS
        Not_Negative [1, 1] AND age IS
        Old [0.5, 0.6], using the upper membership degree of each input term
    2. Chemo_maybe: 1, obtained because NPI IS High [1, 1] AND ER IS
        Not_Negative [1, 1], using
        the upper membership degree of each input term
    3. Chemo_yes: 0.56, obtained because NPI IS High [1, 1] AND ER IS Weak
        [0.56, 0.61], using the lower
        membership degree of each input term
Aggregating these output terms produces the embedded set shown in Fig. 5.4,
    with the centroid 49.16:

The rightmost centroid (52.2) is obtained from firing the following rules:
    1.Chemo_No: 0.5, obtained because NPI IS High [1, 1] AND ER IS
        Not_Negative [1, 1] AND age IS Old [0.5, 0.6], using the lower
        membership degree of each input term
    2.Chemo_Maybe: 1, obtained because NPI IS High [1, 1] AND ER IS
        Not_Negative [1, 1], using the lower membership degree of each input
        term
    3.Chemo_Yes: 0.611, obtained because NPI IS High [1, 1] AND ER IS Weak
        [0.56, 0.611], using the upper membership degree of each input term
Aggregating these output terms produces the embedded set shown in Fig. 5.5,
    with the centroid 52.2:
```

Figure 5.3: *Example of explanation of the output for the classification of the post-operative breast cancer treatment CIT2 FLS.*

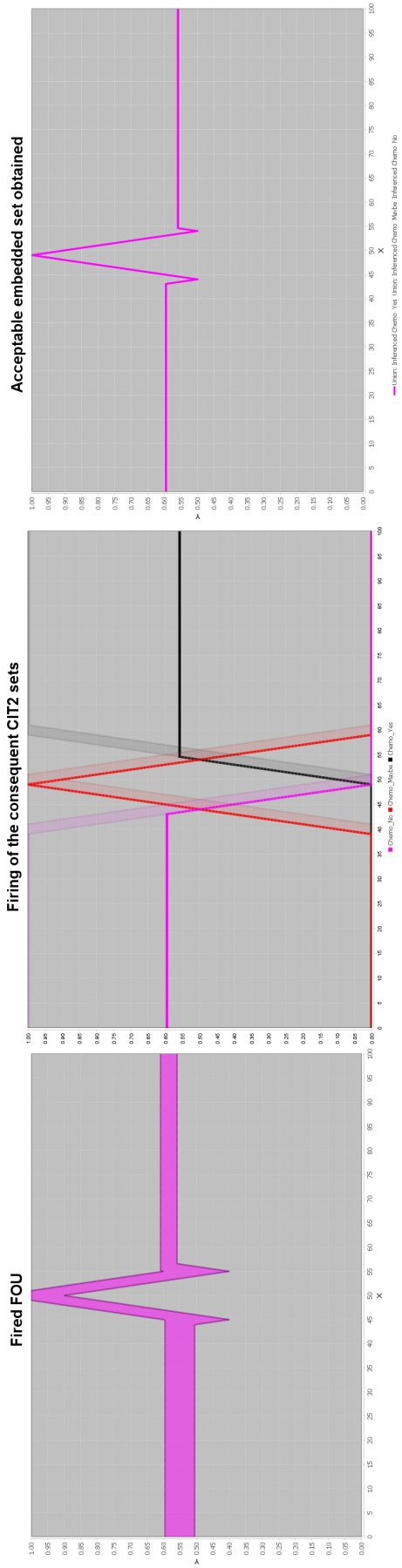


Figure 5.4: *Graphical representation of the process to obtain the AES with the leftmost centroid in the thyroid example in Fig.5.3*

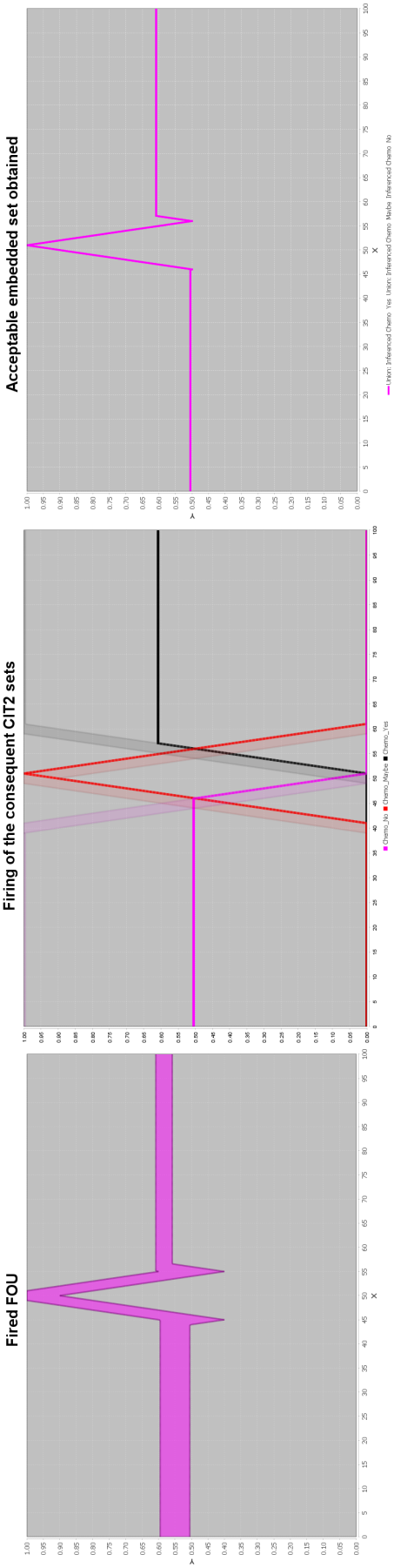


Figure 5.5: *Graphical representation of the process to obtain the AES with the rightmost centroid in the thyroid example in Fig.5.3*

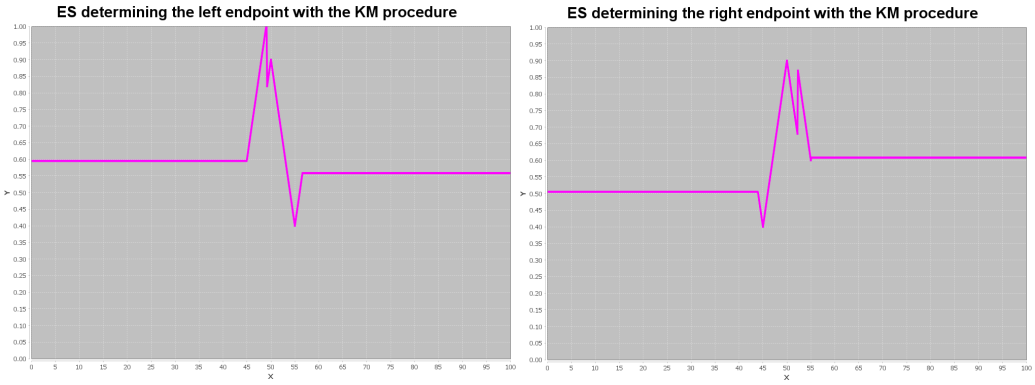


Figure 5.6: *Embedded sets selected by the KM procedure to defuzzify the fired FOU in Fig. 5.3*

5.5.1 Recommendation of post-operative chemotherapy for breast cancer

The first classification system presented here concerns the recommendation of post-operative chemotherapy for breast cancer. After the surgery to remove the tumor, a team of physicians makes a recommendation for the best additional therapy for the patient. In this case, the goal of the system is to replicate the decision process of the group of physicians with respect to the recommendation of chemotherapy. The three possible outcomes are *yes*, *no* and *maybe* with the first two cases denoting a decision in favor or against the use of chemotherapy and the latter represents the scenario in which a clear recommendation cannot be provided (e.g. because there is not an agreement among the physicians) and the post-operative therapy needs to be further discussed with the patient. The problem has already been analyzed by Garibaldi et al. [41], whereby different T1 and non-stationary [39] fuzzy systems have been designed and compared. The CIT2 FLS proposed in this chapter, is based on the T1 FLS denoted as VI-F previously [41]. Its T1 MFs are used as generator sets for the corresponding CIT2 MFs; the displacement set $[-a, a]$ (i.e. the "shifting interval" used to obtain the FOU and the acceptable embedded sets) has been experimentally chosen so that for each MF $|2a| = 2\%$ of the size of the universe of discourse.

The rule-base, as previously [41], is based on a written protocol provided

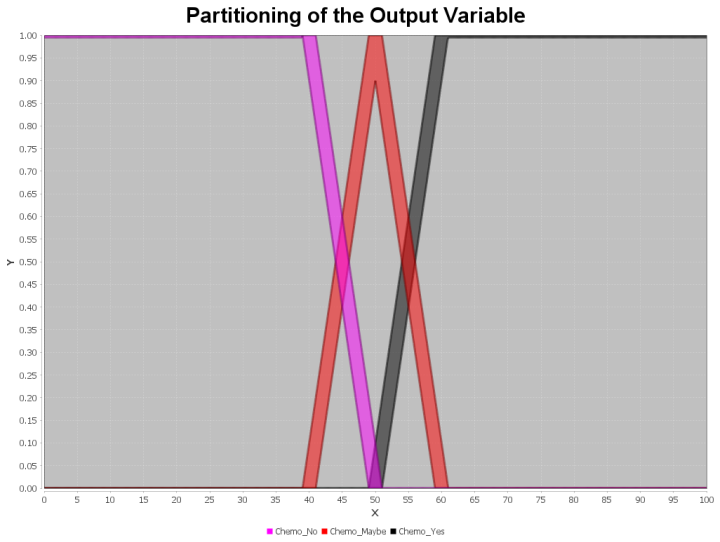


Figure 5.7: *Partitioning of the* chemo recommendation *variable. The FS, from left to right, model the words* no, maybe *and* yes

by the Nottingham University Hospitals Trust, in order to assure a high level of interpretability. Additionally, each of the MFs used in the system models a word, such as *negative, positive, high, low* and *medium*. Fig. 5.3 shows an explanation provided for a case that has been classified as *maybe*, in which the output variable *chemo recommendation* is partitioned as shown in Fig. 5.7.

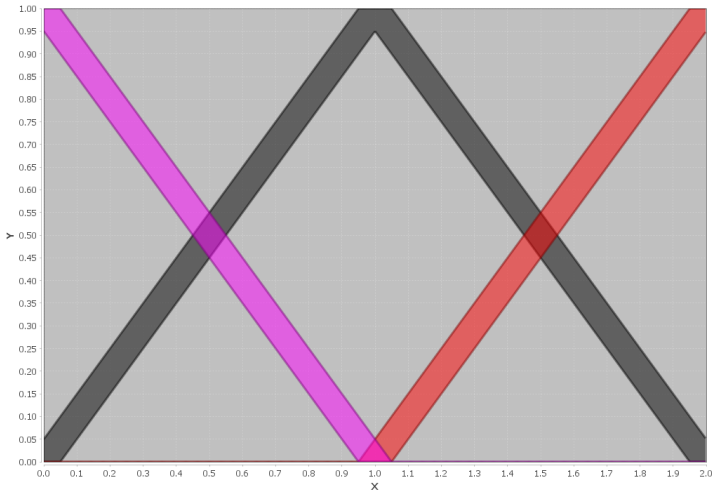


Figure 5.8: *Partitioning used for each of the variable in the thyroid CIT2 FLS*

Using the KM procedure to defuzzify the same FLS output, results in end-points determined by the ESs shown in Fig. 5.6. Since CIT2 fuzzy sets are a subset of IT2 sets, the inferencing can also be carried out using the standard IT2 approach. When using the midpoint of the centroid to perform the classi-

fication, both the CIT2 and IT2 methodologies (using the KM defuzzification procedure) have an accuracy of 72.29% when tested on the same dataset used in [70].

```
The predicted class is Hyperthyroidism, from the midpoint of the output [1.59,
    1.66]
The leftmost centroid (1.59) is obtained from firing the following rules:
    1. Hyperthyroidis: 0.6, obtained because T3resin IS Medium [0.53, 0.63]
        AND Thyroxin
        IS Medium [0.66, 0.76] AND Triiodinthyronine IS Medium [0.91, 1] AND
            TSH_value IS Low [0.92, 1]
        using the lower membership degree for each input term
Aggregating these output terms produces the embedded set shown in Fig. 5.10,
    with the centroid 1.59:


The rightmost centroid (1.66) is obtained from firing the following rules:
    1. Hyperthyroidism: 0.66, obtained because T3resin IS Medium [0.53, 0.63]
        AND Thyroxin
        IS Medium [0.66, 0.76] AND Triiodinthyronine IS Medium [0.91, 1] AND
            TSH_value IS Low [0.92, 1]
        using the upper membership degree for each input term
Aggregating these output terms produces the embedded set shown in Fig. 5.11,
    with the centroid 1.66:
```

Figure 5.9: *Example of explanation of the output for the classification of thyroidal disease CIT2 FLS*

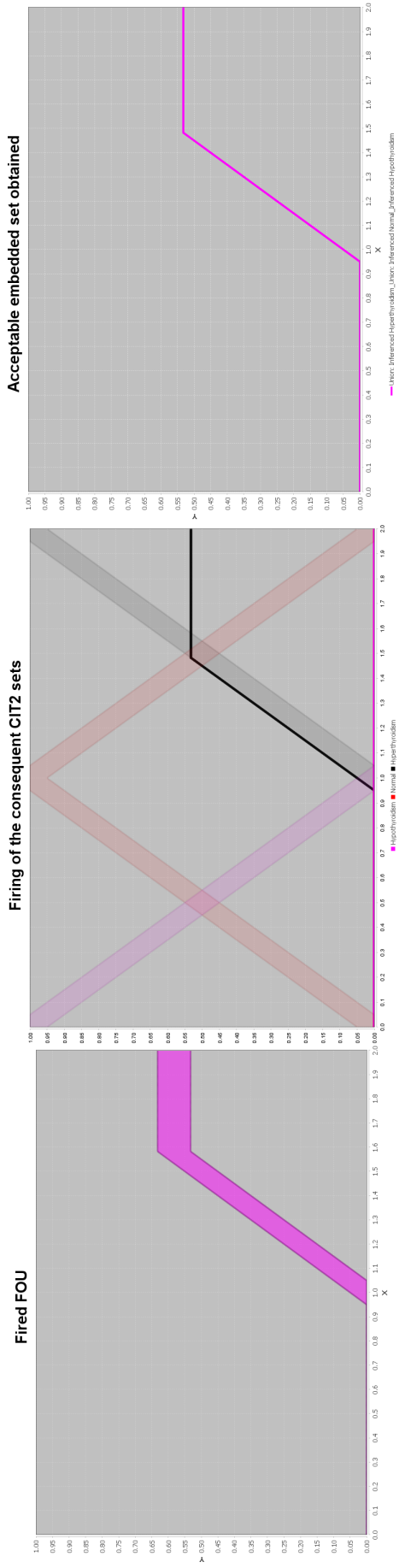


Figure 5.10: *Graphical representation of the process to obtain the AES with the leftmost centroid in the thyroid example in Fig.5.9*

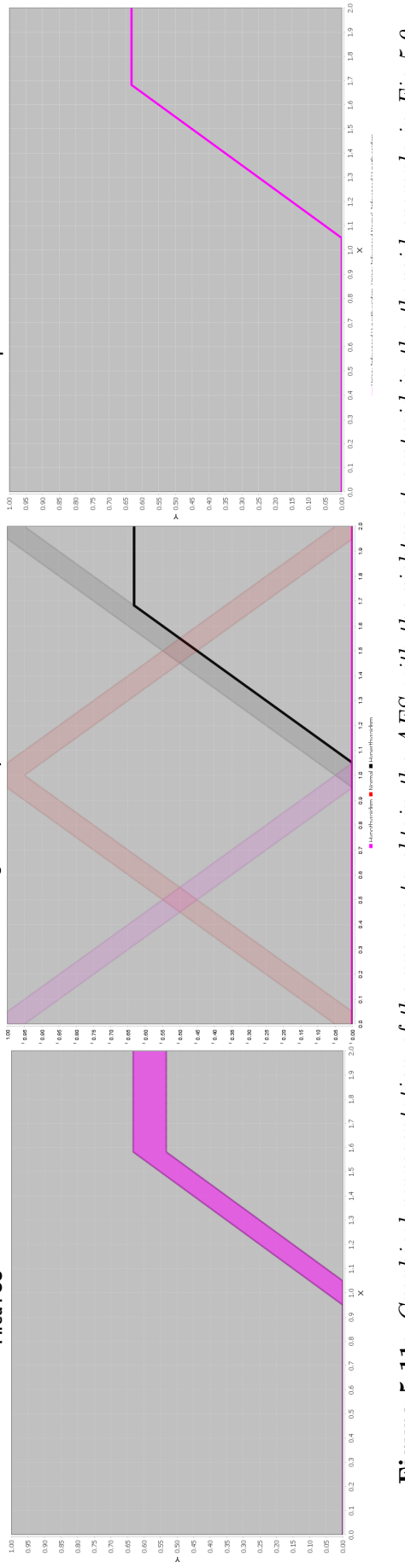


Figure 5.11: *Graphical representation of the process to obtain the AES with the rightmost centroid in the thyroid example in Fig.5.9*

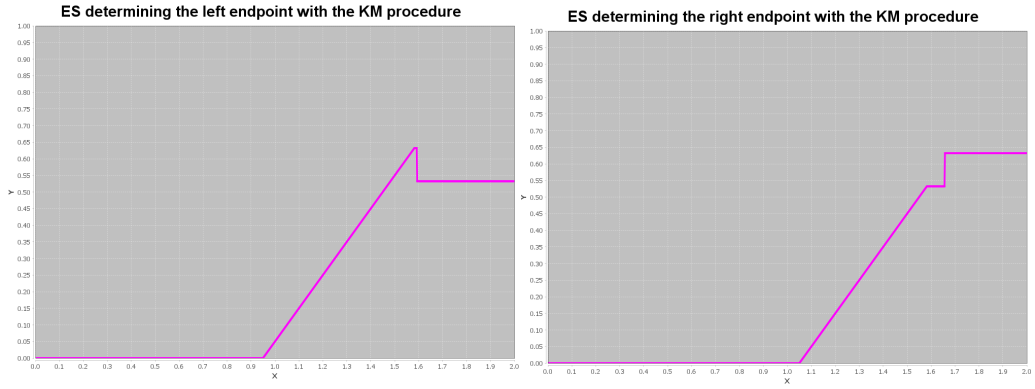


Figure 5.12: *Embedded sets selected by the KM to defuzzify the fired FOU shown in Fig. 5.9*

5.5.2 Thyroid disease diagnosis

In this case study the aim of the system is to predict whether a patient suffers from a thyroid disease (hypothyroidism or hyperthyroidism) on the basis of the analysis of some physiological data. For this system, there was no expert knowledge available from which it was possible to build the rule-base and the MFs. To build an interpretable FLS for this problem, each input variable has been partitioned with three MFs modeling the words *low, medium* and *high*, with the first and last one being implemented as triangular shoulders with their peaks being the endpoints of the universe of discourse, while the *medium* MF is as an isosceles triangle with its peak in the midpoint of the universe of discourse. The partitioning strategy described is shown in Fig. 5.8. The output variable is partitioned in the same way, with the 3 MFs representing respectively the terms *hypothyroidism, normal* and *hyperthyroidism*. The displacement set $[-a, a]$ (i.e. the "shifting interval" of the generator set to obtain the FOU and the acceptable embedded sets) has been experimentally chosen so that for each MF $|2a| = 5\%$ of the size of the universe of discourse.

For the rule-base, ten rules have been created using the same genetic approach described in Chapter 3 for the first stage of the optimization. Although this is one of many ways in which it is possible to generate a FLS from data, this method has been chosen with the only goal of generating a compact rule-base in which each MF identifies a meaningful linguistic label, to keep a high

level of interpretability [10]. The dataset used for the learning phase is the "newthyroid" dataset available on the KEEL website [63]. The accuracy of the system produced on this dataset is 88.37% using the KM defuzzification method and 88.84% for the CIT2 version. Fig. 5.9 shows the explanation produced by the CIT2 FLS for one of the entries of the dataset. In comparison, the ESs that determine the endpoint of the centroid for the same FLS fuzzy output using the KM procedure are shown in Fig. 5.12.

5.6 Discussion

In both the case studies provided, it has been shown how the previously proposed algorithm (Chapter 4) can be used to produce explainable CIT2 FLSs (Figs. 5.3, and 5.9). Each of the outputs, in addition to the predicted class, also provides the interval centroid from which it was determined and an explanation for its generation. Each endpoint is then accompanied by the AES that determined it. For each of these AES an explanation for their creation is also provided, showing which rules contributed, their firing strength and the membership degree of the input values. These explanations can provide valuable information to understand the decision process followed by system for the following reasons:

- The presence of the interval centroid shows the effect of the uncertainty on the final output. Intuitively a 'wider' centroid represents a more uncertain result.

- As it is possible to see in Figs. 5.3 and 5.9, the AESs keep the same level of interpretability of T1 fuzzy outputs, i.e. it is possible to recognize the different terms involved (the consequent MFs) and the firing strengths of the rules they belong to (their 'truncation' heights). This provides an intuitive idea of how the constrained centroid has been obtained.

- Lastly, illustrating the rules that generated each of the AES and the

membership degrees of the antecedent terms, provides a more technical and detailed explanation for the final output of the system.

The last 2 points described above represent a novelty in the IT2 field. In fact, modern algorithms like the KM [18] one and its enhanced versions are nowadays considered the standard for the defuzzification of IT2 FSs. They work by quickly identifying the embedded sets with the lowest and highest centroid value to compute the interval centroid of a set. However, although these embedded sets are mathematically acceptable and solutions to a well-defined optimization problem, their shapes may not carry any particular meaning *in specific contexts*. That is because all the embedded sets are processed, regardless of their shape. Consequently, giving a semantic meaning to the embedded sets determined by the KM procedure may be challenging. These claims are supported by the comparison between the embedded set chosen by the KM procedure in Figs. 5.6 and 5.12, and those produced by the constrained approach, in the explanations in Figs. 5.3 and 5.9, respectively. While the constrained embedded sets have the same level of interpretability of a T1 FLS output in which the different MFs and firing strengths are clearly identifiable, the same can not be said for the embedded sets of the KM approach. Particularly, due to the presence of the *switch point* (that is crucial for the identification of these embedded sets), the shape of the original MFs are partly lost and it is challenging to determine a direct relation between the rules of the FLS and the generation of such shapes. Therefore, building an explanation similar to the one offered by CIT2 FLS would not be straightforward.

The properties of CIT2 FLSs and the level of detailed shown in the explanations presented in the case studies, make CIT2 a valid and attractive *alternative* to IT2 FLS, in any context in which the interpretability of the system *and* a degree of explainability of the output is required.

5.7 Summary

In this chapter it has been described how the defuzzification algorithm presented in Chapter 4 can be used to design explainable CIT2 classification systems in which explanations can be provided for each of the classes predicted. In addition, it has been shown that the embedded sets processed by the CIT2 approach have a higher level of interpretability since they are built in a way that makes the identification of the linguistic terms and the firing strengths easier (see Sec. 5.6).

To support these claims, two case studies have been analyzed, both belonging to the medical domain: the selection of post-operative therapy for breast cancer and the thyroidal disease treatment problem. In both tasks the goal of the system was to analyze some physiological data belonging to the patient in order to make a therapy recommendation or a medical decision. The CIT2 approach has been compared to the standard IT2 one, showing that CIT2 FLSs are able to produce detailed explanations for the system outputs while having similar performances in terms of the accuracy of the classification. For each classification produced, the rules involved and the firing strengths used for each of the endpoints of the centroid have be shown, providing valuable information for the understanding of the decision process of the system.

In future work, statistical data from surveys will be gathered to explore whether the explanations provided by CIT2 FLS are perceived as more interpretable than the IT2 ones by end-users and experts. Furthermore, the information in the explanations will be reorganized in order to generate a more coherent piece of text in natural language, similarly to what has been done for T1 FLSs in other work [11, 12]. Additional work is also needed to understand how interpretable the CIT2 explanations are for the end users compared to the ones produced by T1 systems.

The next chapter, will analyze the concept of meaningfulness in the context of CIT2 fuzzy sets, examining the limitations of the current CIT2 definitions.

In fact, in many scenarios the meaningfulness of a concept is not strictly related to a specific shape but rather to a set of properties that need to be satisfied. In the next chapter, this idea will be used to create a more flexible definition of CIT2 FSs that considers as acceptable the embedded set satisfying a set of context-dependant properties rather than the ones having fixed shape.

Chapter 6

Refining The Concept of Meaningfulness in Constrained Interval Type-2 Fuzzy Sets

6.1 Introduction

Although CIT2 FSs heavily rely on the concept of "meaningfulness" so far, no clear definition has been given of what a "meaningful" shape for a membership function is. At the same time, it has been shown in Chapter 3 that operations on CIT2 FSs may produce IT2 FSs that formally are not CIT2 FSs, i.e. it is not possible to find a generator set (GS) that would generate them. That is because all the AESs obtained from the fuzzy operators have different shapes, regardless of the fact that they could all be reasonable for the operation result they represent. The aim of this chapter is to both clarify the concept of "meaningfulness" in CIT2 FSs and to extend the original CIT2 definitions, in order to include different and more general constraints that go behind the requirement of having ESs with the same shape. By doing this, a more powerful modeling tool will be provided and that can be useful in all the cases where different shapes (e.g. both triangular and Gaussian) are considered acceptable for the representation of a given concept.

6.2 The concept of meaningfulness and CIT2 fuzzy sets: two case studies

Even though the idea behind CT2 FSs was to provide a representation that keeps a "meaningful" relation between T2 FSs and the concept they model, there are some cases in which the restriction of having only AESs sharing the same shape is too limiting. This claim will be supported by providing two practical example of CIT2 application in which the use of different shapes is needed to obtain an accurate representation of the modeled scenario.

6.2.1 Modeling Words

In this thought experiment, the goal is to obtain a CIT2 FS for the concept of medium height.

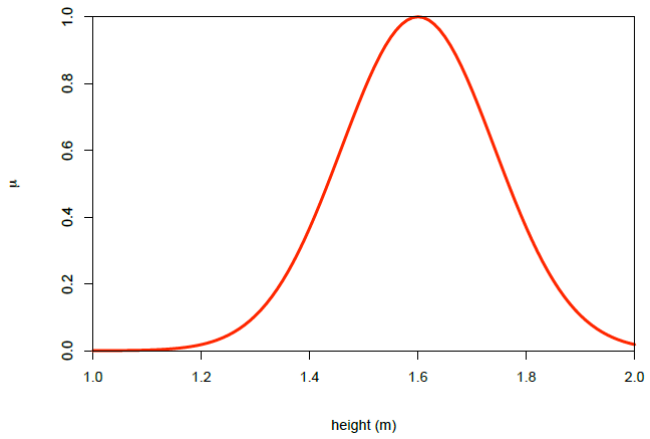


Figure 6.1: *T1 GS modeling medium height (picture from [1])*

In order to obtain the CAES, different people are asked to place a T1 Gaussian FS like the one in Fig. 6.1, on the x-axis (similar approaches can be found in [55, 56]). Since the concept of medium height varies slightly from person to person, it is likely that something similar to what is shown in Fig. 6.2 would be obtained from the experiment.

Using the approach described above, ensures that only the ESs with a "meaningful" shape are included in the AES and then processed by fuzzy operators such as centroid defuzzification. Specifically, all the AES keep a

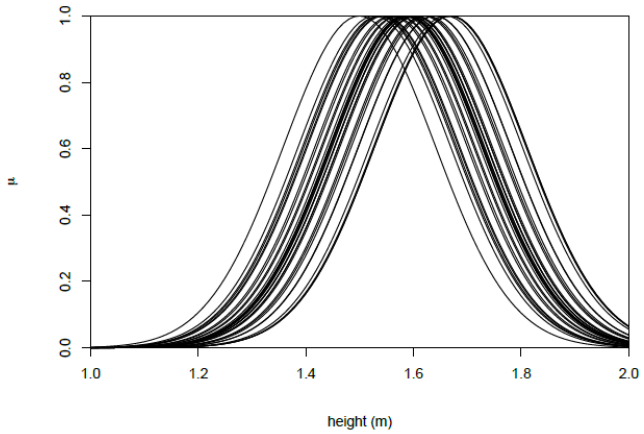


Figure 6.2: *AES obtained from the medium height experiment (picture from [1])*

semantic relation with the concept of medium height they are modeling.

6.2.2 Analysis - I

The idea of imposing the use of one specific Gaussian for the generation of the CAES is very limiting in this case. For example, one could imagine that some of the participants would want to change the spread of the Gaussian or would want to use triangular shapes instead of Gaussian ones. This would be unacceptable by the current CIT2 definitions since a CAES with different shapes would not satisfy Def. 3.2. Nevertheless, in this example there are multiple shapes that can be considered "meaningful", in the sense that they keep the semantic relation with the concept they model. In this case, it can be seen that the concept of "meaningfulness" is not kept by one specific shape but it is rather the result of the satisfaction of a set of constraints that are implicitly imposed on words in human reasoning. For example, one could imagine that in the case of "medium height", the meaningfulness and the semantic relation is kept by all the symmetric shapes that are monotonically increasing up to a plateau and then monotonically decreasing.

The analysis of this experiment suggests that the idea of imposing one shape to all the ESs is only *one* of the possible constraints that a designer would want to use for a T2 FS and that the concept of "meaningfulness" is not related to one specific shape but is rather the result of the satisfaction

of a set of implicit constraints that are related to the concept one is working with. Furthermore, even convexity and normality, which are usually described as "desirable properties" for MFs [1, 21, 52], can be "non-meaningful" in some contexts, as shown in the next subsection.

6.2.3 Fuzzy system outputs: a non-normal and non-convex case

To show that non-normal and/or non-convex MFs can still be meaningful, the following problem will be analyzed. Consider the CIT2 fuzzy rule R:

$$R\text{: IF } x_1 \text{ is } \breve{A} \text{ AND } x_2 \text{ is } \breve{B} \text{ THEN } y \text{ is } \breve{C}$$

The consequent FS $\breve{C}$ is shown in Fig. 6.3. It is obtained using a triangular MF as a GS and a discrete DS to generate the AESs.

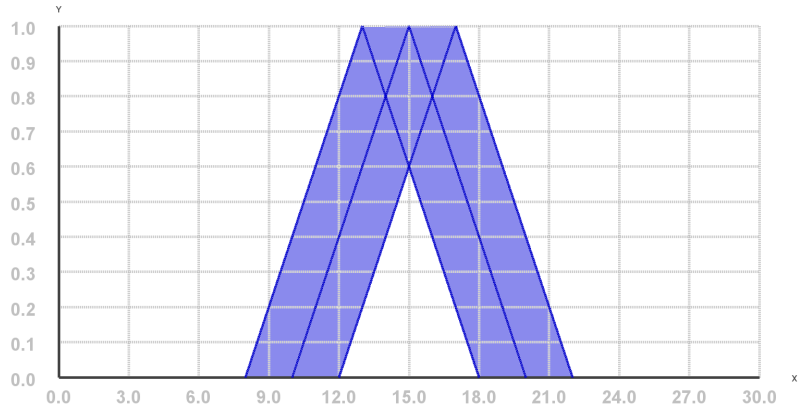


Figure 6.3: *Consequent CIT2 FS $\breve{C}$ used in the rule R (FOU in light blue)*

To carry out the inference, the CIT2 fuzzy rule is expanded in a set of T1 fuzzy rules; each one of them is obtained by substituting the CIT2 FSs involved in the rule with one of their T1 AESs. The goal of the process is to obtain the AESs of the FS resulting from the rule evaluation.

The fuzzy rule output shown in Fig. 6.4 has been obtained using the process described above, with the minimum function for the conjunction (and) and implication operator.

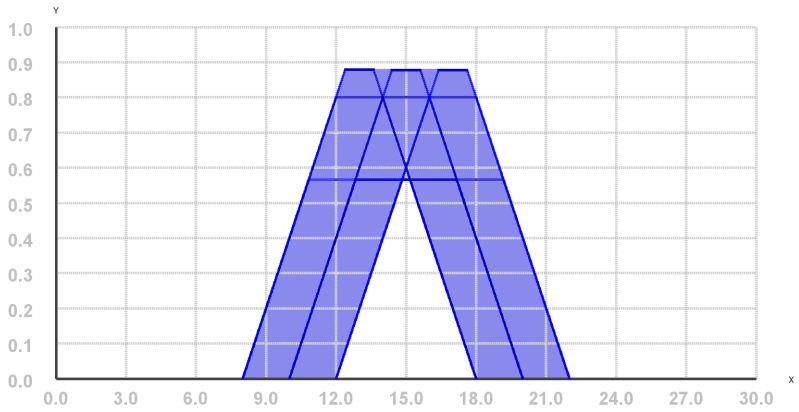


Figure 6.4: *CIT2 output from the inference of a CIT2 rule in which all the sets involved are fixed-shape CIT2 sets (FOU in light blue)*

6.2.4 Analysis - II

It is clear that the collection of T1 ES in Fig.6.4 is not a CAES as defined in Def. 3.2: since they have been obtained from the same triangular shape truncated at different height, it is not possible to identify a valid T1 GS. In other words, it is not possible to choose one of the T1 FS in Fig. 6.4 as a GS, so that the other AESs could be obtained from the translation along the x-axis of the GS. Furthermore, these AESs are non-normal. Therefore, it can be concluded that the fuzzy result of a CIT2 fuzzy rule is not a CIT2 FS. This seems to suggest that the collection of T1 FSs in Fig. 6.4 is not "meaningful" in this context.

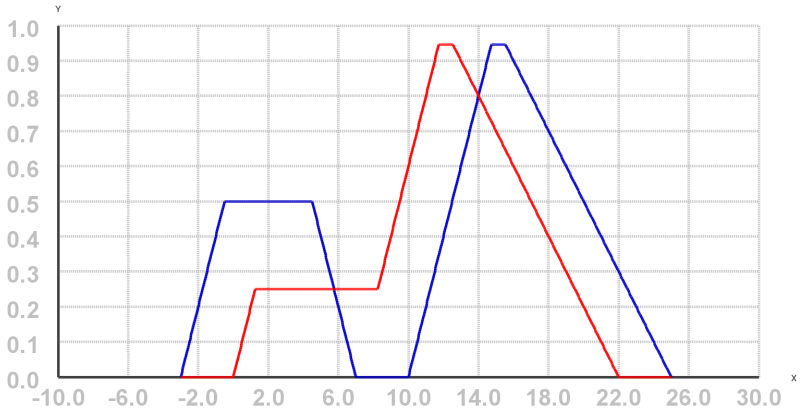


Figure 6.5: *Examples of two AES obtainable from a CIT2 Mamdani fuzzy system*

However, each of those T1 FSs represents a plausible T1 fuzzy rule output since they have been obtained as results of T1 fuzzy rules by picking one of

the AESs of each CIT2 FSs involved in the CIT2 rule. Intuitively, the T1 FSs in Fig. 6.4 represent possible T1 fuzzy rule outputs when the uncertainty modeled around the T1 GS is removed and the CIT2 FS collapses to one of its AESs, i.e. when the one of the *possible* locations of each T1 GS on the x-axis is chosen. Therefore, the collection of T1 FSs in Fig. 6.4 represents all the possible T1 fuzzy outputs that can be obtained by taking into account the uncertainty on the T1 GSs of all the CIT2 FSs involved in the rule.

This analysis supports the fact that the FSs in Fig. 6.4, still carry a "meaningful" connection when it comes to the representation of fuzzy rule outputs even though they do not satisfy the definition of CAES as described in Def. 3.2 and are non-normal T1 FSs. In contrast, the standard IT2 representation would consider as acceptable all the ESs of a given IT2 fuzzy rule output, regardless of the fact that they could or could not represent an actual T1 rule output FS.

In addition to that, in Mamdani fuzzy systems where multiple CIT2 fuzzy rule outputs are combined by the union operator, a collection of T1 FSs which is non-convex (Fig. 6.5) will likely be produced. For the same reasons discussed above, however, those FSs would still be "meaningful" for the representation of a Mamdani system output since they represent plausible T1 system outputs when an exact location for all the GSs is chosen. From these plausible T1 system outputs, additionally, it is possible to extract meaningful information about the firing rules and their firing strengths. It can therefore be concluded that even non-convexity and non-normality *can* be meaningful in some specific contexts.

6.3 Extending Constrained Type 2 Fuzzy Sets

As a result of the analysis carried out in the previous section, new formal definitions for CIT2 FSs are needed. As already discussed, the original concept of "meaningfulness" fulfilled by the use of a single shape for all the ESs can

be limiting in some contexts. Specifically, it is only useful when the kind of uncertainty modelled is restricted to the exact location of the T1 GS on the x-axis. In addition to that, it is not clear when and why a shape is considered to be meaningful in a given scenario. For these reasons, a different formalization of the concept of meaningfulness might help tackle these issues. Specifically, here is proposed a novel representation of the CAES based on the satisfaction of a set of constraints. This approach both formalises the concept of "meaningfulness" into the satisfaction of constraints and provides a representation that makes ESs with different shapes acceptable.

Definition 6.1. *A collection of T1 acceptable embedded sets (CAES), is a set of T1 FSs satisfying a set of n constraints constraints* $C_1, ..., C_n$*:*

$$CAES = \{S \,|\, \mu_S : X \mapsto [0,1], C_1(S) \star ... \star C_n(S)\} \tag{6.1}$$

with X being the UOD and each of the $\star$ *being either* $\wedge$ *or* $\vee$*.*

This new definition of CAES can then be used in Def. 3.3 to obtain new CIT2 FSs. All the other definitions remain unchanged.

In the context of human reasoning, those constraints are implicitly imposed by people on the words they use. For example, as discussed in Sec. 6.2, when using words such as *medium*, one can expect the MF modeling this concept to be monotonically increasing-decreasing, symmetric and convex. In other scenarios, MFs are obtained from data analysis, as in [20]. In this case, the constraints are given by empirical or theoretical relations between the values of the universe of discourse. The original idea of constraining the ESs to share the same shape is only *one* of the possible constraints one may want to impose on the T2 FS modelled.

To prove that this new formulation is more general than the old one, it will be shown that any CAES that satisfies Def. 3.2, can also be obtained by the use of constrains as in Def. 6.1. Specifically, given a CAES A where all its T1

sets are obtained from a T1 GS G and a DS D, A can also be expressed as:

$$A = \{S \,|\, \mu_S : X \mapsto [0,1], C_1(S)\} \tag{6.2}$$

where:

$$C_1(S) = \begin{cases} \textit{true} & \text{if } \exists\, c \in D : \mu_S(x) = \mu_G(x-c), \forall x \in X \\ \textit{false} & \text{otherwise} \end{cases}$$

6.4 Applications

To show a practical application of this new definition of CIT2, a case that is very similar to the one presented in the first part of Sec. 6.2 will be analyzed. Just like in the other thought experiment, the amin is to model a CIT2 FS representing medium height starting from T1 MFs obtained from a survey. The difference is that, this time, each person can freely choose the shape that he or she considers to be the most appropriate for this context. A possible experimental result is shown in Fig. 6.6. Since different MFs (e.g. triangular and Gaussian) would likely be obtained, this scenario could not be modeled with the old CIT2 definition. However, for example, both triangles and Gaussians are appropriate in this context.

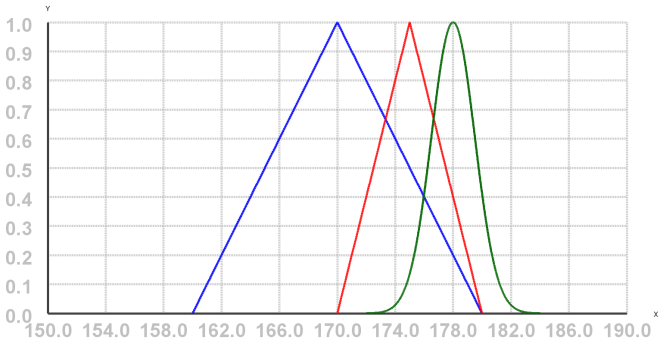


Figure 6.6: *Possible T1 MFs modeling medium height*

When the number of AES is finite and obtained from surveys or data analysis, generating the constraints for the CAES is trivial. One strategy would

be to put these MFs in a set named E and then define the following constraint C_1:

$$C_1(S) = \begin{cases} \mathit{true} & S \in E \\ \mathit{false} & \text{otherwise} \end{cases}$$

This constraint can then be used to build a CAES and a CIT2 FS as in Def. 3.3.

The idea of defining a CAES as T1 MFs satisfying a set of constraints is more powerful when the number of shapes that are acceptable is infinite. For example, one may want to consider as acceptable for medium height all the Gaussians having mean between 170 and 180 and having a standard deviation between 1 and 1.5. This scenario can be easily modeled by the following constraint:

$$C_G(S) = \begin{cases} \mathit{true} & \exists\, \mu, \sigma : 170 \leq \mu \leq 180,\ 1 \leq \sigma \leq 1.5 \\ & \mu_S(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}, \forall x \in X \\ \mathit{false} & \text{otherwise} \end{cases}$$

Even if all the MFs satisfying the constraints C_G are Gaussians, it would have not been possible to model this scenario using the old CIT2 definition. That is because the difference in their variance could not by modeled by Def. 3.2 since it considers only as acceptable Gaussians that differed for their mean, i.e. Gaussians that be obtained as the translation along the x-axis of a GS.

6.5 Discussion

The new CIT2 definition, based on constraints satisfaction, allows us to model a broader set of scenarios, like the ones described in Sec. 6.2, 6.3. In addition to that, no property is imposed a priori, not even normality or convexity.

This represent a significant difference when compared to the other constrained approach introduced by Wu in [21], where any "well shaped"[52] IT2

FS is represented using only normal and convex ESs.

In contrast, the imposition of normality and convexity can be too restrictive in some cases and not sufficient in others. For example, when working with fuzzy outputs these properties are usually not necessary since these sets keep their own interpretability regardless of their non-convex or non-normal shape.

If, instead, only one or a limited set of specific shapes are acceptable for the ESs, normality and convexity alone are not sufficient to guarantee that a T2 FSs will keep a semantic meaning with the concept modelled.

In addition to that, as already analyzed in [52], Wu's approach has the downside of being unusable in Mamdani systems since there is no guarantee that its fuzzy output will maintain the "well shaped" properties when rule outputs are combined using the union operator.

Furthermore, Wu's representation is a special case of the CIT2 definition proposed in this chapter. That is simply because convexity and normality, can be expressed mathematically in terms of constraints (as shown in [21]) that can then be used to generate a CAES and therefore a CIT2 FS.

Finally, it is important to mention that it is possible to build a CAES so that it includes all the ESs of an IT2 FS. In other words, given any IT2 FS, it is always possible to generate a CAES to obtain an equivalent CIT2 FS.

Specifically, given an IT2 FS $\tilde{A}$ with the FOU delimited by the upperbound and lowerbound MFs $\overline{\mu}_{\tilde{A}}$ and $\underline{\mu}_{\tilde{A}}$ it is possible to generate the $\text{CAES}_{\breve{A}}$ of the equivalent (i.e. with the same FOU) CIT2 FS $\breve{A}$ by using the conjunction of the two following constraints C_1 and C_2:

$$C_1(S) = \begin{cases} \mathit{true} & \text{if } \mu_S(x) <= \overline{\mu}_{\tilde{A}}(x), \forall x \in X \\ \mathit{false} & \text{otherwise} \end{cases}$$

$$C_2(S) = \begin{cases} \mathit{true} & \text{if } \mu_S(x) >= \underline{\mu}_{\tilde{A}}(x), \forall x \in X \\ \mathit{false} & \text{otherwise} \end{cases}$$

However, this does not mean that there is an equivalence between the IT2 and CIT2 representations. In fact, whenever CIT2 FSs are preferred to IT2 FSs, the goal is to work with a subset of all the ES described in the representation theorem [44], in order to keep a consisted semantic mapping between the concept modelled and the set.

6.6 Summary

In this chapter, the use of the concept of meaningful shapes in CT2 FSs was analyzed. It was shown how the current definition of CIT2 FSs that only considers as acceptable the ESs with a given shape is sometimes too strict, even in contexts such as human reasoning in which it is important to keep a connection between a concept and the FS that models it. In addition to that, the concept of meaningfulness itself has remained vague and not formally defined. To overcome these limitations, a new, more general definition of CIT2 FSs was given, based on the concept of the satisfaction of mathematical constraints to identify the shapes that are considered "meaningful" for the ESs. These constraints can be extracted, for example, by analyzing the properties that are implicitly associated with the word modelled or they can be determined to keep an empirical or theoretical relations between the values of the universe of discourse. It has also been shown how the old definition can be considered as a special case of the new one and how, given an IT2 FS, is it always possible to obtain its equivalent (i.e. with the same FOU) CIT2 representation by using two constraints. Finally, it was discussed the differences between the proposed approach and the constrained approach proposed by Wu in [21], which can be seen as a special case of CIT2 FSs. The next chapter, will show how this new CIT2 definition based on constraint satisfaction can be used to create a new way to model meaningful data with CIT2 FSs by preserving their semantic meaning through the different of multiple acceptable shapes.

Chapter 7

A Novel Method for Creating Interpretable Fuzzy Sets from Uncertain Data Using a Constraint-Based Representation

7.1 Introduction

Gathering data has become an increasingly important process as a necessary step to build models that perform automatic reasoning. Any data gathered from the real world, however, comes with some degree of uncertainty. When the data is collected from sensors, there are a number of errors to take into account, as well as the possibility of faulty hardware; when the data is obtained from human knowledge, instead, the uncertainty in the answers must be considered. It is therefore important to design approaches that preserve this uncertainty in the modelling phase. A very useful tool in this context is the framework of fuzzy sets and systems. Each individual instance (e.g. a single observation or sensor reading) can be represented through a fuzzy set, in which the uncertainty is

modelled through the membership function.

In some contexts, data on a given subject is collected multiple times from different sources. For example, different experts may be asked to give an opinion about a patient's condition or the same measurement may be obtained through different sensors. In this case, the fuzzy sets representing the individual instances can be combined together in a new type-1 or type-2 fuzzy set into a model that provides a more thorough representation of the data and its *variability* (e.g. how much the experts' answers or sensors' readings differed).

A typical example is represented by surveys: a group of people is asked to answer the same set of questions, often using a Likert [71] scale. The scale is made of values between 1 and 5 (with 1 usually meaning 'strongly disagree' and 5 'strongly agree') and each user can choose a single value per answer. This system, however, is not able to capture the uncertainty of the participants in the answers that they provide. For this reason, intervals have been used as an alternative as they allow the users to select a set of values, rather than a single one, when providing an answer. A wider interval represents a higher level of uncertainty, while a single value means zero uncertainty in the answer given [72]. Intervals can be directly modelled through type-1 fuzzy set, fully preserving the structure and information of the data.

In the literature, there are currently four main fuzzy approaches to combine group of intervals representing the same concept: the interval approach (IA) [73], the enhanced interval approach (EIA) [74], the interval agreement approach (IAA) [38] and the efficient interval agreement approach (EIAA) [75]. Each of these algorithms provides a different representation of the data to satisfy different needs. The IAA produces a non-parametric type-1 (T1) or type-2 (T2) fuzzy set (depending on the kind of uncertainty modelled) in which the membership degree represents the level of agreement, measured as the number of overlapping intervals, without discarding any of the collected data. The EIAA presents a more efficient version of the IAA in which each membership function is represented as the weighted sum of a set of basis functions. The

IA and EIA, instead, provide parametric interval type-2 (IT2) fuzzy set with a triangular or a shoulder shape. They are obtained by removing all the non-sensical opinions and outliers before using the statistics of the remaining data to generate the parameters of the membership functions. The goal in this case is to have a practical and noise-tolerant approach, at the cost of potentially discarding valid intervals and providing a partial representation of all the data collected.

None of the current approaches focus on preserving the shape used to model an individual instance during the aggregation process. However, the shape of a fuzzy set is important for its interpretability, as it is semantically linked to its underlying concept. In this chapter, a novel approach named *constrained parametric approach* (CPA) is proposed, to aggregate data instances modelled through parametric type-1 fuzzy sets in a way that preserves the shape used to model individual opinions, enhancing the interpretability of the produced models.

Specifically, the CPA makes use of constrained interval type-2 fuzzy sets (CIT2) [1, 65, 68] to guarantee that the chosen shape is preserved throughout the generation of the footprint of uncertainty and to ensure that its embedded sets only represent acceptable instances.

The approach has been applied to a case study involving combining interval-valued data gathered from surveys, and compared to the other approaches in the literature. Also, an example of application on data instances modelled through triangular fuzzy sets is illustrated, to show the flexibility of the novel approach and the difference in the shapes obtained compared to the interval-valued case.

The rest of the chapter is organized as follows. After a brief introduction to CIT2 fuzzy sets and the problem of modelling intervals with fuzzy sets, the CPA is described and applied to both synthetic and real-world interval-valued data. The experiments are followed by an extensive discussion, in which the different approaches are analyzed and compared, exploring the characteristics

of each. Lastly, it is shown how the CPA can be used to model data in which each instance is an opinion represented through a triangular fuzzy set.

7.2 Constrained interval type-2 fuzzy sets based on constraint satisfaction

Chapter 6 extended the CIT2 definitions in order to make them more general and make them usable in situations in which *multiple* shapes are considered acceptable for the modelling of a given concept.

In this new formulation, instead of forcing all the embedded sets to have a single, specific shape, the characteristics underpinning the meaningfulness of the modelled concept are modelled through mathematical constraints that need to be satisfied by the acceptable embedded sets. In the case of the word *medium*, for example, any membership function that monotonically increases and then monotonically decreases would be plausible. Therefore, Gaussian memberships with different standard deviations and triangular shapes with different parameters would all be acceptable. The original representation for CIT2 fuzzy sets, however, would not allow to have embedded sets with different shapes, as they could not be obtained by translating a generator set along the x-axis.

Formally, a set of constraints $C_0, ..., C_n$ can be used to build the collection of *acceptable* embedded sets (CAES) that characterizes a CIT2 fuzzy set $\breve{A}$ [3, 65]:

$$\text{CAES}_{\breve{A}} = \{S | X \mapsto [0,1], C_0(S) \wedge ... \wedge C_n(S)\} \tag{7.1}$$

where S is a T1 embedded set, X is the universe of discourse and $C_i(S)$, $i \leq i \leq n$ means that S satisfies the constraint C_i. All the constraints must be satisfied by an embedded set for it to be considered acceptable (hence the $\wedge$ in (7.14)). The CAES is then used to define the upper and lower membership functions of $\breve{A}$:

$$\overline{\mu}_{\breve{A}}(x) = \sup_{S \in \text{CAES}_{\breve{A}}} \mu_S(x) \tag{7.2}$$

$$\underline{\mu}_{\breve{A}}(x) = \inf_{S \in \text{CAES}_{\breve{A}}} \mu_S(x) \tag{7.3}$$

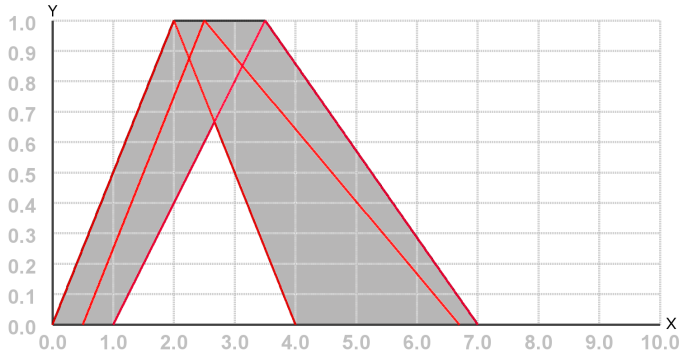


Figure 7.1: *A CIT2 fuzzy sets that makes use of the new representation [3] for acceptable embedded sets*

Fig. 7.1 shows an example of a CIT2 fuzzy set that makes use of the CIT2 definition based on constraint satisfaction. In this case, although a triangular shape has been defined as acceptable, having all the three embedded sets in red would not be possible with the original CIT2 representation, since they cannot be obtained by translating a single set (i.e. the generator set) along the x-axis.

In the rest of the chapter, it is shown how the CIT2 representation based on the satisfaction of constraints can be used to achieve two main goals: (i) maintaining a strong relation between the concept modelled and the generated CIT2 fuzzy set, by preserving the shape used to represent an individual instance (e.g. an interval, in the survey case); (ii) embedding into the footprint of uncertainty only the embedded sets that can actually model an individual instance, by considering as acceptable only the embedded sets with the same parametric shape used to represent individual instances.

7.3 Type-reduction and centroid defuzzification

Type-reduction is an important operation in T2 fuzzy logic as it maps a T2 fuzzy set into a T1 fuzzy set [17] and it is usually used before the defuzzification

process, that turns the type-reduced T1 fuzzy set into a crisp number.

Type-reducing an IT2 fuzzy set $\tilde{A}$ produces a T1 fuzzy set that is fully identified by an interval $[l, r]$, where l and r are respectively the lowest and highest centroid among all the embedded sets of $\tilde{A}$. All the embedded sets are considered at this step, regardless of their shape and it is likely that the two embedded sets determining l and r will not have a shape that could plausibly model an instance (e.g. a single opinion or an observation) in the data representation context. This phenomenon, reduces the interpretability of the defuzzification step and, therefore, of any system or model that uses them.

When type-reducing a CIT2 fuzzy set $\breve{A}$, on the other hand, only the *acceptable* embedded sets are processed. In this context, it means that only the embedded sets satisfying the constraints $C_0, ..., C_n$, contribute to the type-reduction and by extension to the defuzzified value of $\breve{A}$ [65].

7.4 Aggregating Interval-Valued Data with Fuzzy Sets

When modelling concepts, opinions and words with fuzzy sets, it is important to preserve the inherent properties and structure of the data by the use of an appropriate shape.

Interval-valued data can be directly mapped into fuzzy sets. In fact, an interval can be seen as a crisp set of numbers, i.e. a set in which a number either belongs or does not belong to the interval. For example, the interval [2, 5] can be directly modelled as a fuzzy set as shown in Fig. 7.2. Although one could argue that this set is not truly "fuzzy" since the membership degree of its points is either 0 or 1, this phenomenon is the consequence of the fact that, by design, there is no uncertainty on the endpoints of the interval. The uncertainty, in this context, is not given by the membership degree but by the width of the interval itself.

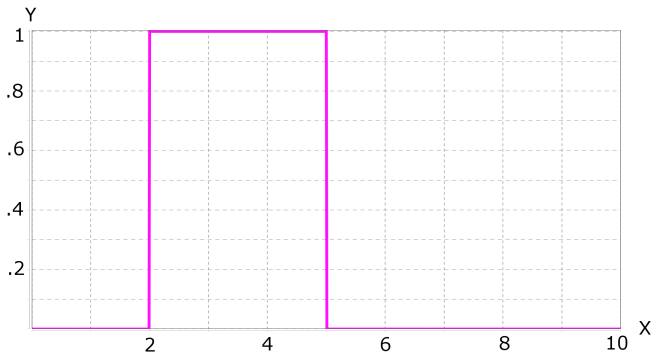


Figure 7.2: *A fuzzy set modelling the interval [2,5]*

There are four main approaches in the literature that generate fuzzy sets from the aggregation of intervals, focusing on different properties.

The IA [73] and EIA [74] have the goal of producing IT2 fuzzy sets with a practical parametric shape. Before generating the IT2 set, a pre-processing step is used to eliminate all the intervals that are considered outliers or non-sensical (i.e. non-overlapping intervals or intervals that do not fall within specific thresholds). The IAA [38] and EIAA [74], on the other hand, have as their main objective the representation of the agreement among the opinions expressed by the participants. They generate T1 or T2 fuzzy sets (depending on the different levels of uncertainty that are modelled) in which the membership degree is determined by the number of overlapping intervals in a given point. In contrast to the IA and EIA, the IAA and EIAA do not require a pre-processing step and model all the available data without discarding any entries, under the assumption that outliers and non-overlapping intervals may represent admissible entries (e.g. valid answers to a survey that differ from all the other ones).

Although the shape of a fuzzy set is one of the main properties by which humans give it a semantic interpretation, none of the approaches in the literature focus on keeping a relation with the shape used to represent a single interval. In this context, specific shapes of the footprint of uncertainty and of the embedded sets can help intuitively understand what the model represents and the effects of uncertainty on it, making it understandable also by

non-experts.

7.5 The constrained parametric approach

In this section, the constrained parametric approach (CPA) is introduced. This novel methodology can be used to model uncertain data acquired from multiple sources in which each instance (e.g. each survey answer) is represented by a parametric fuzzy set. The goal of the approach is to generate fuzzy sets with a strong semantic connection with the concepts in the original data by preserving their structure throughout the modelling process.

In order to do so, the generated fuzzy set and its embedded sets keep the same same parametric shape used to model an individual instance: the footprint of uncertainty is used to model the effect of the uncertainty generated by the aggregation on the parameters of the shape (e.g. the endpoints, for an interval), while restricting the shape of the embedded sets ensures that embedded sets that could not plausibly represent an instance, are excluded. These properties can be implemented with CIT2 fuzzy sets as they can be easily translated into mathematical constraints.

The parametric shape used to represent the individual instances, together with data statistics for each of its n parameters, are used to determine $n+1$ constraints $C_0, ..., C_n$. Specifically, the first constraint C_0 makes sure that all the acceptable embedded sets have the same parametric shape used to model individual observations (e.g. when aggregating intervals, all the embedded sets must have an interval shape too). Then, for each parameters P^i, $1 \leq i \leq n$ used to identify the chosen shape (e.g. the endpoints a, b for intervals or the points a, b, c for triangles) an uncertainty range $[P^i_{min}, P^i_{max}]$ is defined. The value of each parameter P^i for each embedded set must be $P^i \in [P^i_{min}, P^i_{max}]$ for it to be considered acceptable.

The process can be summarized in the following steps:

1. Determine the n parameters $P^1, ..., P^n$ required by the shape used to

model each individual instance (e.g. interval, triangular, trapezoidal).

2. For each parameter $P^i, 1 \leq i \leq n$ of the parametric shape (e.g. for each endpoint, in the case of an interval), determine its uncertainty range $[P^i_{min}, P^i_{max}]$. In this step, a combination of different data statistics such as mean, variance and standard deviation can be used to determine the uncertain endpoints (more on this in Sec. 7.6).

3. Build the constraint C_0 to ensure that all the acceptable embedded sets have a membership function that can be written in the same form of the parametric shape used in step 1 (e.g. when combining intervals, all the acceptable embedded sets must have an interval shape too, but with specific constraints limiting the possible variations).

4. For each parameter $P_i, 1 \leq i \leq n$ of the parametric shape, define the constraint C_i that is satisfied if and only if $P_i \in [P^i{}_{min}, P^i{}_{max}]$.

5. Use the constraints $C_0, ..., C_n$ defined in the steps 3, 4 to build the CIT2 fuzzy set that aggregates the observations, as in (7.1).

7.5.1 Combining intervals with the CPA

In this subsection, the CPA is used to aggregate intervals. A step-by-step application of the approach described here can be found in Sec. 7.6.1. Since an interval $[a, b]$ is identified by the two parameters a, b, the total number of constraints to define is three:

- C_0:

$$\exists\, a, b, \in \mathbb{R}, a \leq b : \mu_S(x) = \begin{cases} 1, & x \in [a, b] \\ 0, & \text{otherwise} \end{cases} \tag{7.4}$$

- C_1: $a \in [a_{min}, a_{max}]$

- C_2 $b \in [b_{min}, b_{max}]$

for each acceptable embedded set, C_0 ensures that it is shaped as an interval, while C_1 and C_2 make sure that its endpoints are within the respective uncertainty ranges. Intuitively, the CIT2 fuzzy set generated using C_0, C_1 and C_2 is determined by the collection of all the intervals with the endpoints within the uncertainty ranges used in C_1 and C_2. To fully characterize the generated CIT2 fuzzy set, the boundaries of its footprint of uncertainty must be determined. In this case, the upper-bound membership function is:

$$\overline{\mu}_{\breve{A}}(x) = \begin{cases} 1, & x \in [a_{min}, b_{max}] \\ 0, & \text{otherwise} \end{cases} \tag{7.5}$$

Proof. For all the values x, $x < a_{min} \vee x > b_{min}$ the membership degree for both the boundary function must be 0, since a set S' such that $\mu_{S'}(x) = 1$, $x < a_{min} \vee x > b_{min}$ would not satisfy C_0 and would therefore not be part of the CAES. For the values of $x' \in [a_{min}, b_{max}]$, trivially $\overline{\mu}_{\breve{A}}(x') = 1$.

In fact, for each value x', there exists a set S' modelling the interval $[a', b'] \subseteq [a_{min}, b_{max}]$, $a' \leq x' \leq b'$ for which $\mu_{S'}(x') = 1$ (because of C_0). The lower-bound membership function $\underline{\mu}_{\breve{A}}$ can be obtained analogously by analysing the two parts of the universe of discourse $[a_{min}, a_{max}) \cup (b_{min}, b_{max}]$ and $[a_{max}, b_{min}]$.

□

While the lower-bound membership (7.3) is:

$$\underline{\mu}_{\breve{A}}(x) = \begin{cases} 1, & a_{max} < b_{min} \wedge x \in [a_{max}, b_{min}] \\ 0, & \text{otherwise} \end{cases} \tag{7.6}$$

Proof. Using a proof that is similar to the one used for the upper-bound membership function in the previous subsection, it is possible to show that $\forall x' \in [a_{min}, a_{max}) \cup (b_{min}, b_{max}]$, $\underline{\mu}_{\breve{A}}(x') = 0$. For each value of x', in fact, there exists a set S' modelling the interval $[a', b'] \subset ([a_{min}, a_{max}) \cup (b_{min}, b_{max}])$, $a' > x' \vee b' < x'$ for which $\mu_{S'}(x') = 0$ (because of C_0).

In the case in which $x' \in [a_{max}, b_{min}]$, instead, there are two cases:

1. $[a_{max}, b_{min}]$ is not a well-formed interval, i.e. $b_{min} < a_{max}$: in this circumstance, the whole section of the universe of discourse $[a_{min}, b_{max}]$ is covered by the case analysed above (i.e. when $x' \in [a_{min}, a_{max}) \cup (b_{min}, b_{max}]$). In this case, the lower-bound membership degree of the CIT2 set $\breve{A}$ is:

$$\underline{\mu}_{\breve{A}}(x) = \begin{cases} 0, & x \in [a_{max}, b_{min}] \\ 0, & \text{otherwise} \end{cases} \tag{7.7}$$

   I.e., $\underline{\mu}_{\breve{A}}(x) = 0$, $\forall x \in X$, with X universe of discourse

2. $[a_{max}, b_{min}]$ is a well-formed interval, i.e. $b_{min} > a_{max}$: in this sub-case, the membership degree of the lower-bound function is 1 for $\forall x' \in [a_{max}, b_{min}]$. To prove that, it is shown that for all the sets S' modelling an interval $[a', b'] \supseteq [a_{max}, b_{min}]$[1] and satisfying C_0, C_1, C_2, $\mu_{S'}(x) = 1$, $\forall x \in [a_{max}, b_{min}]$.

   Since $x \in [a_{max}, b_{min}] \wedge [a_{max}, b_{min}] \subseteq [a', b'] \implies x \in [a', b']$. Since S' satisfies C_0, it follows that $\mu_{S'}(x) = 1$, $\forall x \in [a_{max}, b_{min}]$. In this sub-case, the lower-bound membership function of the CIT2 set $\breve{A}$ can therefore be written as:

$$\underline{\mu}_{\breve{A}}(x) = \begin{cases} 1, & x \in [a_{max}, b_{min}] \\ 0, & \text{otherwise} \end{cases} \tag{7.8}$$

□

The constraints chosen for the modelling of intervals also simplify the type-reduction process. To reduce a CIT2 fuzzy set $\breve{A}$, it is necessary to identify the two acceptable embedded sets with the lowest and highest centroid value, respectively l and r. Since all the acceptable embedded sets model intervals

[1] Any set S' modelling an interval $[a', b'] \subset [a_{max}, b_{min}]$ would not satisfy C_1, C_2

(because they satisfy C_0), the formula for the calculation of their centroid can be simplified as follows:

$$C(S) = \frac{a' + b'}{2} \tag{7.9}$$

with S being a generic acceptable embedded set modelling the interval $[a', b']$, $a' \in [a_{min}, a_{max}]$, $b' \in [b_{min}, b_{max}]$. Taking into account the constraints C_1, C_2 described above, the lowest centroid l can be determined as:

$$l = \min_{S \in \text{CAES}_{\breve{A}}} C(S) = \min_{a \in [a_{min}, a_{max}], b \in [b_{min}, b_{max}]} \frac{a' + b'}{2} \tag{7.10}$$

To minimize the fraction $\frac{a'+b'}{2}$ in (7.10), it is sufficient to minimize a and b. Therefore (7.10) can be rewritten as:

$$l = \frac{a_{min} + b_{min}}{2} \tag{7.11}$$

Analogously, it can be proven that r is computed as:

$$r = \frac{a_{max} + b_{max}}{2} \tag{7.12}$$

By construction, the acceptable embedded sets providing l and r are, respectively, the lowest and highest intervals embedded in the CIT2 set. Therefore, they represent the lowest and highest opinion within the footprint of uncertainty. This property also contribute to the interpretability of the model, compared to models consisting of IT2 fuzzy sets. In fact, since the defuzzified value of the CIT2 set produced by the CPA is computed as $(l+r)/2$, it can be intuitively expressed as the average of the lowest and highest embedded opinions. This provides a clear, meaningful explanation for how the type-reduction and defuzzification process is carried out, understandable also by non-experts.

Explaining the type-reduction and defuzzification for a generic IT2 fuzzy set like the ones produced by the other approaches in the literature, on the other

hand, would be more challenging. When procedures like the Karnik-Mendel one are used for the type-reduction, there is no guarantee that the embedded sets providing l and r will carry a particular meaning or can be interpreted as opinions, due to their shape. Therefore, giving a human understandable reason for how the defuzzified value was computed is significantly harder.

7.5.2 Modelling other shapes: triangles

As already mentioned in this section, the CPA can be applied not only to intervals but to any uncertain data with instances modelled through parametric fuzzy sets. Here, the case in which each observation is modelled as a triangular T1 fuzzy set is analyzed. A triangular membership function is described by three parameters a, b, c identifying respectively the start, peak and end points of the triangle. The first constraint C_0 ensures that each acceptable embedded set has a triangular membership function, while the constraints C_1, C_2, C_3 (one per parameter) are defined to check that each of the parameters of the membership function are within the respective uncertainty ranges.

Therefore, the constraints that each acceptable embedded set S must satisfy are the following:

- C_0: this constraints ensures that S is an actual triangle with parameters $[a, b, c]$, $a < b < c$, $a, b, c \in \mathbb{R}$. The membership function of S must be in the following form:

$$\mu_S(x) = \begin{cases} \frac{x-a}{b-a}, & x \in [a, b] \\ \frac{b-x}{c-b} & x \in (b, c] \\ 0 & x < a \vee x > c \end{cases} \tag{7.13}$$

- C_1: $a \in [a_{min}, a_{max}]$

- C_2: $b \in [b_{min}, b_{max}]$
- C_3: $c \in [c_{min}, c_{max}]$

The CAES is defined as the set of acceptable embedded sets satisfying these four constraints:

$$\text{CAES} = \{S \,|\, \mu_S : X \mapsto [0,1], C_0(S) \wedge C_1(S) \wedge C_2(S) \wedge C_3(S)\} \tag{7.14}$$

The upper membership functions of each CIT2 fuzzy set $\breve{A}$ can be expressed as follows:

$$\overline{\mu}_{\breve{A}}(x) = \begin{cases} \frac{x-a_{min}}{b_{min}-a_{min}}, & x \in [a_{min}, b_{min}] \\ 1 & x \in (b_{min}, b_{max}) \\ \frac{b_{max}-x}{c_{max}-b_{max}}, & x \in [b_{max}, c_{max}] \\ 0, & x < a \vee x > c \end{cases} \tag{7.15}$$

Proof. To prove (7.15), first the universe of discourse is divided in three parts $[a_{min}, b_{min}], (b_{min}, b_{max}), [b_{max}, c_{max}]$ and show that for each of these intervals:

$$\max_{S \in \text{CAES}_{\breve{A}}} \mu_S(x) = (7.15) \tag{7.16}$$

- $x \in [a_{min}, b_{min}]$: since each triangular set S with parameters is in the form of (7.13), when $x \in [a_{min}, b_{min}]$, considering that S must also satisfy $C_1, ..., C_3$ to be in the CAES, its membership function becomes:

$$\mu_S(x) = \frac{x-a}{b-a} \tag{7.17}$$

Therefore, to obtain the maximum in (7.16), it is necessary to choose the values $a \in [a_{min}, a_{max}]$ and $b \in [b_{min}, b_{max}$ that maximize $\frac{x-a}{b-a}$. The values that maximize it are $a = a_{min}$ and $b = b_{min}$ as shown by the following proof. For readability, $\underline{a}$ and $\underline{b}$ will be used instead of a_{min} and b_{min}.

Theorem 7.1. *Given two values* $b' \in (b_{min}, b_{max}], a' \in (a_{min}, a_{max}]$ *(i.e.* $a' > a_{min}$, $b' > b_{min}$*) then:*

$$\frac{x - \underline{a}}{\underline{b} - \underline{a}} \geq \frac{x - a'}{b' - a'}, \underline{a} = a_{min}, \underline{b} = b_{min} \tag{7.18}$$

Proof. (7.18) is equivalent to:

$$\frac{x - \underline{a}}{\underline{b} - \underline{a}} - \frac{x - a'}{b' - a'} \geq 0 \tag{7.19}$$

$$\frac{(x - \underline{a})(b' - a') - (x - a')(\underline{b} - \underline{a})}{(\underline{b} - \underline{a})(b' - a')} \geq 0 \tag{7.20}$$

The numerator and denominator of (7.20) can be analyzed separately.

$$\begin{aligned} &(x - \underline{a})(b' - a') - (x - a')(\underline{b} - \underline{a}) = \\ &x(b' - a' - \underline{b} + \underline{a}) - \underline{a}b' + a'\underline{b} \geq 0 \end{aligned} \tag{7.21}$$

Since $a' > \underline{a}$ and $b' > \underline{b}$ that can be written respectively as:

$$a' = \underline{a} + \epsilon_a, \epsilon_a > 0 \quad b' = \underline{b} + \epsilon_b, \epsilon_b \geq 0 \tag{7.22}$$

(7.21) can be rewritten as:

$$\begin{aligned} &x(\underline{b} + \epsilon_b - \underline{a} - \epsilon_a - \underline{b} + \underline{a}) - \underline{a}(\underline{b} + \epsilon_b) + (\underline{a} + \epsilon_a)\underline{b} = \\ &x(\epsilon_b - \epsilon_a) + \underline{b}\epsilon_a - \underline{a}\epsilon_b = \\ &x\epsilon_b - x\epsilon_a + \underline{b}\epsilon_a - \underline{a}\epsilon_b = \\ &\epsilon_a(\underline{b} - x) + \epsilon_b(x - a) \geq 0 \end{aligned} \tag{7.23}$$

□

Since $\epsilon_a, \epsilon_b > 0$, (7.18) is true $\forall x \in [\underline{a} = a_{min}, \underline{b} = b_{min}]$. The denominator of (7.20) is trivially always positive: $\underline{b}$ and $\underline{a}$ are two of the parameters of an acceptable embedded set $\underline{S}$ modelling a triangle and satisfying the constraints $C_0, ..., C_3$. Therefore, it must be that $\underline{b} > \underline{a}$ because of C_0. Analogously, $b' > a'$.

- $x \in (b_{min}, b_{max})$. This case is trivial. Since for each acceptable embedded set $S \in \text{CAES}\mu_S(b) = 1$, the acceptable embedded set S' modelling the triangle with parameters $(a_{min}, x, c_{max}$ satisfies all the constraints $C_0, ..., C_3$ and $\mu_{S'}(x) = 1$.

- $x \in [b_{max}, c_{max}]$. The proof is analogous to the $x \in [a_{min}, b_{min}]$ case.

□

The lower membership, instead, function is:

$$\underline{\mu}_{\breve{A}}(x) = \begin{cases} 0, & x < a_{max} \vee x > c_{min} \\ \min\left(\frac{x-a_{max}}{b_{max}-a_{max}}, \frac{b_{min}-x}{c_{min}-b_{min}}\right), & otherwise \end{cases} \tag{7.24}$$

Proof. For this proof, the universe of discourse will be split into two intervals: $[-\infty, a_{max}) \cup (c_{min}, +\infty]$ and $[a_{max}, c_{min}]$ and it will be shown that for both cases $\mu_{\breve{A}}(x) =$ (7.24).

- $x \in [a_{max}, c_{min}]$. Since the membership function of each set S in the CAES is in the form of (7.13), the lower-bound membership function of $\breve{A}$ can be written as:

$$\min_{S \in \text{CAES}_{\breve{A}}} \mu_S(x) = \min\left(1, \frac{x-a}{b-a}, \frac{b-x}{c-b}\right) \tag{7.25}$$

  I.e., it is the minimum among the three cases. Similarly to what has been done for the the proof of the upper-bound membership function (in which these quantities had to be maximized), it can be shown that:

$$\begin{aligned} \min\left(\frac{x-a}{b-a}\right) &= \frac{x-a_{max}}{b_{max}-a_{max}} \\ min\left(\frac{b-x}{c-b}\right) &= \frac{b_{max}-x}{c_{max}-b_{max}} \end{aligned} \tag{7.26}$$

- $x \in [-\infty, a_{max}) \cup (c_{min}, +\infty]$. This case can be split in two sub-cases, i.e. when $x \leq a_{max}$ and when $x \geq c_{min}$. Consider the two sets $S', S'' \in$

$\text{CAES}_{\breve{A}}$ modelling respectively the triangles with parameters $a_{min}, b_{min}, c_{min}$ and $a_{max}, b_{max}, c_{max}$. When $x \leq a_{max}$, then:

$$\mu_{S''}(x) = 0 = \min_{S \in \text{CAES}_{\breve{A}}} \mu_S(x) = \min\left(1, \frac{x-a}{b-a}, \frac{b-x}{c-b}\right) \tag{7.27}$$

Instead, when $x \geq c_{min}$:

$$\mu_{S'}(x) = 0 = \min_{S \in \text{CAES}_{\breve{A}}} \mu_S(x) = \min\left(1, \frac{x-a}{b-a}, \frac{b-x}{c-b}\right) \tag{7.28}$$

Therefore, in both sub-cases:

$$\min\left(1, \frac{x-a}{b-a}, \frac{b-x}{c-b}\right) = 0 \tag{7.29}$$

□

Just like any other CIT2 (and in general, IT2) fuzzy set, $\breve{A}$ can be type-reduced into a T1 fuzzy set fully represented by the interval $[l, r]$, where l and r are respectively the minimum and maximum centroids obtainable by defuzzifying all the acceptable embedded sets of $\breve{A}$. Since each acceptable embedded set is a T1 triangular fuzzy set with parameters (a, b, c), $[l, r]$ can be computed as follows:

$$[l, r] = [\frac{a_{min} + b_{min} + c_{min}}{3}, \frac{a_{max}, b_{max}, c_{max}}{3}] \tag{7.30}$$

7.6 Applications

To use the CPA in practice, it is necessary to determine the endpoints of the uncertainty ranges (i.e. the values P^i_{min} and P^i_{max} for each parameter P^i) in order to use them in the constraints and define the acceptable embedded sets.

The specific metrics and process used to determine them, can vary and depend on the quality and size of the data that the designer is dealing with. As an initial suggestion, it is proposed the following process to determine the

uncertainty range for each of the n parameters P^i, $1 \leq i \leq n$:

- The values P^i_{avg} is computed averaging the value of the parameter P^i among all the observations or opinion modelling to aggregate (e.g. all the opinions for the word *medium*).
- Similarly, the standard deviation P^i_{std} is computed.
- The endpoints of the uncertainty range $[P^i_{min}, P^i_{max}]$ are defined as follows:

$$P^i_{min} = P^i_{avg} - (P^i_{std}/2) \tag{7.31}$$

$$P^i_{max} = P^i_{avg} + (P^i_{std}/2) \tag{7.32}$$

To summarize, each uncertainty range in this chapter is computed as follows:

$$[P^i_{avg} - (P^i_{std}/2), P^i_{avg} + (P^i_{std}/2)] \tag{7.33}$$

The choice of using the average and standard deviation has been made heuristically. Hence, alternative metrics may equally be used by the designer, according to their needs and the quality of the data they have. In case of noisy data, for example, a preprocessing step can be added to remove all the nonsensical data and the outliers or different data statistics can be used. One of the strengths of the CPA, in fact, is that it can be easily adapted to different scenarios simply by changing the process that computes P^i_{min} and P^i_{max} for each parameter P^i.

7.6.1 Step-by-step application on interval-valued synthetic data

The following synthetic example shows a step-by-step application of the CPA on interval-valued data, to facilitate the understanding of how it can be used in practice. In the next subsection, the CPA is applied on real world-data, in a more thorough case study.

Consider the three intervals $I_1 = [1, 3]$, $I_2 = [2, 5]$, $I_3 = [4, 7]$ represented in Fig. 7.3 modeling a given concept (e.g. the word *some*) for three different people.

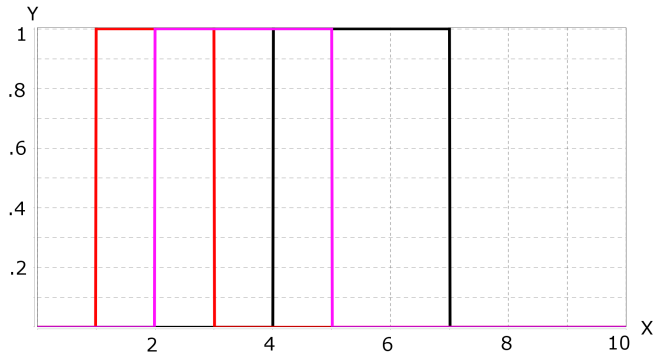


Figure 7.3: *Three intervals (in red, magenta and black) modelling the same concept*

In order to build a CIT2 fuzzy set $\breve{A}$ that aggregates them, the uncertainty range for each parameter of the chosen shape must be computed. Since the data is modelled as intervals, the shape has only two parameters: the endpoints a and b. In this case, they are calculated as described in (7.33), producing for the left endpoint the uncertainty range:

$$[2.333 - (1.247/2),\ 2.333 + (1.247/2)] = [1.709,\ 2.956] \tag{7.34}$$

And for the right endpoint:

$$[5 - (1.632/2),\ 5 + (1.632/2)] = [4.183,\ 5.816] \tag{7.35}$$

Therefore, the acceptable embedded sets S satisfy the following constraints:

- C_1: S must correctly model an interval, i.e.:

$$\exists\, a, b \in \mathbb{R},\ a \leq b : \mu_S(x) = \begin{cases} 1, & x \in [a, b] \\ 0, & \text{otherwise} \end{cases} \tag{7.36}$$

- C_2: $a \in [1.709,\ 2.956]$
- C_3 $b \in [4.183,\ 5.816]$

The collection of these acceptable embedded represents the CAES, as defined in (7.1).

The upper (7.5) and lower (7.6) membership functions of the CIT2 fuzzy set are respectively:

$$\overline{\mu}_{\breve{A}}(x) = \begin{cases} 1, & x \in [1.709, 5.815] \\ 0, & \text{otherwise} \end{cases} \tag{7.37}$$

$$\underline{\mu}_{\breve{A}}(x) = \begin{cases} 1, & x \in [2.956, 4.183] \\ 0, & \text{otherwise} \end{cases} \tag{7.38}$$

The CIT2 fuzzy set $\breve{A}$ obtained is shown in Fig. 7.4.

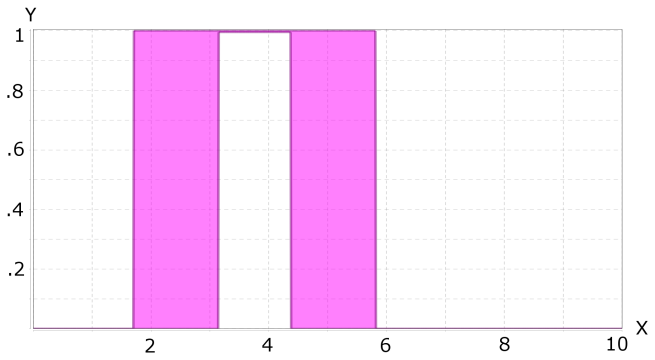


Figure 7.4: *The CIT2 fuzzy set $\breve{A}$ obtained from the aggregation of the three intervals in Fig. 7.3 with the CPA*

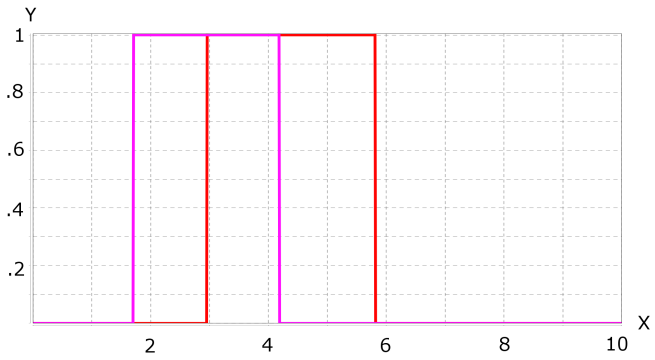


Figure 7.5: *The two acceptable embedded sets (in magenta and red) determining the type-reduced set of $\breve{A}$*

From the picture, it is possible to see how this set keeps a shape relation with the T1 representation of an interval, making it easily interpretable. Just

by looking at the CIT2 set, it is clear that it models an interval with uncertainty on its endpoints (i.e. the two thick magenta sections of the footprint of uncertainty). The level of uncertainty on each endpoint is easy to understand as it is modelled by the 'width' of the uncertain boundaries (i.e. a wider footprint of uncertainty means higher uncertainty).

The two acceptable embedded sets representing the lowest and highest acceptable opinion that determine the type-reduced set $[2.945, 4.386]$ are shown in Fig. 7.5.

7.6.2 Application on real-world interval-valued data and comparison with IA, EIA, IAA

The CPA has also been applied to intervals gathered from real surveys, to see how it performs on non-synthetic data that can be noisy and contain outliers or 'bad' (i.e. non-sensical) answers. The data chosen is available online[2] and has been already used in other research works [74, 76]. A total of 174 participants were asked, in an online survey, to provide the interval in $[0, 10]$ that in their opinion better represents a given word (such as *small, medium, some*), for a total of 32 words. In addition to the novel CIT2 modelling technique presented in this chapter, also the interval approach (IA, [73]), the enhanced interval approach (EIA, [74]) and the interval agreement approach (IAA, [38]) have been used to model the data, for comparison. For the IA and EIA implementation, the freely available Matlab library has been used [2]; the IAA

[2]http://sipi.usc.edu/∼ mendel/publications/index.html

algorithm, instead, has been taken from the Python library FuzzyCreator [77].

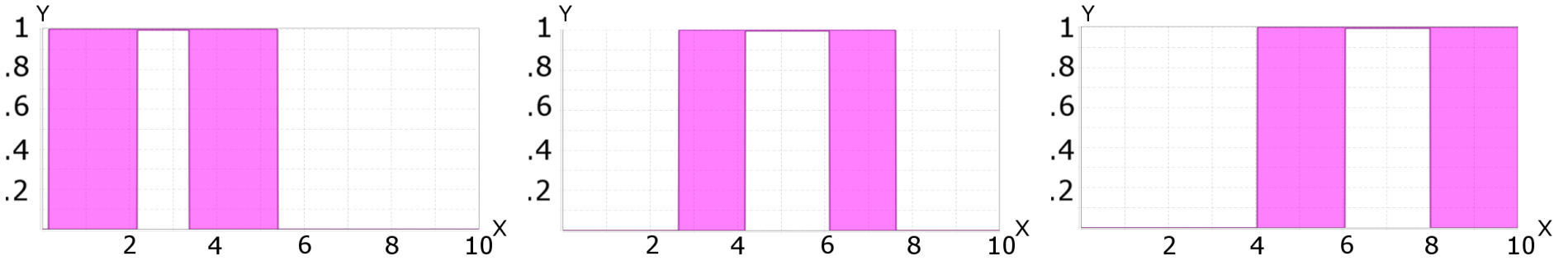
Table 7.1: Comparison of the type-reduced sets

Word	**CIT2**	**EIA**	**IA**
Small	[1.283, 3.314]	[0.452, 1.825]	[0.454, 2.222]
Medium	[4.363, 5.878]	[3.164, 6.923]	[1.983, 8.167]
Large	[6.500, 8.506]	[7.861, 9.385]	[6.503, 9.377]

Table 7.2: Comparison of the centroid defuzzified values

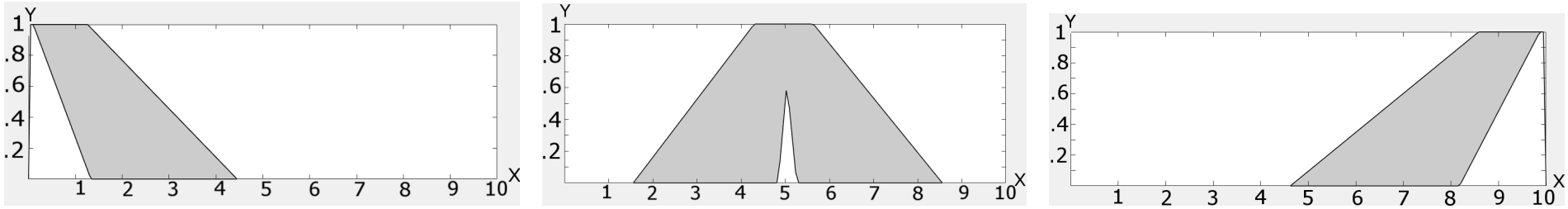
Word	**CIT2**	**EIA**	**IA**	**IAA**
Small	2.298	1.138	1.338	3.488
Medium	5.120	5.043	4.369	5.154
Large	7.504	8.623	7.940	6.522

From the 32 words available, because of space limitations, three words have been chosen for the comparison: *small, medium* and *large*. The fuzzy sets obtained with the CPA, EIA, IA and IAA approaches are shown in Figs. 7.6, 7.7, 7.8, 7.9 respectively while the type reduced and centroid defuzzified values for each of the sets are reported in the Tables 7.1, 7.2.



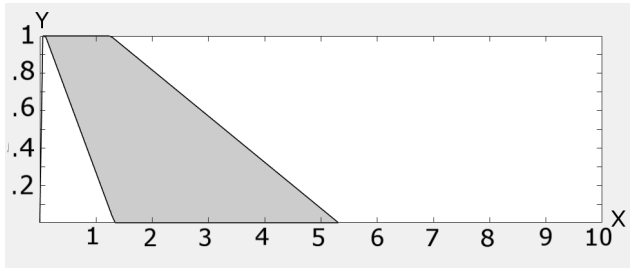
(a) *CIT2 fuzzy set for* small **(b)** *CIT2 fuzzy set for* medium **(c)** *CIT2 fuzzy set for* large

Figure 7.6: *Modelling of the words* small, medium *and* large *with the CPA*

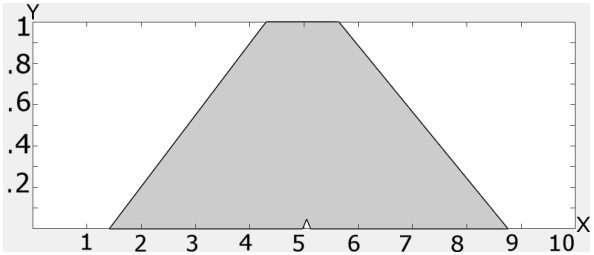


(a) *IT2 fuzzy set for* small **(b)** *IT2 fuzzy set for* medium **(c)** *IT2 fuzzy set for* large

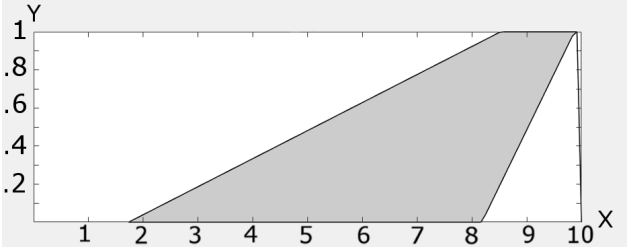
Figure 7.7: *Modelling of the words* small, medium *and* large *with the EIA*



(a) *IT2 fuzzy set for* small

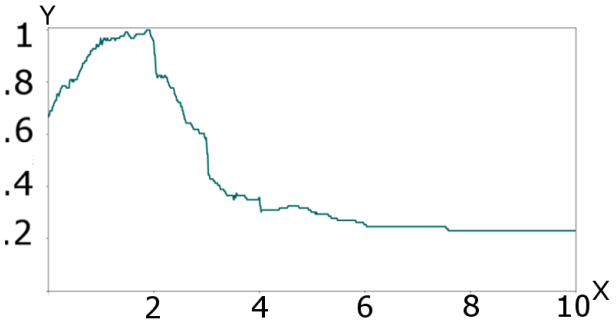


(b) *IT2 fuzzy set for* medium

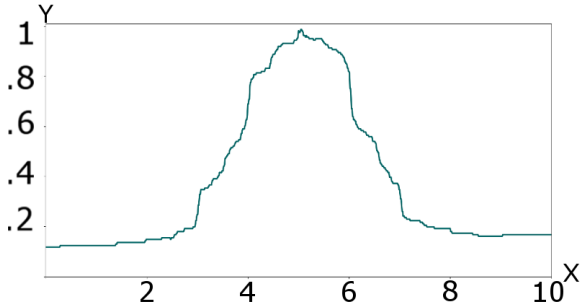


(c) *IT2 fuzzy set for* large

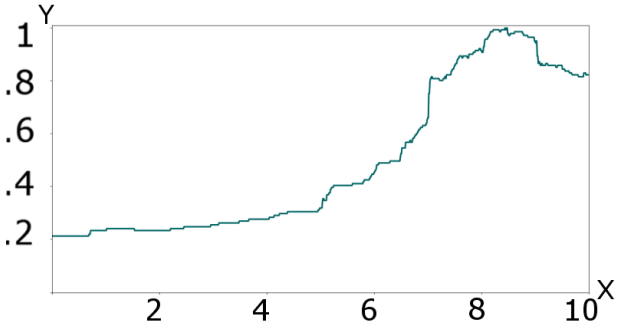
Figure 7.8: *Modelling of the words* small, medium *and* large *with the IA*



(a) *T1 fuzzy set for* small



(b) *T1 fuzzy set for* medium

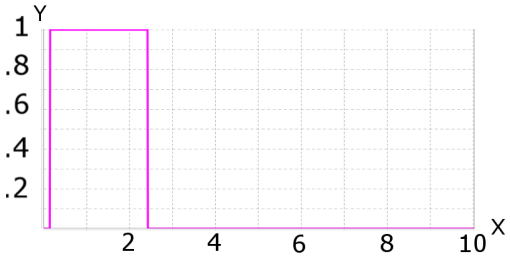


(c) *[T1 fuzzy set for* large

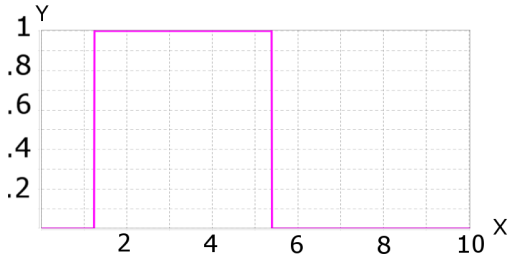
Figure 7.9: *Modelling of the words* small, medium *and* large *with the IAA*

7.6.3 Application on real world triangular data

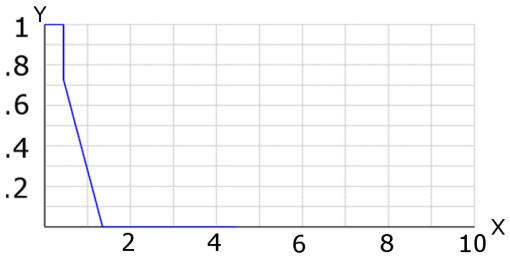
The experiments so far were focused on the modelling of intervals. The CPA, however, can be used with uncertain data with instances modelled through any parametric fuzzy set, such as triangular ones. Each triangle, in fact, can be represented with three parameters a, b, c being respectively the starting point, the peak and the endpoint of the triangle. To show how the CPA can be used to generate triangles, the same intervals gathered from surveys for the experiments in the previous section have been turned into triples with the following strategy: each interval $[a, b]$ has been transformed into the triple $(a, (a+b)/2, b)$ representing the three parameters (a, b, c) of a triangular shape. This conversion has been adopted just to show an example of how the CPA could be used to model triangles or if a designer explicitly wants to obtain triangular shapes from interval data.



(a) *Acceptable embedded set determining the left-endpoint of the type-reduced set of the CIT2 set in Fig. 7.6a*



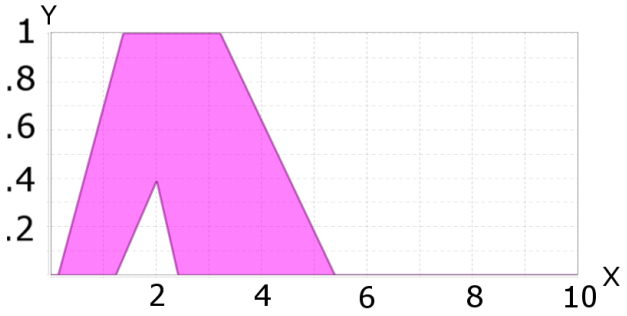
(b) *Acceptable embedded set determining the right-endpoint of the type-reduced set of the CIT2 set in Fig. 7.6a*



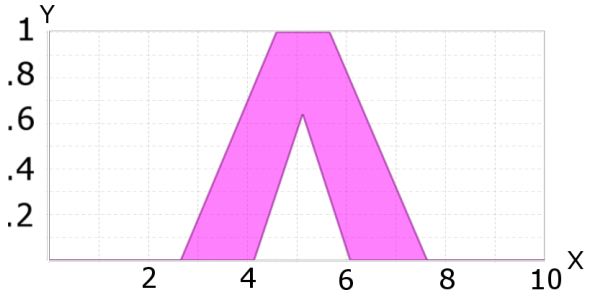
(c) *Embedded set determining the left-endpoint of the type-reduced set of the IT2 set in Fig. 7.7a (EIA)*

(d) *Embedded set determining the right-endpoint of the type-reduced set of the IT2 set in Fig. 7.7a (EIA)*

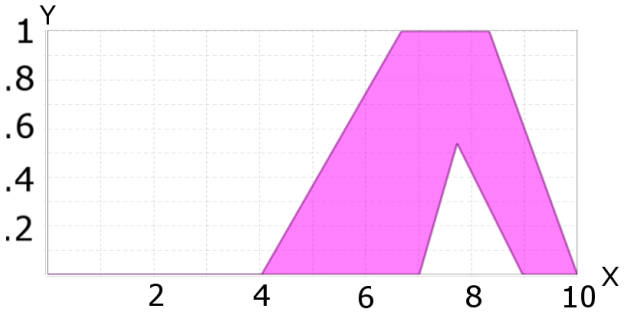
Figure 7.10: *Embedded sets determining the type-reduced set for the word small with the CIT2 (top row) and EIA*



(a) *CIT2 fuzzy set modelling the word* small *when representing each opinion as a triangle*

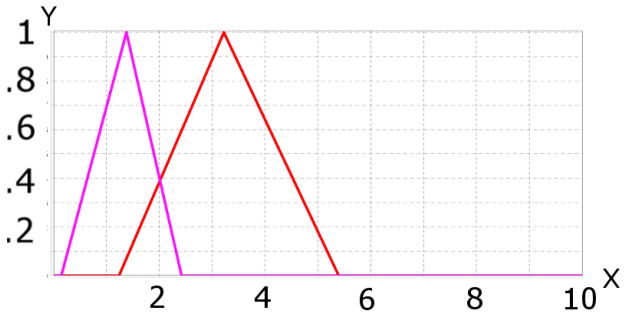


(b) *CIT2 fuzzy set modelling the word* medium *when representing each opinion as a triangle*

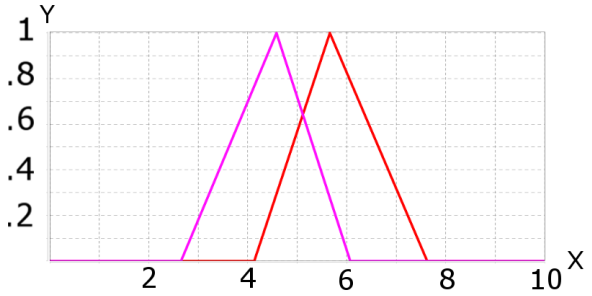


(c) *CIT2 fuzzy set modelling the word* large *when representing each opinion as a triangle*

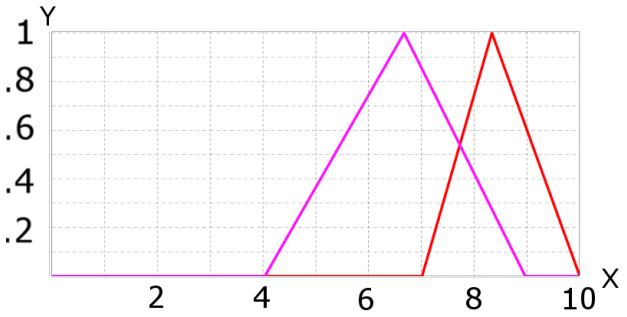
Figure 7.11: *Modelling of the words* small, medium *and* large *with the CPA when each collected opinion is modelled as a triangle*



(a) *Acceptable embedded sets used to type-reduce the CIT2 fuzzy set modelling the word* small



(b) *Acceptable embedded sets used to type-reduce the CIT2 fuzzy set modelling the word* medium



(c) *Acceptable embedded sets used to type-reduce the CIT2 fuzzy set modelling the word* large

Figure 7.12: *Acceptable embedded sets determining the type-reduced set of the CIT2 fuzzy sets in Fig. 7.11*

The three CIT2 fuzzy set modelling the words *small, medium* and *large* using a triangular shape to model each of the collected opinions, are shown in Fig. 7.11. The acceptable embedded sets determined the respective $[l, r]$ values during the type-reduction, are depicted in Fig. 7.12 instead.

7.7 Discussion

7.7.1 Interval-valued data

From the experiments carried out in the previous section, the most evident difference between the approaches is in the shape and type of the fuzzy sets they produce.

The IA (Fig. 7.8) and EIA (Fig. 7.7) both produce triangular or shoulder shape, with the EIA being specifically designed to generate a narrower footprint of uncertainty. Their main goal is to produce practical parametric IT2 fuzzy sets to model each word. Outliers, non-sensical and non-overlapping intervals are removed in a preprocessing step, before the set is created (for more details, the reader can refer to the original papers [73, 74]). The shape and the tolerance to noise and "bad" data, makes both IA and EIA very useful for practical applications. However, the heavy preprocessing applied can discard up to 90% of the collected intervals before the IT2 fuzzy set is generated [76]. While this behaviour does not represent a problem in this context, it may reduce the amount of data to just a few points, if the starting dataset is already small. Additionally, in some contexts, outliers may represent valid opinions that simply differ from the rest; therefore removing them may not be desirable.

The IAA (Fig. 7.9) models a different aspect of the data and focuses on the representation of the agreement among the participants. In contexts in which only inter-expert variability is present, as in this experiment, the algorithm produces T1 fuzzy sets. The membership degree for each point x in the universe

of discourse is determined by the number of overlapping intervals in x. As a result, the shape generated is non-parametric and one of the possible drawbacks of this approach is that the sets produced may less practical for uses in real-world systems (although this issue is partly tackled by the EIAA).

The IAA does not require a preprocessing step (although it can be added) and does not remove any outliers. The goal, in fact, is to fully represent the data as is, without discarding any of the collected intervals or making any assumptions on what may or may not be a non-sensical answer.

The CPA proposed in this chapter (Fig. 7.6), prioritizes different aspects compared to the other algorithms. The aim is to preserve a strong connection between the representation of a single opinion and the CIT2 fuzzy set in order to increase the intuitive understanding of the model. In fact, the shapes used to generate the CIT2 set make them easily interpretable as intervals with uncertainty on their endpoints. The footprint of uncertainty models the uncertainty around the endpoints, with a wider footprint of uncertainty corresponding to a higher degree of uncertainty.

The other main objective of the CIT2 approach is to avoid embedded sets that could not model valid opinions and to make the type-reduction more explainable. Thanks to the constraints $C_0, C_1.C_2$, only the embedded sets that model an interval and that lie within the footprint of uncertainty are considered as acceptable. As a consequence, the two acceptable embedded sets that determine the type-reduced set represent respectively the lowest and highest acceptable opinion. These acceptable embedded sets for the word *small* are shown in Figs. 7.10a and 7.10b. For comparison, the embedded sets used to type-reduce the set modelling the same word with the EIA are reported in Figs. 7.10c, 7.10d. The defuzzified value of the CIT2 set they belong to, is therefore obtained as the average of the lowest and highest acceptable opinion.

In the set generated by the EIA, since the latter algorithm focuses on other properties of the data, it is harder to give an intuitive meaning to the embedded sets determining the type-reduction and as a consequence, also providing an

interpretable explanation for the generation of the defuzzified output. This phenomenon is not limited to the word *small* as it extends to any fuzzy set that can be modelled both as a CIT2 and IT2 fuzzy set [1, 3, 65] since it depends on the way in which CIT2 and IT2 sets represent their embedded sets.

The properties showed by the CPA and the use of meaningful shapes for both the CIT2 fuzzy sets generated and their acceptable embedded sets, add an additional layer of interpretability that makes both the model itself and the operations on it intuitively understandable even by non-experts. The use of shapes with a clear semantic meaning, together with the capability of expressing complex operations such as the type-reduction and defuzzificaiton in simple, human-understandable terms represents the main advantage and contribution of this approach, making it a valid alternative to the other methodologies in the literature, in situations which have specific needs with respect to interpretability of the model.

One of the possible downsides of the CPA is that even though no preprocessing is necessary, using only the mean and standard deviation of the data to define C_1, C_2 implies that some of the opinions gathered during the survey, are not considered as *acceptable* anymore, i.e. some of the collected intervals would not lie within the footprint of uncertainty of the CIT2 set produced and would in fact be discarded. In a scenario in which this behaviour is problematic, however, the designer can simply use different constraints or data statistics to produce the CIT2 set from the collected data.

The CPA, in conclusion, represent a valid alternative to the other approaches in all the contexts in which it is necessary to produce a model with additional interpretability properties and for which the operations carried out can be explained in human-understandable terms. These requirements are becoming increasingly important in many real-world applications, especially in explainable artificial intelligence (XAI,[7]).

7.7.2 Triangular data - Flexibility of the approach

Also in the experiment in which the CPA has been applied on data where each individual observation is modelled as a triangle, it is possible to see how the CIT2 set produced preserves the shape throughout the modelling process. Changing the constraints is sufficient to generate a CIT2 fuzzy set with a triangular shape, showing the flexibility and potential of the novel approach.

All the three CIT2 fuzzy sets in Fig. 7.11 keep a triangular shape and by looking at the footprint of uncertainty, it is intuitively easy to understand that a plausible opinion is represented by any T1 triangle lying within it. In addition to that, during the type-reduction, the acceptable embedded sets determining the T1 set $[l, r]$, also have a triangular shape. This property creates a direct and humanly-understandable link between the CIT2 fuzzy set modelling the aggregation of opinion and the defuzzified value, obtained as the average of the lowest and highest opinion.

7.8 Limitations

Although this chapter introduced the CPA and presented a set of experiments showing its specific characteristics and its applicability in real-world contexts, there are some limitations that need to be addressed in future research work.

Firstly, for the CIT2 fuzzy sets to be constructed, in fact, it is necessary to establish a set of mathematical constraints $C_0, ..., C_n$ (for a shape with n parameters) that must be satisfied by all the acceptable embedded sets. Determining which constraints better suit a particular scenario or the designer's needs may not be trivial. In this chapter, the mean and standard deviation of each parameter have been recommended as they have experimentally produced sensible CIT2 fuzzy sets, even with noisy data. However, they may not be as suitable in other contexts.

Another limitation of the current approach concerns the upper and lower membership function and the type-reduction formulae that need to be derived

every time a new set of constraints is used. In this chapter, the formulae for both the interval and triangular shapes have been presented but determining the boundary functions given an arbitrary set of constraints may be very challenging (although it can be speculated that Gaussian, shoulder and trapezoidal shapes can be implemented just as easily as triangles).

7.9 Summary

This chapter proposes the constrained parametric approach: a novel method to model uncertain data with instances modelled through parametric fuzzy sets in a more interpretable way. By modelling the underpinning characteristics of the meaningfulness of a concept through a set of mathematical constraints, the CPA produces models that are intuitively easier to understand, both in the kind of data they represent and the uncertainty produced by the aggregation. Additionally, by restricting the shape of the embedded sets, the CPA also makes the type-reduction and defuzzification steps more explainable by expressing it in terms of the lowest and highest instance embedded in the footprint of uncertainty.

The novel approach has been compared with the three algorithms from the literature for the modelling of interval-valued data: the interval approach (IA), enhanced interval approach (EIA) and the interval agreement approach (IAA). They have all been applied to data gathered from real surveys and the features of the CPA have been extensively discussed and compared to the main characteristics of the other approaches.

Finally, it has also been shown how the CPA can be used with not only intervals but any parametric shape simply by changing the constraints upon which the model is built. Specifically, also the constraints and formulas for triangular instances have been derived and applied on real-world data.

In future work, it is desirable to develop a strategy to simplify the determination of the upper and lower membership functions of the combined fuzzy

set generated by the CPA given an arbitrary set of constraints, as well as comparing this novel approach with the other algorithms present in the literature on a wider set of problems and data types.

Chapter 8

Conclusion

8.1 Contributions

Fuzzy logic has been successfully used in the explainable artificial intelligence (XAI) field thanks to its ability to model the decision process through words and rules, in a manner that is similar to human reasoning and intuitive to understand.

In the real world, however, there are many sources of uncertainty that need to be taken into account when designing an intelligent system: sensor readings, faulty hardware, noise in the collected data or inaccuracies in human knowledge.

To better represent the uncertainty in intelligent systems, interval type-2 (IT2) fuzzy logic is often used. However, the ability of better handling the uncertainty, comes at the cost of an increased complexity and the need for additional steps during the inference. As a consequence, the semantic value of fuzzy sets is partially lost during the input-output mapping of traditional interval type-2 fuzzy logic systems (FLSs). By semantic value, we specifically refer to the capacity of interpreting the output of the fuzzy system in respect to the pre-defined and thus understood linguistic variables used for the antecedents and consequents of the system. As traditional interval type-2 fuzzy systems process *all* the embedded sets during the type-reduction step,

the resulting outputs can be substantially different from any of the input sets, making intuitive interpretation (based on the known models of the linguistic antecedent/consequent labels) challenging.

The goal of this thesis was to extend the recently established foundational work on constrained type-2 (CT2) fuzzy sets (FSs), specifically focusing on constrained interval type-2 (CIT2) FSs, to develop a framework that preserves the semantic value of CIT2 FSs throughout the inference, type-reduction and defuzzification stages in order to create a new class of CIT2 FLSs with improved semantic interpretability. These FLSs make the semantic mapping from the inputs to the outputs more intuitively interpretable, making them a valuable alternative to type-2 (T2) and IT2 FLSs in XAI.

To achieve this goal the following objectives were pursued:

1. Formally define the generation of CIT2 FSs for practical use.

2. Explore how to generate explainable systems from interpretable sets.

3. Make CIT2 FLSs usable in practical applications by producing a practical inference framework.

4. Validate the theory with real-world applications.

The first objective has been accomplished in Chapter 3, where all the required properties and theorems were established to facilitate the creation of CIT2 FSs. The formal definitions enable the practical creation of CIT2 FSs from predefined T1 FSs, by simply choosing the displacement interval. The creation of an open-source Java library (Chapter A in the Appendix), further simplifies the use of CIT2 FSs and FLSs by providing a free and customizable toolkit, available to the research community.

Objective 2, i.e. the generation of explainable systems from meaningful CIT2 FSs, has been extensively discussed in Chapters 3, 5 and 6. It has been shown how CIT2 FLSs are able to provide a meaningful explanation for the inference, type-reduction and defuzzification steps; this differs from traditional

IT2 FLSs with which it is challenging to produce a linguistic explanation for the type-reduction, as they process embedded sets that may not carry any semantic meaning in that specific context.

Chapter 4 fulfilled objective 3, regarding the definition of a practical inference framework. Specifically, the algorithm proposed in Chapter 4 is shown to have run-times that are comparable to some of the most popular type-reduction IT2 algorithms.

The last objective, has been partly achieved by Chapters 5 and 7. These two chapters, with some limitations discussed in detail in the next subsection, focus on the practical application of CIT2 FLSs, showing in which contexts they can be deployed and what the advantages of using them are. The interpretability of the CIT2 models is analyzed and compared to other approaches in the literature.

8.2 Limitations

The work carried out in this thesis, is subject to a number of limitations. One of the main ones concerns the lack of the interaction with the end-users to collect feedback about the interpretability of CIT2 FSs and the explanations produced by CIT2 FLSs. It will be essential to understand the real-world applicability of the novel approach, to gather opinions on the perceived usefulness and clarity of the explanations and models generated. These opinions should be collected not only from experts but also from people outside of this research area (e.g. physicians or engineers) for whom the intelligent system is designed.

Another limitation regards the use of CIT2 FLSs in a limited number of applications, focusing mainly on the medical domain and expert knowledge representation. To get a global understanding of the performances of the CIT2 approach in different scenarios, more case studies should be examined, from different domains and with different characteristics.

Lastly, this thesis addresses only one aspect regarding the interpretability

of fuzzy models and AI models in general. First of all, the definition of what makes a model interpretable or explainable is itself very *fuzzy*. There is no clear or formal agreement on what these concepts identify; their description is usually vague, based on natural language rather than mathematical formulas, making it hard to determine the "degree of interpretability" of a model. Furthermore, in the fuzzy logic field there are additional factors that need to be taken into account to model an interpretable system, such as the size of the rulebase or the structure of rules. They have not been studied as the contribution of this thesis is in the preservation of the meaningfulness of IT2 FSs when generated from T1 FSs and in the definition of an inference and defuzzification framework that facilitates the generation of explanation for system outputs.

8.3 Future Work

The work carried out in this thesis leaves room for additional research work that needs to be carried out in the future.

Surveys involving researches and end-users could be carried out to improve the quality of the explanations of CIT2 FLSs. The participants involved could provide valuable feedback on what are the main improvements to make in the current explanations and why. The process could be iterated multiple times, revising the design of the explanations at each step, until the participants are satisfied with the result.

The computational complexity of the *switch indices* approximation type-reduction approach presented in Chapter 4 could be improved. The identification of the switch indices, for now, is carried out using a brute force approach. However, determining a different stopping criterion or a direct way to identify them (similarly to what happens with the switch points in the KM procedure) would further improve the computational complexity of the novel procedure presented here. Additionally, it would be useful to carry out a formal study of the relation between the switch indices approach and the exhaustive method,

to understand if and under which circumstances they provide the same results.

The explanations for the type-reduction and defuzzification obtainable from CIT2 fuzzy systems, as shown in Chapter 5, can be further refined. Currently, the presence of technical terms such as the firing strengths and the schematic organization of the information may make the explanations harder to understand for a non-expert of the field. However, the material can be reorganized in a more natural piece of text, similarly to what happens for some T1 systems [11, 12] or with linguistic summaries [78, 79], to facilitate the understanding of the decision process by the end user. Furthermore, experts and non-experts of the field should be involved in these studies to have a better understanding of what aspects to improve, how to do it and why.

Lastly, the relation between the concept of meaningfulness and the novel and more flexible definitions for CIT2 fuzzy sets developed in Chapter 6, could be further studied. The models produced using the old CIT2 definitions, that make use of generator sets, and the new ones that model the meaningfulness through mathematical constraints could be compared to understand which ones are perceived as more interpretable and in which circumstances. Additionally, the definition of meaningful shapes through a set of contextual mathematical constraints is a concept that can be applied not only to CIT2 but to any kind of fuzzy set. This could represent a first step towards a more formal definition of the idea of a *meaningful fuzzy set*: a vague but recurring concept in fuzzy logic within the broad research area of XAI.

Bibliography

[1] J. M. Garibaldi and S. Guadarrama, "Constrained type-2 fuzzy sets," in *Advances in Type-2 Fuzzy Logic Systems (T2FUZZ), 2011 IEEE Symposium on*. IEEE, 2011, pp. 66–73.

[2] A. H. M. Pimenta and H. A. Camargo, "Interval type-2 fuzzy classifier design using genetic algorithms," in *International Conference on Fuzzy Systems*, July 2010, pp. 1–7.

[3] P. D'Alterio, J. M. Garibaldi, and R. John, "On the concept of meaningfulness in constrained type-2 fuzzy sets," in *International Conference on Fuzzy Systems (FUZZ-IEEE 2019)*, 2019.

[4] D. Castelvecchi, "Can we open the black box of ai?" *Nature News*, vol. 538, no. 7623, p. 20, 2016.

[5] J. M. Alonso, C. Castiello, and C. Mencar, *Interpretability of Fuzzy Systems: Current Research Trends and Prospects*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 219–237.

[6] B. Goodman and S. Flaxman, "European union regulations on algorithmic decision-making and a "right to explanation"," *AI Magazine*, vol. 38, no. 3, pp. 50–57, 2017.

[7] D. Gunning, "Explainable artificial intelligence (xai)," *Defense Advanced 894 Research Projects Agency, DARPA/I20 (DARPA)*, 2017.

[8] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338 – 353, 1965.

[9] L. A. Zadeh, "Fuzzy logic = computing with words," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 2, pp. 103–111, May 1996.

[10] A. Fernandez, F. Herrera, O. Cordon, M. Jose del Jesus, and F. Marcelloni, "Evolutionary fuzzy systems for explainable artificial intelligence: Why, when, what for, and where to?" *IEEE Computational Intelligence Magazine*, vol. 14, no. 1, pp. 69–81, Feb 2019.

[11] I. Baaj and J.-P. Poli, "Natural language generation of explanations of fuzzy inference decisions," in *International Conference on Fuzzy Systems (FUZZ-IEEE 2019)*, 2019.

[12] J. M. Alonso, A. Ramos-Soto, E. Reiter, and K. van Deemter, "An exploratory study on the benefits of using natural language for explaining

fuzzy rule-based systems," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, July 2017, pp. 1–6.

[13] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*. USA: Prentice-Hall, Inc., 1987.

[14] J. M. Mendel, "Uncertain rule-based fuzzy systems," in *Introduction and new directions*. Springer, 2017, p. 684.

[15] ——, "Computing with words, when words can mean different things to different people," in *Proc. of Third International ICSC Symposium on Fuzzy Logic and Applications*, 1999, pp. 158–164.

[16] L. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—i," *Information Sciences*, vol. 8, no. 3, pp. 199 – 249, 1975.

[17] J. M. Mendel, R. I. John, and F. Liu, "Interval type-2 fuzzy logic systems made simple," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 6, pp. 808–821, Dec 2006.

[18] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Information Sciences*, vol. 132, no. 1-4, pp. 195–220, 2001.

[19] D. Wu and J. M. Mendel, "Enhanced karnik–mendel algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 923–934, 2009.

[20] J. Aisbett, J. T. Rickard, and D. Morgenthaler, "Multivariate modeling and type-2 fuzzy sets," *Fuzzy Sets and Systems*, vol. 163, no. 1, pp. 78 – 95, 2011, theme: Classification and Modelling.

[21] D. Wu, "A constrained representation theorem for interval type-2 fuzzy sets using convex and normal embedded type-1 fuzzy sets, and its application to centroid computation," *Proceedings of World Conference on Soft Computing*, 2011.

[22] D. Wu and W. W. Tan, "A type-2 fuzzy logic controller for the liquid-level process," in *2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542)*, vol. 2, July 2004, pp. 953–958 vol.2.

[23] L. Amador-Angulo, O. Castillo, and M. Pulido, "Comparison of fuzzy controllers for the water tank with type-1 and type-2 fuzzy logic," in *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/-NAFIPS)*, June 2013, pp. 1062–1067.

[24] M. A. Sanchez, O. Castillo, and J. R. Castro, "Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems," *Expert Systems with Applications*, vol. 42, no. 14, pp. 5904 – 5914, 2015.

[25] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 fuzzy logic systems," *IEEE transactions on Fuzzy Systems*, vol. 7, no. 6, pp. 643–658, 1999.

[26] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: theory and design," *IEEE Transactions on Fuzzy systems*, vol. 8, no. 5, pp. 535–550, 2000.

[27] R. Sepúlveda, O. Castillo, P. Melin, A. Rodríguez-Díaz, and O. Montiel, "Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic," *Information Sciences*, vol. 177, no. 10, pp. 2023 – 2048, 2007, including Special Issue on Hybrid Intelligent Systems.

[28] P. A. S. Birkin and J. M. Garibaldi, "A comparison of type-1 and type-2 fuzzy controllers in a micro-robot context," in *2009 IEEE International Conference on Fuzzy Systems*, Aug 2009, pp. 1857–1862.

[29] S. Coupland, M. A. Gongora, R. John, and K. Wills, "A comparative study of fuzzy logic controllers for autonomous robots." in *IPMU 2006 Conference*, 2006.

[30] C. Lynch, H. Hagras, and V. Callaghan, "Embedded type-2 flc for real-time speed control of marine and traction diesel engines," in *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.*, May 2005, pp. 347–352.

[31] O. Castillo, L. Amador-Angulo, J. R. Castro, and M. Garcia-Valdez, "A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems," *Information Sciences*, vol. 354, pp. 257 – 274, 2016.

[32] A. H. M. Pimenta and H. d. A. Camargo, "Genetic interval type-2 fuzzy classifier generation: A comparative approach," *2010 Eleventh Brazilian Symposium on Neural Networks*, pp. 194–199, Oct 2010.

[33] E. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1 – 13, 1975.

[34] M. B. Gorzałczany and F. Rudziński, "Interpretable and accurate medical data classification – a multi-objective genetic-fuzzy optimization approach," *Expert Systems with Applications*, vol. 71, pp. 26 – 39, 2017.

[35] N. Potie, S. Giannoukakos, M. Hackenberg, and A. Fernandez, "On the need of interpretability for biomedical applications: Using fuzzy models for lung cancer prediction with liquid biopsy," in *International Conference on Fuzzy Systems (FUZZ-IEEE 2019)*, 2019.

[36] R. Khezri, R. Hosseini, and M. Mazinani, "A fuzzy rule-based expert system for the prognosis of the risk of development of the breast cancer," *INTERNATIONAL JOURNAL OF ENGINEERING*, 2014.

[37] J. M. Alonso, A. Ramos-Soto, C. Castiello, and C. Mencar, "Explainable ai beer style classifier." in *SICSA ReaLX*, 2018.

[38] C. Wagner, S. Miller, J. M. Garibaldi, D. T. Anderson, and T. C. Havens, "From interval-valued data to general type-2 fuzzy sets," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 2, pp. 248–269, 2015.

[39] J. M. Garibaldi, M. Jaroszewski, and S. Musikasuwan, "Nonstationary fuzzy sets," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 4, pp. 1072–1086, 2008.

[40] J. M. Garibaldi and T. Ozen, "Uncertain fuzzy reasoning: A case study in modelling expert decision making," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 1, pp. 16–30, 2007.

[41] J. M. Garibaldi, S.-M. Zhou, X.-Y. Wang, R. I. John, and I. O. Ellis, "Incorporation of expert variability into breast cancer treatment recommendation in designing clinical protocol guided fuzzy rule system models," *Journal of Biomedical Informatics*, vol. 45, no. 3, pp. 447 – 459, 2012.

[42] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.

[43] J. M. Mendel and R. John, "Footprint of uncertainty and its importance to type-2 fuzzy sets," in *Proceedings 6th IASTED Int'l. Conf. on Artificial Intelligence and Soft Computing (ASC 2002)*, July 2002, pp. 587 – 592.

[44] J. M. Mendel and R. I. John, "Type-2 fuzzy sets made simple," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 117–127, Apr 2002.

[45] S. Greenfield, F. Chiclana, R. John, and S. Coupland, "The sampling method of defuzzification for type-2 fuzzy sets: Experimental evaluation," *Information Sciences*, vol. 189, pp. 77 – 92, 2012.

[46] S. Greenfield, R. John, and S. Coupland, "A novel sampling method for type-2 defuzzification," in *Proc. UKCI 2005*, 09 2005, pp. 120–127.

[47] F. Liu, "An efficient centroid type-reduction strategy for general type-2 fuzzy logic system," *Information Sciences*, vol. 178, no. 9, pp. 2224 – 2236, 2008.

[48] J. M. Mendel, F. Liu, and D. Zhai, "*alpha*-plane representation for type-2 fuzzy sets: Theory and applications," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 5, pp. 1189–1207, Oct 2009.

[49] C. Wagner and H. Hagras, "Toward general type-2 fuzzy logic systems based on zslices," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 4, pp. 637–660, Aug 2010.

[50] R. John, "Perception modelling using type-2 fuzzy sets." Ph.D. dissertation, De Montfort Univerisity, 2000.

[51] J. M. Mendel, "Explaining the performance potential of rule-based fuzzy systems as a greater sculpting of the state space," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 4, pp. 2362–2373, 2018.

[52] D. Wu, H. Zhang, and J. Huang, "A constrained representation theorem for well-shaped interval type-2 fuzzy sets, and the corresponding constrained uncertainty measures," *IEEE Transactions on Fuzzy Systems*, pp. 1–1, 2018.

[53] H. Hagras, "Developing a type-2 flc through embedded type-1 flcs," in *Fuzzy Systems, 2008. FUZZ-IEEE 2008.(IEEE World Congress on Computational Intelligence). IEEE International Conference on.* IEEE, 2008, pp. 148–155.

[54] D. Castelvecchi, "Can we open the black box of ai?" *Nature News*, vol. 538, no. 7623, p. 20, 2016.

[55] J. M. Mendel, "Computing with words and its relationships with fuzzistics," *Information Sciences*, vol. 177, no. 4, pp. 988 – 1006, 2007.

[56] A. Norwich and I. Turksen, "A model for the measurement of membership and the consequences of its empirical implementation," *Fuzzy Sets and Systems*, vol. 12, no. 1, pp. 1 – 25, 1984.

[57] S. Greenfield, R. John, and S. Coupland, "A novel sampling method for type-2 defuzzification," in *Proc. UKCI 2005*, 09 2005, pp. 120–127.

[58] O. Cordon, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, "Ten years of genetic fuzzy systems: current framework and new trends," *Fuzzy Sets and Systems*, vol. 141, no. 1, pp. 5 – 31, 2004.

[59] A. Fernandez, F. Herrera, O. Cordon, M. J. del Jesus, and F. Marcelloni, "Evolutionary fuzzy systems for explainable artificial intelligence: Why, when, what for, and where to?" *IEEE Computational Intelligence Magazine*, vol. 14, no. 1, pp. 69–81, 2019.

[60] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2, pp. 191 – 203, 1984.

[61] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[62] D. Coomans, I. Broeckaert, M. Jonckheer, and D. L Massart, "Comparison of multivariate discrimination techniques for clinical data— application to the thyroid functional state," *Methods of information in medicine*, vol. 22, pp. 93–101, 05 1983.

[63] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework." *Journal of Multiple-Valued Logic & Soft Computing*, vol. 17, 2011.

[64] C. Wagner, "Juzzy - a java based toolkit for type-2 fuzzy logic," in *2013 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ)*, April 2013, pp. 45–52.

[65] P. D'Alterio, J. M. Garibaldi, R. John, and A. Pourabdollah, "Constrained interval type-2 fuzzy sets," *IEEE Transactions on Fuzzy Systems*, 2020.

[66] C. Chen, D. Wu, J. M. Garibaldi, R. I. John, J. Twycross, and J. M. Mendel, "A comprehensive study of the efficiency of type-reduction algorithms," *IEEE Transactions on Fuzzy Systems*, pp. 1–1, 2020.

[67] P. D'Alterio, J. M. Garibaldi, R. I. John, and C. Wagner, "Juzzy constrained: Software for constrained interval type-2 fuzzy sets and systems in Java," in *2020 IEEE World Congress on Computational Intelligence (WCCI 2020)*, July 2020.

[68] P. D'Alterio, J. M. Garibaldi, and A. Pourabdollah, "Exploring constrained type-2 fuzzy sets," in *2018 IEEE World Congress on Computational Intelligence (WCCI 2018)*, July 2018.

[69] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, 1985.

[70] P. D'Alterio, J. M. Garibaldi, R. John, and C. Wagner, "A fast inference and type-reduction process for constrained interval type-2 fuzzy system," *IEEE Transactions on Fuzzy Systems*, pp. 1–1, 2020.

[71] R. Likert, "A technique for the measurement of attitudes." *Archives of psychology*, 1932.

[72] K. J. Wallace, C. Wagner, and M. J. Smith, "Eliciting human values for conservation planning and decisions: a global issue," *Journal of Environmental Management*, vol. 170, pp. 160–168, 2016.

[73] F. Liu and J. M. Mendel, "An interval approach to fuzzistics for interval type-2 fuzzy sets," in *2007 IEEE International Fuzzy Systems Conference*, 2007, pp. 1–6.

[74] D. Wu, J. M. Mendel, and S. Coupland, "Enhanced interval approach for encoding words into interval type-2 fuzzy sets and its convergence analysis," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 3, pp. 499–513, 2012.

[75] T. C. Havens, C. Wagner, and D. T. Anderson, "Efficient modeling and representation of agreement in interval-valued data," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017, pp. 1–6.

[76] J. McCulloch, Z. Ellerby, and C. Wagner, "On comparing and selecting approaches to model interval-valued data as fuzzy sets," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2019, pp. 1–6.

[77] J. McCulloch, "Fuzzycreator: A python-based toolkit for automatically generating and analysing data-driven fuzzy sets," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017, pp. 1–6.

[78] D. Anderson, R. H. Luke, J. M. Keller, M. Skubic, M. Rantz, and M. Aud, "Linguistic summarization of video for fall detection using voxel person and fuzzy logic," *Computer vision and image understanding*, vol. 113, no. 1, pp. 80–89, 2009.

[79] M. Ros, M. Pegalajar, M. Delgado, A. Vila, D. T. Anderson, J. M. Keller, and M. Popescu, "Linguistic summarization of long-term trends for understanding change in human behavior," in *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, 2011, pp. 2080–2087.

[80] A. Taskin and T. Kumbasar, "An open source matlab/simulink toolbox for interval type-2 fuzzy logic systems," in *2015 IEEE Symposium Series on Computational Intelligence*, Dec 2015, pp. 1561–1568.

[81] O. Castillo and P. Melin, "Computational intelligence software: Type-2 fuzzy logic and modular neural networks," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, June 2008, pp. 1820–1827.

[82] J. R. Castro, O. Castillo, and P. Melin, "An interval type-2 fuzzy logic toolbox for control applications," in *2007 IEEE International Fuzzy Systems Conference*, July 2007, pp. 1–6.

[83] J. R. Castro, O. Castillo, P. Melin, and A. Rodríguez-Díaz, *Building Fuzzy Inference Systems with a New Interval Type-2 Fuzzy Logic Toolbox*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 104–114.

[84] J. McCulloch, "Fuzzycreator: A python-based toolkit for automatically generating and analysing data-driven fuzzy sets," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2017, pp. 1–6.

[85] C. Wagner, S. Miller, and J. M. Garibaldi, "A fuzzy toolbox for the r programming language," in *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, June 2011, pp. 1185–1192.

[86] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, p. 10–18, Nov. 2009.

[87] "IEEE standard for fuzzy markup language," *IEEE Std 1855-2016*, pp. 1–89, May 2016.

[88] J. M. Soto-Hidalgo, J. M. Alonso, G. Acampora, and J. Alcala-Fdez, "JFML: A java library to design fuzzy logic systems according to the ieee std 1855-2016," *IEEE Access*, vol. 6, pp. 54 952–54 964, 2018.

[89] G. Acampora, J. Alcala-Fdez, R. Siciliano, J. M. Soto-Hidalgo, and A. Vitiello, "VisualJFML: A visual environment for designing fuzzy systems according to ieee std 1855-2016," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, June 2019, pp. 1–6.

[90] C. Wagner, M. Pierfitt, and J. McCulloch, "Juzzy online: An online toolkit for the design, implementation, execution and sharing of type-1 and type-2 fuzzy logic systems," *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 2321–2328, 2014.

[91] C. Wagner and H. Hagras, "Toward general type-2 fuzzy logic systems based on zslices," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 4, pp. 637–660, Aug 2010.

Appendix A

Juzzy Constrained: a Java Library for CIT2 Fuzzy Sets and Systems

A.1 Introduction

Although many practical applications of CIT2 FLS have already been shown in Chapter 3 and Chapter 4, there is no library that can be used by the research community to easily deploy CIT2 FLS. The aim of this chapter is to present the first software library, named *Juzzy Constrained*, that implements CIT2 fuzzy sets and systems to favour their use in the research community. Written in Java, *Juzzy Constrained* is an extension of the already well-known Java library *Juzzy* [64] and follows its conventions to facilitate its use for developers. The new toolkit makes possible the design of CIT2 FLS using the defuzzification algorithms proposed in Chapter 3 and Chapter 4 and is capable of using the constrained representation to provide human-readable explanations for the constrained interval centroids produced by the systems.

The rest of the chapter is organized as follows: after a short description of other software libraries focused on T1 and T2 fuzzy logic (Sec. A.2), the new library *Juzzy Constrained* will be analyzed, describing its structure, its main classes and its relation with *Juzzy* (Sec. A.4). Finally, a working example will be presented: a CIT2 FLS will be built from scratch, with the help of code snippets to facilitate the understanding of the usage of the toolkit (Sec. A.6).

A.2 Related works

Many tools for the development of T1 and T2 FLS have been released over the years. One of the most famous ones is the Fuzzy Logic Toolbox for MATLAB[1]. It allows developers to design T1 fuzzy sets and systems through a set of functions or the use of a graphical interface. The sets built with the toolbox, the control surfaces and the rules can then be easily visualized. Similar toolboxes that include IT2 fuzzy sets have been proposed in [80–83].

Software libraries for different programming languages have also been released. In [84] a Python toolkit for the automatic generation of T1, IT2 and T2 fuzzy sets from data has been presented; [85], instead, describes a software library in R for the modelling of T1, IT2 and T2 FLS that also includes functions for the graphical visualization of fuzzy sets and control surfaces.

Software for the creation of fuzzy systems has also been included in famous suites for machine learning such as KEEL [63] and Weka [86]. Both offer various methods to learn fuzzy rules and sets from data (e.g. with the use of genetic algorithms) and to perform fuzzy clustering.

After the introduction of the IEEE Standard for Fuzzy Markup Language for the definition of fuzzy sets and systems in a "human-readable and hardware independent way" [87], new software libraries adhering to the novel standard have been developed such as JFML [88] and VisualJFML [89].

A.3 Juzzy

Juzzy Online [90], is a software for the design, execution and sharing of T1 and T2 fuzzy sets and systems through the use of an online dashboard that is usable with no knowledge of programming.

A Java version of the same tool has been released. Juzzy [64] is a library for the implementation of T1, IT2 and T2 fuzzy sets and systems. It is written in Java, it is open-source and available online at http://juzzy.wagnerweb.net/, http://www.lucidresearch.org/software.html and on the Maven Central Repository[2]. The toolkit implements T2 fuzzy sets with the zSlices representation [91] and also supports multi-core execution of the code.

A.4 Juzzy Constrained

The Java library for CIT2 fuzzy sets and systems has been conceived as an extension of Juzzy: it makes use of its T1 membership functions to define the generator sets of CIT2 fuzzy sets and also adopts some of its conventions (e.g. for the creation of rules) and utility classes (such as `Input` and `Output` to model the input and output variables of a CIT2 FLS). Therefore, for Juzzy Constrained to work, also Juzzy must be included in the given Java project.

The source-code released under the BSD 3-Clause license, the documentation and the JAR archive of Juzzy Constrained are freely available on GitHub at https://github.com/PasqualeDAlterio/JuzzyConstrained. The library is

[1] https://www.mathworks.com/products/fuzzy-logic.html
[2] https://search.maven.org/artifact/com.github.chwagnlucid/Juzzy/2.0/jar

also available on the Maven Central Repository and can be quickly included in any Java Maven project (for more information, see the GitHub page).

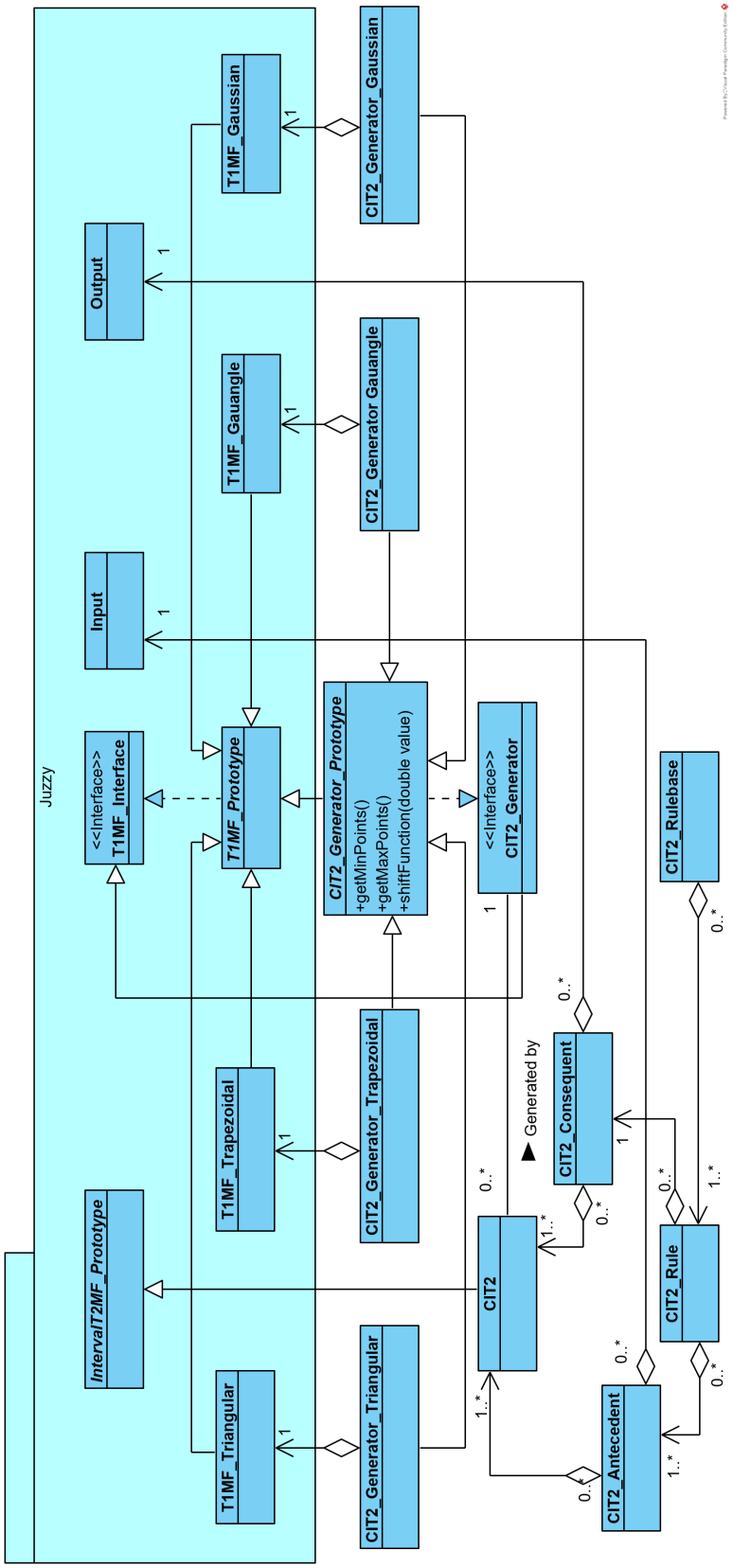


Figure A.1: *The class diagram of Juzzy Constrained*

A.4.1 Library structure

The packages included in the library are shown in Fig. A.2. The package `CIT2_Generator` includes the T1 membership functions that are usable as generator sets. Triangular, Gaussian, Gauangle and trapezoidal membership functions are currently supported. Each one of them is a wrapper of the correspondent `T1MF` defined in Juzzy; the main difference between them is that each of the `CIT2_Generator` must implement a method that returns all the points of local (and global) maximum and a method that returns all points of local minimum of the membership function, as requested by the `CIT2_Generator` interface. These two methods are needed to determine the upper and lower membership functions of the CIT2 fuzzy set, as discussed in Theorem A.1, in the Appendix.

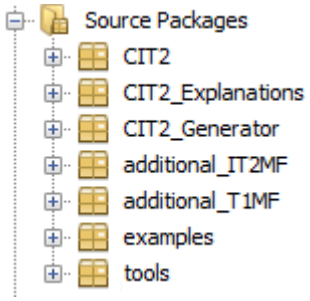


Figure A.2: *The package structure of the library*

The `CIT2` package is the core of the library: it implements CIT2 fuzzy sets and systems using the same style used by Juzzy. Once the sets have been defined they can be used to build antecedents and consequents that are then organized in `CIT2_Rule`. Once the rules are created, they can be organized in a `CIT2_Rulebase` to implement a FLS.

`CIT2_Explanations` contains all the objects that are used in the generation of the explanation of the output of a `CIT2_Rulebase`. They mostly focus on organizing and formatting information in a humanly readable piece of text in order to show how the endpoints of the constrained centroid have been obtained. The remaining packages offer additional utilities and tools that were not originally implemented in Juzzy but are useful in the Juzzy Constrained context (e.g. an IT2 fuzzy set where the upper and lower bounds can have arbitrary shape or a T1 fuzzy set modeling the result of the inference operation).

Fig. A.1 shows the class diagrams with all the main classes used in Juzzy Constrained and their relation with the original classes in Juzzy. Each `CIT2` fuzzy set, to be instantiated, needs a `CIT2_Generator`. This interface extends the `T1MF_Interface` defined in Juzzy, since the generator set is a T1 set, and requires the implementation of three additional methods: `getMaxPoints`, `getMinPoints` and `shiftFunction`. The first two, as described earlier in this section, are needed to determine the boundary functions of the generated CIT2 fuzzy; the shifting method, instead, is needed to generate the acceptable embedded sets: since they are translations along the x-axis of the generator set, this method takes a real number `value` as an argument and returns a new T1 membership function representing the generator set shifted by `value`. Additionally, since CIT2 fuzzy sets are a special case of IT2 fuzzy sets, i.e. they have been obtained by adding a set of additional mathematical constrained

to the original IT2 definition, the class `CIT2` extends the Juzzy abstract class `IntervalT2MF_Prototype`.

The generator sets implemented in the library extend `CIT2_Generator_Prototype`, i.e. an abstract class that already implements some functionalities that are used by all the generators provided. To add a new generator membership function, it is only required to implement the `CIT2_Generator` interface. This operation, as it will be shown in Sec. A.6, is straightforward for all the widely used T1 membership functions.

All the classes related to the the construction of a rule and a rule-base follow the same conventions used in Juzzy, making them easy to work with for the developers that are already used to the T1, IT2 and T2 rule-bases of the original library.

A.4.2 Defuzzification algorithms, other features and limitations

The toolkit provides two algorithms for the defuzzification of the output of a CIT2 FLS. The first one, implemented by the method `sampleCentroid` in the class `CIT2_Rulebase`, is based on the *sampling approach* proposed in Chapter 3, itself an adaptation for CIT2 sets of the *sampling method* for T2 fuzzy sets [57]. Since the extensive computation of the centroid by processing all the acceptable embedded sets has a prohibitive cost (similarly to what happens with "standard" T2 fuzzy sets) and each of these embedded sets only gives a small contribution to the final result, the idea is to calculate an approximation by sampling a subset of the acceptable embedded sets and use only them to compute the constrained centroid.

The other defuzzification algorithm included in Juzzy Constrained is the one presented in Chapter 4, based on the concept of *switch indices* instead of the *switch points* used by the KM procedure for IT2 fuzzy sets [18]. This approximation method is faster than the sampling one as it uses the properties of CIT2 fuzzy sets to quickly identify the small subset of acceptable embedded sets that will be used to determine the constrained centroid. For more details about this algorithm, please refer to Chapter 4. This approach can also be used to produce human-readable explanations for CIT2 FLSs as shown in Chapter 5 and in Sec. A.6.

In addition to the methods implemented in Juzzy for the visualization of T1 and IT2 fuzzy sets, Juzzy Constrained integrates the popular Java graphical library JFreeChart[3]. This represents a more flexible way of building plots, since they are easily and widely customizable, while also giving the opportunity of better highlight the FOUs of the CIT2 and IT2 fuzzy sets, as shown in Fig. A.3.

Being currently still under development, Juzzy Constrained has some limitations. Specifically, `CIT2_Rule` only implements the *and* operator in the antecedent composition and does so with the *min T-Norm*. In addition to that, each rule can currently has only one consequent. At the moment, this limitation can be overcome by replacing a rule with n consequents with n replicas of the rule, one per consequent. In future works, the library will be

[3] http://www.jfree.org/jfreechart/

expanded by adding the support to multiple-consequent rules and more antecedent connectors.

A.5 Determining the boundary functions of a CIT2 fuzzy set

The `CIT2_Generator` interface used in this library requires a method that returns all the points of local maximum and one that returns all the points of local minimum of a membership function for it to be used as a generator set. The reason why these points are needed is to easily determine the boundary functions of the generated CIT2 fuzzy set. As shown in Chapter 3, these two membership functions for a generic CIT2 fuzzy set $\breve{A}$ can be expressed as:

$$\overline{\mu}_{\breve{A}}(x) = \sup_{S \in \text{CAES}_{\breve{A}}} \mu_S(x) \tag{A.1}$$

$$\underline{\mu}_{\breve{A}}(x) = \inf_{S \in \text{CAES}_{\breve{A}}} \mu_S(x) \tag{A.2}$$

where $\breve{A}$ is a CIT2 set and the CAES is the collection of its acceptable embedded sets. The following theorem proves that to determine the upper and lower bounds of the FOU of a CIT2 fuzzy set, it is sufficient to know the generator set, its points of local minimum and maximum and the displacement interval used.

Theorem A.1. *Given a CIT2 fuzzy set $\breve{A}$, to determine its upper membership function $\overline{\mu}_{\breve{A}}$ it is sufficient to know the T1 generator set G (with a continuous membership function) its displacement interval $[a, b]$ with $a \leq 0, b \geq 0, a, b \in \mathbb{R}$ and the set M of all the local points of maximum of μ_G.*

Proof. To prove the theorem, it will be shown that the upperbound function of $\breve{A}$ $\overline{\mu}_{\breve{A}}$ can be expressed as:

$$\overline{\mu}_{\breve{A}}(x) = \begin{cases} \max\left(\star, \max_{k \in M}(\mu_G(k))\right) & M \neq \emptyset \\ \star & \text{otherwise} \end{cases} \tag{A.3}$$

where M is the set of all the local points of maximum of μ_G in $[x - b, x - a]$ and $\star$ is:

$$\star = \max\Big(\mu_G(x - a), \mu_G(x - b)\Big) \tag{A.4}$$

Since each S in (A.1) is obtained as a shifting of G using the values in the displacement interval (see [65] for more details), it can be rewritten as:

$$\overline{\mu}_{\breve{A}}(x) = \max_{z \in [a,b]} \mu_G(x - z) \tag{A.5}$$

Using (A.5), the upperbound membership function (A.3) can be rewritten as:

$$\max_{z\in[a,b]} \mu_G(x-z) = \begin{cases} \max\left(\star, \max_{k\in M}(\mu_G(k))\right) & M \neq \emptyset \\ \star & \text{otherwise} \end{cases} \tag{A.6}$$

At this point, it must be proved that the upperbound membership function (A.5) is determined either by $\mu_G(x-a)$ and $\mu_G(x-b)$ or by one of the points of maximum of μ_G in the interval $[x-b,\, x-a]$, i.e. by one of the points in M. To do so, the two possible scenarios must be considered:

1. $\max_{z\in[a,b]} \mu_G(x-z) = \star$ (A.7)

2. $\max_{z\in[a,b]} \mu_G(x-z) \neq \star$ (A.8)

In (1), it is assumed that the upperbound membership function is determined by maximum between $\mu_G(x-a)$ and $\mu_G(x-b)$. In this case, (A.3) trivially holds, both when M is empty and when it contains at least one element. In (2), instead, it must be proved that when the upperbound membership function is not determined by $\mu_G(x-a)$ and $\mu_G(x-b)$, then it is be determined by one of the points of maximum in M. In fact, since the upperbound membership degree of x is different from both $\mu_G(x-a)$ and $\mu_G(x-b)$, it must be determined by another value w that is different from $x-a$ and $x-b$. Formally:

$$\exists\, w \in (x-b,\, x-a) : \forall z \in [a,b],\, \mu_G(w) \geq \mu_G(x-z) \tag{A.9}$$

By definition, w is a point of local maximum in $[x-b, x-a]$ and must therefore be equal to the maximum $k \in M$ in (A.6) when $M \neq \emptyset$. Therefore the thesis holds in 2) as well. Since (A.6) holds in all the possible cases, it is true. □

Similarly, it can be proven that to determine the lowerbound membership function of a CIT2 fuzzy set it is sufficient to know the generator set, its points of local minimum and the displacement interval used.

A.6 Applications and examples

This section will show how Juzzy Constrained can be used in practice to develop CIT2 FLS, starting from the creation of CIT2 fuzzy sets and then illustrating how they can be put together to make rules and rulebases.

The example analyzed in this chapter is the *tipping problem*. This system has been chosen for its simplicity and *not* to show the full potential of CIT2 FLSs. A more thorough analysis of the advantages of the use of CIT2 FLSs and case studies on real world datasets can be found in Chapter 5. The tipping problem has the following structure: it has 2 input variables, food and service, and the goal is to use them to determine the adequate percentage to give as tip.

The first thing to do, is to instantiate the generator sets. Their creation is identical to the creation of T1 fuzzy sets in Juzzy. Here there is an example of how the generator sets for the *service* membership functions can be created.

```
T1MF_Generator_Gauangle unfriendlyServiceMF=
    new T1MF_Generator_Gauangle("Unfriendly",0.0, 0.0, 6);
unfriendlyServiceMF.setLeftShoulder(true);
T1MF_Generator_Gauangle okServiceMF =
    new T1MF_Generator_Gauangle("OK",2.5, 5.0, 7.5);
T1MF_Generator_Gauangle friendlyServiceMF =
    new T1MF_Generator_Gauangle("Friendly",4, 10, 10);
friendlyServiceMF.setRightShoulder(true);
```

With the generator sets, it is possible to create CIT2 fuzzy sets. In addition to the generator sets, also the displacement intervals need to be specified. They determine how "wide" the shifting and therefore the FOU will be. In the example below, the positive shifting values `shifting_size_2` is used to generate the displacement interval `[-shifting_size_2, shifting_size_2]`.

```
CIT2 cit2_unfriendlyServiceMF =  new CIT2(unfriendlyServiceMF.getName
    (), unfriendlyServiceMF, shifting_size_2);
CIT2 cit2_okServiceMF = new CIT2(okServiceMF.getName(), okServiceMF,
    shifting_size_2);
CIT2 cit2_friendlyServiceMF = new CIT2(friendlyServiceMF.getName(),
    friendlyServiceMF, shifting_size_2);
```

The definition of the input and output variables, is taken from Juzzy since it uses the same `Input` and `Output` objects.

```
Input food = new Input("Food Quality", new Tuple(0,10));
Input service =new Input("Service Level", new Tuple(0,10));
Output tip = new Output("Tip", new Tuple(0,30));
```

The partitioning of the variables can then be plotted using JFreeChart as shown below. The results of this operation for the food, service and tip are shown respectively in Fig. A.3, Fig. A.4 and Fig. A.5.

```
JFreeChartPlotter.plotMFs("Food partitioning", new CIT2[]{
    cit2_badFoodMF, cit2_greatFoodMF},food.getDomain(), 1000);
JFreeChartPlotter.plotMFs("Service partitioning", new CIT2[]{
    cit2_friendlyServiceMF, cit2_okServiceMF, cit2_unfriendlyServiceMF
    }, service.getDomain(), 1000);
JFreeChartPlotter.plotMFs("Tip partitioning", new CIT2[]{cit2_lowTipMF
    , cit2_mediumTipMF, cit2_highTipMF}, tip.getDomain(), 1000);
```

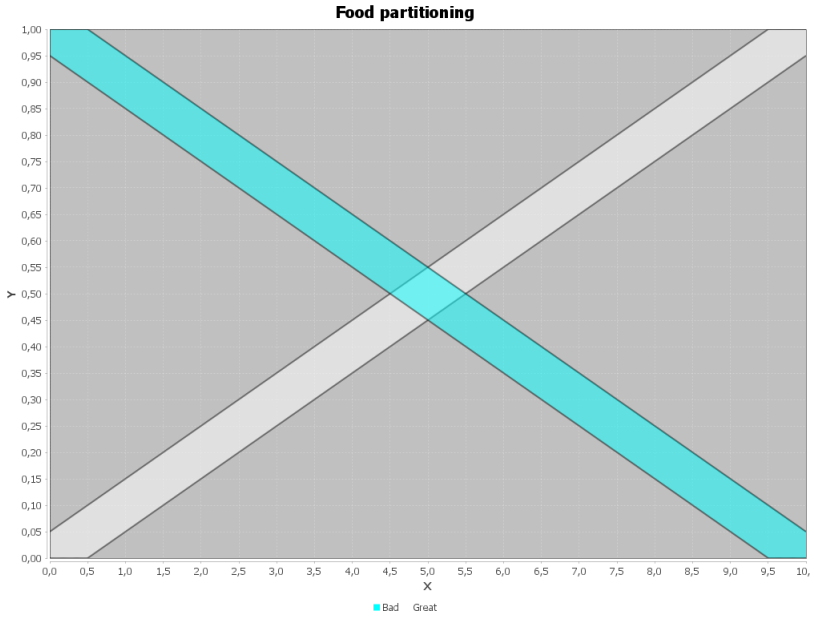


Figure A.3: *Partitioning of the* food *variable (from left to right: Bad, Great)*

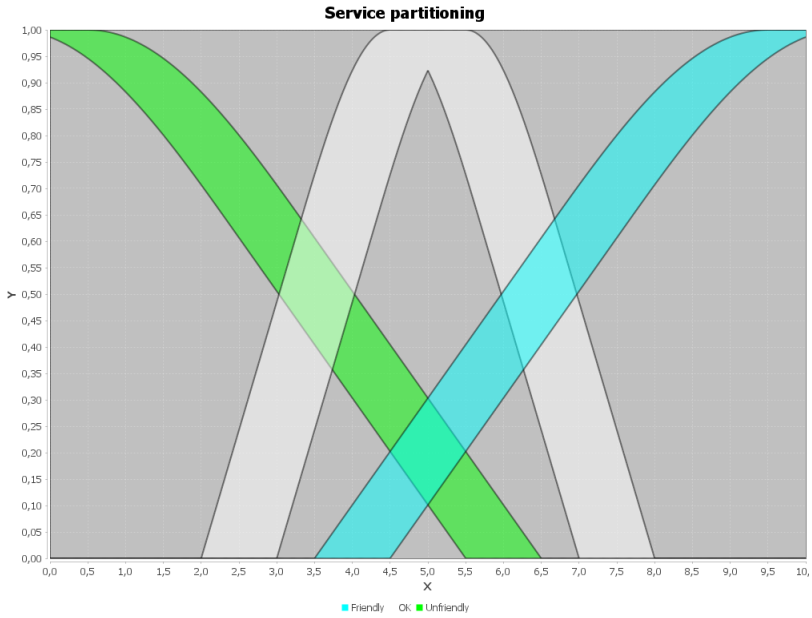


Figure A.4: *Partitioning of the* service *variable (from left to right: Friendly, Ok, Unfriendly)*

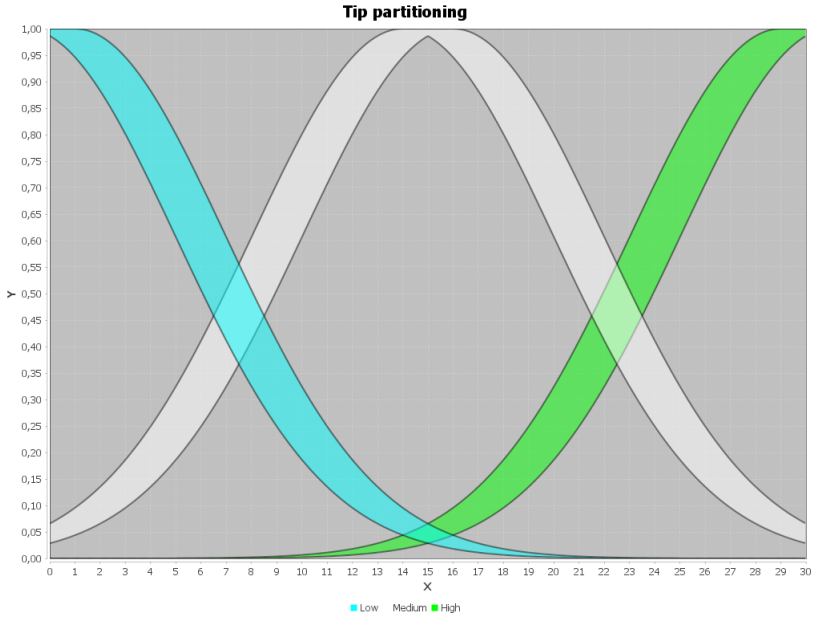


Figure A.5: *Partitioning of the* tip *variable (from left to right: Low, Medium, High)*

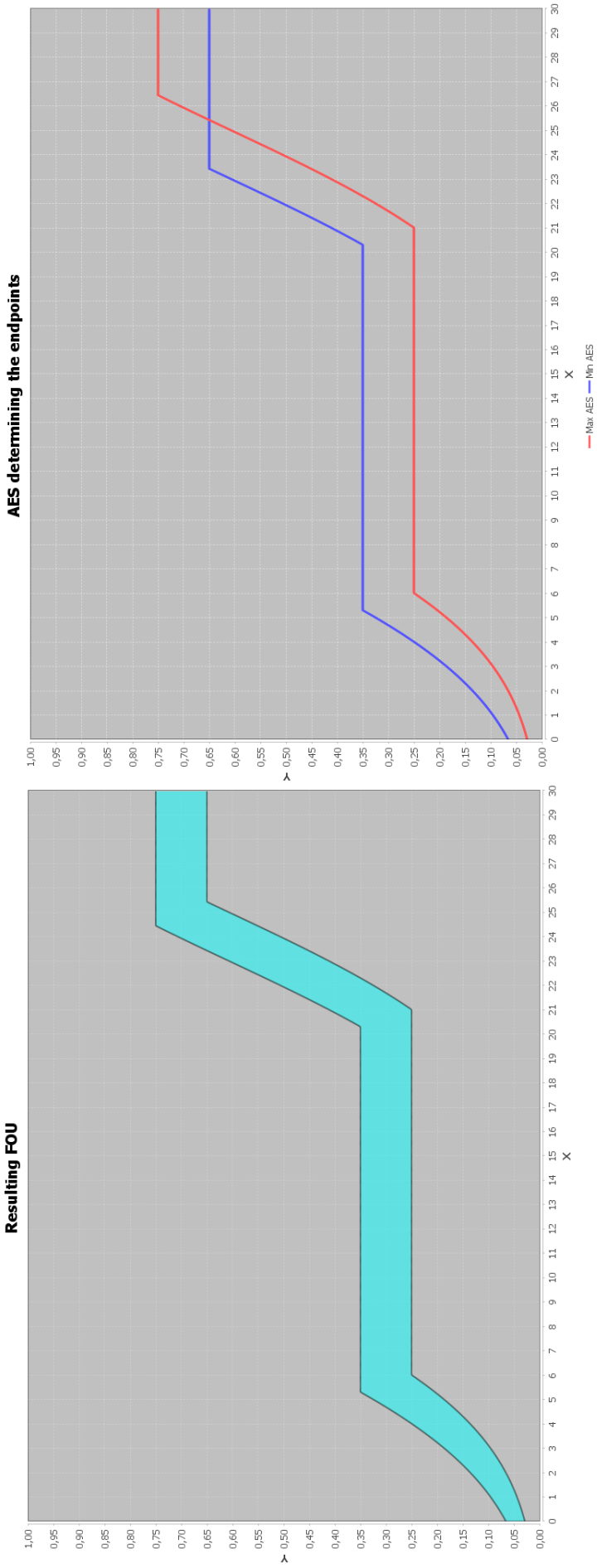


Figure A.6: *FOU obtained from the inference (on the left) and the acceptable embedded sets determining the endpoints of the constrained centroid (on the right)*

Once the CIT2 fuzzy sets have been defined, they can be paired with the input and output variables to define the antecedents and the consesequents that will be used in the rulebase. In this case, Juzzy Constrained follows the same conventions used by Juzzy, making the creation of `CIT2_Antecedent` and `CIT2_Consequent` very similar to the creation of IT2 antecedents and consequents in the original library.

```
CIT2_Antecedent unfriendlyService =
    new CIT2_Antecedent(cit2_unfriendlyServiceMF, service);
CIT2_Antecedent okService =
    new CIT2_Antecedent(cit2_okServiceMF, service);
CIT2_Antecedent friendlyService =
    new CIT2_Antecedent(cit2_friendlyServiceMF, service);
```

Once the antecedents and consequents have been defined, they can be put together to create the rulebase. Again, the initialization of a CIT2 rulebase is very similar to the creation of T1 and IT2 rulebases in Juzzy.

```
CIT2_Rulebase rulebase = new CIT2_Rulebase();
rulebase.addRule(new CIT2_Rule(new CIT2_Antecedent[]{badFood,
    unfriendlyService}, lowTip));
rulebase.addRule(new CIT2_Rule(new CIT2_Antecedent[]{badFood,
    okService}, lowTip));
rulebase.addRule(new CIT2_Rule(new CIT2_Antecedent[]{badFood,
    friendlyService}, mediumTip));
rulebase.addRule(new CIT2_Rule(new CIT2_Antecedent[]{greatFood,
    unfriendlyService}, lowTip));
rulebase.addRule(new CIT2_Rule(new CIT2_Antecedent[]{greatFood,
    okService}, mediumTip));
rulebase.addRule(new CIT2_Rule(new CIT2_Antecedent[]{greatFood,
    friendlyService}, highTip));
```

After the input values are set, there are two algorithms that can be used to do the inference and defuzzify the result: the sampling strategy (Chapter 3) and the switch index method (Chapter 4). In the first case, the algorithm can be executed invoking the method `rulebase.samplingDefuzzification(50)` where 50 is the number of samples used to compute the constrained centroid. The function returns a `Tuple` representing the centroid.

```
food.setInput(7);
service.setInput(8);
Tuple constrained_centroid_sampling=
    rulebase.samplingDefuzzification(50);
Tuple constrained_centroid_si=
    rulebase.switchIndexDefuzzification(100);
ExplainableCentroid result=
    rulebase.explainableDefuzzification(100);
```

The switch index approach, instead, can be used in two different ways: using the method `rulebase.switchIndexDefuzzification(100)` where 100 is the level of discretization used to defuzzify the acceptable embedded sets, the library returns a `Tuple` containing the value of the constrained centroid, just like in the sampling method case; the method `rulebase.explainableDefuzzification(100)`, instead, returns the constrained centroid and the explanation for its generation in an `ExplainableCentroid` object.
As already discussed in Chapter 3 and 4, the properties of CIT2 fuzzy sets can be used to link the endpoints of the centroid to the specific acceptable embedded sets that generated them. They can then be used to determine which rules and input values led to the creation of the constrained interval centroid, in order to create a human-readable explanation. The selected acceptable em-

bedded sets also have an interpretable structure: the consequent membership functions that contributed to their generation are clearly visible and so are the firing strengths of the rules they belong to (i.e. the heights at which they have been "truncated"). For other IT2 defuzzification procedures like the KM one, on the other hand, there is no guarantee that the chosen embedded sets will have any *meaningful* shape nor that it is possible to link them directly to the rules to produce an explanation. The ability to provide interpretable results when computing the constrained interval centroid is one of the reasons why CIT2 fuzzy sets can represent a valuable alternative to IT2 fuzzy sets in the context of XAI.

Once the `ExplainableCentroid` object is obtained, the acceptable embedded sets determining the constrained centroid can be plotted as shown below, together with the fired FOU. The plots for this example are show in Fig. A.6.

```
JFreeChartPlotter.plotMFs("Resulting FOU", new IntervalT2MF_Interface
    []{rulebase.getFiredFOU()}, tip.getDomain(), 1000);
JFreeChartPlotter.plotMFs("AES determining the endpoints", new
    T1MF_Interface[]{left_aes, right_aes}, tip.getDomain(), 1000);
System.out.println("The recommended tip percentage is in the range:"+
    result.getIntervalCentroid());
//Print the explanations
System.out.println(result.printableExplanation());
```

The `ExplainableCentroid` structure also stores the information necessary for the creation of the human-readable explanation using the method `result.printableExplanation()`. The piece of text below, links each of the endpoints of the constrained centroid to the rules in the rulebase that generated them, also showing the firing values of the rules, the input values and their membership degrees with respect to the antecedent membership functions.

```
The recommended tip percentage is in the range: left = 18.06 and right = 19.67
The leftmost centroid (18.06) is obtained from firing the following rules:
Medium: 0.35 obtained because Food Quality IS Bad [0.25, 0.35] AND Service Level IS
     Friendly [0.71, 0.88] using the UPPER membership degree of each input terms
High: 0.65 obtained because Food Quality IS Great [0.65, 0.75] AND Service Level IS
     Friendly [0.71, 0.88] using the LOWER membership degree of each input terms

The rightmost centroid (19.67) is obtained from firing the following rules:
Medium: 0.25 obtained because Food Quality IS Bad [0.25, 0.35] AND Service Level IS
     Friendly [0.71, 0.88] using the LOWER membership degree of each input terms
High: 0.75 obtained because Food Quality IS Great [0.65, 0.75] AND Service Level IS
     Friendly [0.71, 0.88] using the UPPER membership degree of each input terms
```

A.7 Adding a new CIT2 generator membership function

Juzzy Constrained currently supports 4 types of generator sets: Gaussian, Gauangle, triangular and trapezoidal. To add additional shapes, it is necessary to define a new class that implements the `CIT2_Generator` interface. The new class needs to provide methods that return the points of minimum and maximum of the membership function (so that the FOU of the CIT2 fuzzy set can be determined, see Theorem A.1, in the Appendix) and a method for the shifting of the generator set (to generate the acceptable embedded sets).

Although implementing the methods that determine the points of minimum and maximum may seem challenging, it is relatively easy for many shapes. In the code snippet below, the implementation of these method is shown for the trapezoidal membership function.

```
@Override
protected ArrayList<Interval> computeMinPoints()
{
   ArrayList<Interval> min_points=new ArrayList<>();
    min_points.add(new Interval(trapezoid.getA()));
    min_points.add(new Interval(trapezoid.getD()));
    return min_points;
}

@Override
protected ArrayList<Interval> computeMaxPoints()
{
    ArrayList<Interval> max_points=new ArrayList<>();
    max_points.add(new Interval(trapezoid.getB(), trapezoid.getC()));
    return max_points;
}
```

The minimum and maximum points are stored in `Interval` objects which store generic intervals of the form $[a, b]$. The reason why intervals are used rather than points is that in some membership functions the points of minimum or maximum are infinite and all within a given interval. For example, in the case of a trapezoidal membership function, the points of maximum are all the points that make the shorter base, i.e. all the points in the segment $\overline{BC}$. The minimum points, instead, are only A and B; in this case the `Interval` object is initialized using a single value a, representing the interval $[a, a]$.

In other functions, the points of local minimum or maximum may not exist. For example, the Gaussian shape does not have any points of local minimum. In that situation, the `getMinPoints()` method can return a null value.

A.8 Summary

In this chapter, the new open-source library *Juzzy Constrained* has been presented. This toolkit, written in Java, has been developed as an extension of the fuzzy library *Juzzy* (for type-1 and type-2 fuzzy logic) and adds the support to constrained interval type-2 (CIT2) fuzzy sets and systems. This new class of fuzzy sets represents a useful alternative to the standard interval type-2 representation in the contexts in which a high level of interpretability is needed. Through the addition of some mathematical constraints, it ensure that a meaningful connection is kept between the shape of the footprint of the uncertainty, the embedded sets and the concept the CIT2 set is modeling. In the literature, it has also been shown how these properties can be used to produce explainable interval type-2 systems by processing only embedded sets with a *meaningful* shape for the determination of the interval centroid.

The chapter introduces the library and showcases the properties and utility of CIT2 models using a worked, practical example, clearly highlighting the advantages of CIT2 FLSs from an XAI point of view.

The toolkit presented here, is the first one to support CIT2 fuzzy sets and systems. The aim of this chapter is therefore to present the new library to the research community and also to encourage the discussion regarding the possible interpretability issues that may arise with the use of T2 and IT2 FSs (objective 4. of this thesis).

The structure of the library, its main classes and the defuzzification algorithms provided have been discussed, while in Sec. A.6 a CIT2 fuzzy logic system is built from scratch, with the help of code snippets to facilitate the

understanding of how the toolkit can be used.

Being still under development, the library has some limitation such as the fact that rules only support a single consequent or that the antecedents can only be connected using the *and* operator. In future works these aspects will be improved, adding rules with multiple consequents and different connectors for the antecedents as well as making the library compliant with the fuzzy markup language.

Appendix B

List of Common Abbreviations

FS	Fuzzy set
FLS	Fuzzy logic system
ES	Embedded set
AES	Acceptable embedded set
T1	Type-1
T2	Type-2
FOU	Footprint of uncertainty
UOD	Universe of discourse
IT2	Interval type-2
CT2	Constrained type-2
CIT2	Constrained interval type-2
GS	Generator set
DS	Displacement set
KM	Karnik-Mendel
EKM	Enhanced Karnik-Mendel
AI	Artificial intelligence
XAI	Explainable artificial intelligence