

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Ciencias Químicas e Ingeniería

Maestría y Doctorado en Ciencias e Ingeniería



“Modelo de Aprendizaje Automático Interpretable mediante Cómputo Granular”

TESIS

PARA OBTENER EL GRADO DE

DOCTOR EN CIENCIAS

Presenta:

RAÚL IGNACIO NAVARRO ALMANZA

Bajo la dirección de:

DR. JUAN R. CASTRO

Co-dirigido por:

DR. MAURICIO A. SÁNCHEZ

TIJUANA, BAJA CALIFORNIA, MÉXICO

OCTUBRE DEL 2022

AUTONOMOUS UNIVERSITY OF BAJA CALIFORNIA

School of Chemical Sciences and Engineering

Masters and Doctorate in Science and Engineering



“Interpretable Machine Learning Model through Granular Computing”

THESIS

TO OBTAIN THE DEGREE OF

DOCTOR IN SCIENCES

Presents:

RAÚL IGNACIO NAVARRO ALMANZA

Under the direction of:

DR. JUAN R. CASTRO

Co-directed by:

DR. MAURICIO A. SÁNCHEZ

TIJUANA, BAJA CALIFORNIA, MÉXICO

OCTOBER 2022

*With love for my parents who were my
support throughout all this journey.*

Universidad Autónoma de Baja California
FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

Folio No.326
Tijuana, B.C., a 06 de Mayo del 2022

C. Raúl Ignacio Navarro Almanza
Pasante de: Doctorado en Ciencias
Presente

El tema de trabajo y/o tesis para su examen profesional, en la
Opción TESIS

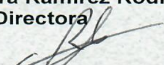
Es propuesto, por las C. Dr. Juan Ramón Castro Rodríguez y
Dr. Mauricio Alonso Sánchez Herrera

Quienes serán las responsables de la calidad del trabajo que usted presente,
referido al tema "Modelo de Aprendizaje Automático Interpretable mediante
Cómputo Granular"

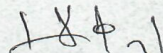
El cual deberá usted desarrollar, de acuerdo con el siguiente orden:

- I. INTRODUCCIÓN
- II. FUNDAMENTOS TEÓRICOS
- III. MODELO DE APRENDIZAJE
AUTOMÁTICO INTERPRETABLE
- IV. EXPERIMENTACIÓN Y DISCUSIÓN
DE RESULTADOS
- V. CONCLUSIONES Y TRABAJO FUTURO


Dra. Ana Alejandra Ramírez Rodríguez
Sub-Directora


M.C. Roberto Alejandro Reyes Martínez
Director Provisional




Dr. Juan Ramón Castro Rodríguez
Director de Tesis


Dr. Mauricio Alonso Sánchez Herrera
Co-Director de Tesis

Acknowledgements

I wish to extend my special thanks to my supervisors, Dr. Juan Ramón Castro Rodríguez and Dr. Mauricio Sánchez Herrera, for the assistance and support provided during the journey. Besides my advisors, I would like to thank Dra. Olivia Mendoza Duarte, Dr. Guillermo Licea Sandoval, and Dr. Antonio Rodríguez Díaz for their help with theoretical foundations and feedback. My gratitude to the *Facultad de Ciencias Químicas e Ingeniería de la Universidad Autónoma de Baja California* for being my Alma Mater; to CONACYT (by its acronym in Spanish of *Consejo Nacional de Ciencia y Tecnología*) for the financial support.

Finally, I must express my very profound gratitude to my friends, family, and fellows who helped me through all these years, making my days better. This accomplishment would not have been possible without them. Thank you.

Raúl Ignacio Navarro Almanza

Abstract

Interpretable machine learning is trending as it aims to build a human-understandable decision process. There are two main types of machine learning systems: white-box and black-box models. White-box models are inherently interpretable but commonly suffer from under-fitting phenomena; on the other hand, black-box models perform quite well in a wide range of application domain problems, but their reasoning behind a decision is hard or even impossible to understand. In the soft-computing area, fuzzy inference systems are rule-based systems that use fuzzy reasoning, bringing human perception modeling and computing with word capability. These rule-based systems are designed either manually or automatically but are commonly optimized to fit better some phenomena' data (in a supervised learning task). After the optimization process, the initial semantic meaning of fuzzy sets is modified (slightly, in the best cases), creating a gray-box model. The principal objective of the proposed methodology in this research work is to extract a high-quality rule in terms of comprehensibility, accuracy, and fidelity. This is accomplished by using a fuzzy linguistic interpretable model from an optimized neuro-fuzzy model, considering the initial knowledge context with which it was built.

This research work proposes a linguistic granule model representing generic entities affected by a linguistic description restricted by a context-free grammar. These abstract elements interact in an environment, and their aptitude or performance can be measured given a

particular metric.

The linguistic modifier changes the entities' behavior somehow; therefore, there are linguistic modifiers that maximize their aptitude in the environment. This work also proposes a methodology to find the pseudo-optimal linguistic modifier to better fit entities to the environment through grammar-guided genetic programming. Obtained results show that neuro-fuzzy systems could play an essential role in interpretable machine learning, providing natural language explanations from previous knowledge, keeping its semantic meaning after the optimization process by defining a linguistic granule.

Resumen

El aprendizaje automático interpretable está en tendencia, ya que tiene como objetivo construir un proceso de decisión comprensible para los humanos. Hay dos tipos principales de sistemas de aprendizaje automático: modelos de caja blanca y de caja negra. Los modelos de caja blanca son inherentemente interpretables, pero comúnmente sufren fenómenos de ajuste insuficiente; por otro lado, los modelos de caja negra funcionan bastante bien en una amplia gama de problemas de dominio de aplicación, pero su razonamiento detrás de una decisión es difícil o incluso imposible de entender. En el área de *soft-computing*, los sistemas de inferencia difusos son sistemas basados en reglas que usan razonamiento difuso, brinda la capacidad de modelado computable de la percepción humana mediante palabras. Estos sistemas basados en reglas se diseñan de forma manual o automática, pero normalmente se optimizan para adaptarse mejor a los datos de algunos fenómenos (en una tarea de aprendizaje supervisado). Tras el proceso de optimización, se modifica (ligeramente, en el mejor de los casos) el significado semántico inicial de los conjuntos borrosos, creando un modelo de caja gris. El objetivo principal de la metodología propuesta en este trabajo de investigación es extraer una regla de alta calidad en términos de comprensibilidad, precisión y fidelidad. Esto se logra utilizando un modelo lingüístico difuso interpretable a partir de un modelo neuro-difuso optimizado, considerando el contexto de conocimiento inicial con el que se construyó.

Este trabajo de investigación propone un modelo de gránulo lingüístico que representa

entidades genéricas afectadas por una descripción lingüística restringida por una gramática libre de contexto. Estos elementos abstractos interactúan en un entorno y su aptitud o rendimiento se puede medir dada una determinada métrica.

El modificador lingüístico cambia el comportamiento de las entidades de alguna manera; por lo tanto, existen modificadores lingüísticos que maximizan su aptitud en el medio. Este trabajo también propone una metodología para encontrar el modificador lingüístico pseudoóptimo para adaptar mejor las entidades al entorno a través de la programación genética guiada por la gramática. Los resultados obtenidos muestran que los sistemas neurodifusos pueden desempeñar un papel esencial en el aprendizaje automático interpretable, proporcionando explicaciones en lenguaje natural a partir de conocimientos previos, manteniendo su significado semántico después del proceso de optimización mediante la definición de un gránulo lingüístico.

Contents

List of Figures	3
List of Tables	7
1 Introduction	9
1.1 Hypothesis	13
1.2 Objectives	13
1.2.1 Particular objectives	13
1.3 Research contributions	14
1.3.1 Book chapters	14
1.3.2 Journal articles	14
1.3.3 Developed software	15
1.4 Document organization	15
2 Theoretical foundations	17
2.1 Machine learning	17

2.1.1	Interpretable machine learning models	18
2.1.2	Scope of interpretability	20
2.1.3	Interpretable models	21
2.2	Granular Computing	22
2.2.1	Granular Computing Models	23
2.2.2	Principle of justifiable granularity	27
2.3	Fuzzy Logic	27
2.3.1	Fuzzy sets	28
2.3.2	Fuzzy Systems	28
2.4	Grammar-Guided Genetic Algorithms	34
3	Interpretable Machine Learning Model	37
3.1	Linguistic Granule-based Grammar-Guided Genetic Programming Algorithm	37
3.1.1	Evolutionary operators	44
3.2	Problem definition	48
3.3	Unary hedge transformation over fuzzy sets	49
3.4	Methodology for building Interpretable Mamdani type Neuro-Fuzzy Model .	57
3.4.1	Knowledge base construction	57
3.4.2	Neuro-fuzzy model optimization	60
3.4.3	Linguistic granule optimization for semantic enhancement of pseudo-optimal knowledge base using binary relationships	65

4	Experimentation and results	75
4.1	Semantic enhancement of fuzzy sets using unary hedge transformations . . .	75
4.1.1	Finding linguistic modifiers for the approximation of same kind membership functions	78
4.1.2	Finding hedges for the approximation of different kind membership functions	80
4.1.3	Sensitivity Analysis	88
4.2	Semantic enhancement of fuzzy variables using binary hedge transformations	93
4.2.1	Generation of linguistic modifiers to explain model adjustment parameters	94
5	Conclusion	101
5.1	Unary hedge transformations over fuzzy sets	102
5.2	Binary hedge transformations over fuzzy variables	104
6	Future work	107
	References	109
	Bibliography	109

List of Figures

2.1	Relationship between coverage and specificity in granular models.	24
2.2	a) Example of Gaussian membership function with values $m = 5, \sigma = 3$. b) Example of Triangular membership function with values $a = 3, b = 5$. c) Example of Trapezoidal membership function with values $a = 2, b = 4, c =$ $6, d = 8$	29
2.3	In a) The linguistic hedge “ <i>Extremely</i> ” is applied to the fuzzy set “ <i>Hot</i> ”, which performs a concentration operation over its membership function. In b) The linguistic hedge “ <i>More-or-less</i> ” is applied to the fuzzy set “ <i>Hot</i> ”, which performs a dilation operation over its membership function. In c) The linguistic hedge “ <i>Upper than</i> ” is applied to the fuzzy set “ <i>Cold</i> ”, which performs a translation operation over its membership function.	33
2.4	Grammar-Guided Genetic Programming flow	35
3.1	Pareto front of specificity and coverage trade-off.	39
3.2	Derivation tree exploration example to create individuals restricted by CFG.	40

3.3	Crossover operator over two individuals. Nodes with blue backgrounds represent common nodes. The nodes with dotted lines indicate the selection over the common nodes. Child 1 and child 2 are the resulting individuals of crossover operation.	45
3.4	Mutation operator	47
3.5	Example of an unknown label of the fuzzy set after some optimization process.	48
3.6	Example of shift domain and dilation operation over a fuzzy set.	50
3.7	All hedge transformations. Blue line represents the original membership function and orange dotted line represents the transformed membership function by the linguistic hedge.	52
3.8	The Wang&Mendel procedure selects the k most relevant rule composition by membership belongingness. A and B are fuzzy variables of input feature fuzzification in the antecedent and Y to the consequent. Each fuzzy variable is evaluated by instance, and the higher membership value in the fuzzy set is selected to create a proposition. For instance 1, the rule “IF A is <i>low</i> and B is <i>med</i> THEN Y is <i>high</i> ” is selected	59
3.9	The Mamdani Neuro-fuzzy representation is composed by a non-fully connected 5-layer artificial neural network, all the computations corresponds to the involved operations in a Mamdani-type fuzzy inference system for q inputs, m rules, p fuzzy sets in consequent with Center-of-Sets defuzzification method. $C(G^i)$ denotes the centroid of consequent G^i	61
3.10	The proposed binary linguistic hedge <i>between</i> is applied over the two initial fuzzy sets <i>Cold</i> and <i>Hot</i>	67

3.11	In a) All their fuzzy sets are equally distributed and interpretable. In b) Its initial design changed due to the parameter tuning process, hence the semantic meaning. In c) is the reconstructed fuzzy variable from the initial to the optimized one by automatically hedge chain generation.	74
4.1	Trapezoidal membership function in source and target fuzzy set either a) and b). In a) the value of β is 0.05, which prioritizes accuracy over interpretability; and, in b) the value of β is 0.99, which equilibrates accuracy and interpretability.	81
4.2	Gaussian membership function in source and target fuzzy set, where the target is shifted to the right respect to source in either a) and b). In a) the value of β is 0.05, which prioritizes accuracy over interpretability; and, in b) the value of β is 0.99, which equilibrates accuracy and interpretability.	82
4.3	Gaussian membership function in source and target fuzzy set, where the target is shifted to the left with respect to source in both a) and b). In a) the value of β is 0.05, which prioritizes accuracy over interpretability; and, in b) the value of β is 0.99, which equilibrates accuracy and interpretability.	83
4.4	Gaussian membership function in source and target fuzzy set, where the target is shifted to the left with respect to source in both a) and b). In a) the value of β is 0.05, which prioritizes accuracy over interpretability; and, in b) the value of β is 0.99, which equilibrates accuracy and interpretability.	84
4.5	The overall evolution of fitness value over generations with different β values in the eight proposed experiments related to same membership function type.	85
4.6	The overall evolution of fitness value over generations with different β values in the eight proposed experiments related to different membership function type.	88

4.7	System performance changes in terms of similarity and hedge score with respect to the changes in β when the source and target membership functions are of the same type.	90
4.8	System performance changes in terms of similarity and hedge score with respect to the changes in β when the source and target membership functions are of different kind.	91
4.9	Overall mean performance of Mamdani Neuro-fuzzy model for each number of the rules, grouped by their three input clusters configurations.	97
4.10	Overall fidelity score (equation 3.6) grouped by each fuzzy variable configuration: 3, 5 and, 7 fuzzy sets each one.	98

List of Tables

4.1	Summary of the proposed linguistic terms, certainty, shift, hedge modifier value used in the cases studies.	77
4.2	Parameters used in the Grammar-Guided Genetic Programming in each step of the methodology.	78
4.3	Case study results where the source and target membership functions are of the same type, and β with value 0.01.	79
4.4	Case study results where the source and target membership functions are of the same type, and β with value 0.99.	80
4.5	Case study results where the source and target membership functions are of a different type, and β with value 0.01	86
4.6	Case study results where the source and target membership functions are of a different type, and β with value 0.99	87
4.7	Performances of the model, approximating membership functions of the same type, in terms of similarity, hedge score, and fitness; with respect to the following values of β : 0.01, 0.05, 0.1, 0.5, 0.7, 0.99.	89

4.8	Performances of the model, approximating membership functions of different type, in terms of similarity, hedge score, and fitness; with respect to the following values of β : 0.01, 0.05, 0.1, 0.5, 0.7, 0.99.	91
4.9	Selected hyper-parameters values in the optimization of automatically building fuzzy inference system, using Mamdani-type Neuro-fuzzy representation. . .	94
4.10	Result f1-score and standard deviations of 5 k-fold cross-validation of automatically built neuro-fuzzy model for each of 16 datasets.	96
4.11	Overall result of approximate model similarity between the optimized membership functions and linguistically transformed original knowledge. The similarity measure is shown in equation 3.6.	99

Chapter 1

Introduction

Nowadays, Artificial Intelligence is immersed in daily human life through automated decision-making systems. It is common to interact with those systems in various services such as bank loans, recommender systems, medical contexts. The decision-making by those systems directly impacts the final user due to the possibility of service denegation.

Just as there are situations in which the impact of the decision is not critical, there may be situations where it affects the final user considerably. Because of this, it is necessary to know in-depth how the models reach some conclusions and generate their outputs.

Interpretable Machine Learning (IML) area aims to create Machine Learning models that are easily understandable by humans. IML has been growing in popularity due to the decision made by those models that can directly affect people. Moreover, there are spreadly adopted by the enterprises to automatize specific processes; there are so many services that use ML models involved in daily human life. The goal of create interpretable models can be achieved by various methods, such as: using intrinsic interpretable models, regularization techniques, and post hoc explanation techniques. One trend is to build surrogate models (lower complexity) than the original one and more interpretable to understand the decision

process, such as rule-based models (Bastani et al., 2017; Chan and Chan, 2020; Vasilev et al., 2020).

Learning models called black boxes have been widely adopted by both the scientific and business community to solve problems of very different contexts. The common aspect between these applications is the objective of achieving the best possible performance (minimum error). However, it is vital to know why the learning model gives specific results to achieve confidence in the model. In black-box learning models, the data scientist does not know the why of the model when it is generated to solve a particular task, so this type of model is critical in problems such as medical diagnosis, detection of terrorism, and other applications that suppose a risk for people (Ribeiro et al., 2016).

There is special attention in achieving that the models of automatic learning grant to the user explanations of the reasoning to arrive at a particular result, the name of the area that is in charge to solve this situation is commonly called Explainable Artificial Intelligence or Interpretable Machine Learning; In this work, the concepts will be mentioned indistinctly. Explanations are necessary for understanding the models and generating confidence in machine learning models. If there are explanations of the model and do not correspond with the expert knowledge of the domain, the use of that particular model could easily be omitted (even if it presents an acceptable performance); the most important thing is that it would offer the user the possibility of creating new models in order to avoid the errors detected through the explanations (Smith and Nolan, 2018).

The machine learning models have been able to solve increasingly complex problems. However, these models have become more complex every time they are challenging to understand in the development of their conclusions. Due to this tendency, an interest has arisen in the area of Explainable Artificial Intelligence, as indicated by the emergence of different

government programs, such as DARPA’s Explainable Artificial Intelligence program ¹ and the General Regulation of Data Protection of the European Union (Goodman and Flaxman, 2016) where the emphasis is placed on the user’s right to an explanation, this made from the point of view of the end final user.

In the area of Explainable Artificial Intelligence, there are several proposals to create explanations of machine learning models. Some of them are focused on including the user in the learning process (human in the cycle) (Goodman and Flaxman, 2016), unification of logic and probability to provide learning models with a formal representation (Belle, 2017), building knowledge bases from results generated by learning models supported by explicit knowledge of the domain expert (Zhuang et al., 2017).

One of the main problems in creating interpretable learning models lies in the form of representation of the data, for which there are several possibilities, such as treating the data as fuzzy sets, probabilistic sets, rough sets, in order to reduce the complexity of the data. The reduction of dimensionality of data through feature transformation or relevant feature detection might benefit the tracking of the transformation of the data to generate rules later. This is one of the arguments by which it is feasible to use Granular Computation, whose paradigm is inspired by how the human mind processes information through multiple perspectives in hierarchy form. This offers the possibility of abstracting at different levels the data for the generation of different types of explanations, from the most specific to the most general.

The primary process in Granular Computing (GrC) is to represent objects or patterns through units called granules; that process is called granulation. These granules can be composed of other granules. One of the primary granule’s characteristics is that they can

¹<https://www.darpa.mil/program/explainable-artificial-intelligence>

be represented by any model such as those discussed above. The most important about this characteristic is that a composition of many mathematical models can model the granule; that is to say, it can be modeled from different perspectives.

In this work, we take advantage of the inherent interpretability that brings Fuzzy Inference Systems (FIS), rule-based models, and their antecedents are described linguistically. The antecedent is represented by fuzzy variables and sets, proposed by Zadeh (Zadeh, 1975). The inference process of those systems is performed through fuzzy reasoning, which aims to represent human perception and their inference mechanism under uncertainty.

An interesting characteristic of FIS is that their knowledge representation is composed of IF-THEN rules, where their antecedents and consequents are in natural language. For example, a proposition can be “**Temperature** is **hot**”, where **Temperature** is related to an input attribute and **hot** to the set which partially belong. As was described, this formal notation potentially brings an intrinsic high interpretability degree if their components are well-defined (Mencar and Fanelli, 2008).

The FIS are used in a variety of application domains in ML context, such as medical (De Medeiros et al., 2017; Gayathri and Sumathi, 2016; Honka et al., 2011; Pota et al., 2017; Yang et al., 2014; Panoutsos et al., 2010), robotics (Deshpande and Bhosale, 2013; Adhyaru et al., 2010), decision making (Zein-Sabatto et al., 2013; Cheng et al., 2008; Azadeh et al., 2016). Often this FIS modeling is data-driven in which their input partition, membership functions parameters, and rule structure are discovered by some unsupervised technique (e.g., clustering). However, in the optimization process in which these systems’ performance generally increases, there is a detach in the semantic meaning between the initial knowledge base and the optimized one. We propose a linguistic granular model to bring back the semantic meaning to the resulting optimized knowledge base; therefore, it increases the interpretability and explainability of the resulting model.

1.1 Hypothesis

1. Linguistic granules can be used to bring semantic meaning to optimized fuzzy inference systems by the principle of justifiable granularity.

1.2 Objectives

- Methodology for automatically designing the initial interpretable fuzzy system.
- Formal definition of the linguistic granule.
- Design the optimization method to find the pseudo-optimal granule design parameters in terms of specificity and coverage.

1.2.1 Particular objectives

- Designing and development of software framework to build type-1 and interval-type-2 fuzzy inference systems.
- Designing and development of software framework to granular linguistic optimization.
- Develop and publish the solution as a framework to promote research in explainable artificial intelligence applying fuzzy logic.

1.3 Research contributions

1.3.1 Book chapters

Navarro-Almanza R., Juárez-Ramírez R., Licea G., Castro J.R. Automated Ontology Extraction from Unstructured Texts using Deep Learning (2020) Studies in Computational Intelligence, 862, pp. 727 - 755 DOI: 10.1007/978-3-030-35445-9_50

Navarro-Almanza R., Castro J.R., Sanchez M.A. Interpretable machine learning from granular computing perspective (2019) Studies in Systems, Decision and Control, 209, pp. 185 - 197 DOI: 10.1007/978-3-030-17985-4_8

1.3.2 Journal articles

Navarro-Almanza R., Sanchez M.A., Castro J.R., Mendoza O., Licea G. Interpretable Mamdani neuro-fuzzy model through context awareness and linguistic adaptation (2022) Expert Systems with Applications, 189, art. no. 116098 DOI: 10.1016/j.eswa.2021.116098

Raul Navarro-Almanza, Mauricio A. Sanchez, Guillermo Licea, Juan R. Castro, Knowledge transfer for labeling unknown fuzzy sets using Grammar-Guided Genetic Algorithms, Applied Soft Computing, Volume 124, 2022, 109019, ISSN 1568-4946, DOI: 10.1016/j.asoc.2022.109019.

Navarro-Almanza R., Sanchez M.A., Castro J.R., Mendoza O., Licea G. Hierarchical Decision Granules Optimization Through The Principle of Justifiable Granularity. (2022) Special Issue of “Computación y Sistemas” on “Emerging Issues and Applications of Fuzzy Systems Neural Networks, and Metaheuristics” DOI: 10.13053/CyS-26-2-4252.

1.3.3 Developed software

Publication and copyright registration of the generated software:

- *FuzzySystem* Framework (Copyright registration in process).
URL: <https://fuzzy-framework.readthedocs.io/en/latest/>
- *IT2FuzzySystem* Framework (Copyright registration in process)
- *GGGP* Framework (Copyright registration in process)
- *Neuro-fuzzy* Framework (Copyright registration in process)
- *Symbolic Transformer* Framework (Copyright registration in process)

1.4 Document organization

The following sections correspond to theoretical foundations related to this thesis's research and contributions in Section 2. Section 3 provides a detailed description of the proposed solution to the definition and optimization of linguistic granules to build interpretable machine learning models applying the principle of justifiable granularity. Section 4 presents a set of the conducted experiments on the model evaluation and sensitivity. Finally, Section 5 finishes with the conclusion and implications of the proposed approach.

Chapter 2

Theoretical foundations

2.1 Machine learning

Machine learning is an artificial intelligence sub-area that aims to automatically find the best parameters of some mathematical model to describe some phenomena. When there are many data measured that describe these phenomena, it can be find a function that approximates its behavior through the samples, and this specific case is called supervised learning. Those models can be used to predict some quantity called regression task (whose output are real numbers) or to predict some class called classification task (whose output are natural numbers). The best parameters of some models are found by optimization methods, for example, the well-known gradient descent that is very popular in Neural Networks models.

A machine learning model can be represented as a function $f : X \times W \rightarrow Y$ where X and Y represent the inputs and outputs respectively, these data is contained in the training dataset; and W are the weights of that adjust the output of the system (in supervised training). The objective of learning methods is to find the best weights (W^*) that satisfy the next

relation $f(X, W^*) \approx Y$. In that brief machine learning model description, it is noticeable that the main objective of these models is to approximate the target value, diminishing the importance of how and why the decision is generated.

2.1.1 Interpretable machine learning models

The interpretability in the machine learning context is defined as the capability of some machine learning model to be easily understandable by the data scientist and even the final user of the model, such as customers of some application. They are some definitions of interpretability from a non-mathematical perspective identified by Molnar in (Molnar, 2019), such as interpretability, the degree to which humans can understand the cause of a decision (Miller, 2017). Furthermore, interpretability is the degree to which a human can consistently predict the model's result

The use of machine learning models in daily human life has become ubiquitous (Varshney, 2016). They are present in applications and services such as loans application, recommender systems for multimedia content, online shopping, and so many others. The decision-making by those systems directly affects the final user. However, all of them have the right to know why some decision has been made; of course, in many cases, machine learning model decisions do not negatively impact the user in case of making some mistake, which means those systems are not critical.

The interpretability in machine learning is growing in importance because of critical systems whose decisions could negatively affect the user, such as medical applications (Li et al., 2017), loan approval systems (Tsakonas et al., 2006), pedestrian detection in the autonomous driving car (Haspiel et al., 2018). The interpretability not only is focused on giving the final user an explication of why some decisions have been made, but it might be given to the data scientist insights of how the ML model is working to arise the conclusions

to make predictions, and more importantly, fix undesired behavior of the system in the face of some rare input data (different distribution of data in the training phase).

In the machine learning domain, there are many different mathematical models to represent some phenomena. There is a consensus of which models are easily interpretable (or simply interpretable) in terms of interpretability. There is a common idea that the simpler, the more interpretable model is, that sounds logic if we consider the fewer parameters and generally the lack of non-lineal transformations.

They are some fields that already have approaches to get a better understanding of the problem and solution through IML, such as e-learning to help students to a better understanding of subjects (Williams et al., 2018); In biology, understanding high-order interactions drive gene expression (Basu et al., 2018); In medicine to build a system to predict patients mortality with ST-elevation myocardial infarction (Li et al., 2017), building interpretable decision tree for diagnosis, and other tasks (Valdes et al., 2016); In neuroscience to studied a family of N-back working memory (Caywood et al., 2017); In vision-based systems to achieve better interpretability by a dictionary of atomic shapes in decision boundaries (Varshney, 2017). In-text analytics to document categorization as a classification task with individual word relevance (Arras et al., 2017); In robotics to generate explanations in humanoid robots (Beaton, 2018; Huang et al., 2018).

In machine learning problems, there is a common threshold between the quality of *what* is the predicted result of the model and *why* the model gives a specific output. The main problem is that appeared to improve one of them affects negatively another one.

The machine learning models can be classified by their transparency. In this work, this concept refers to the ease with which the user can understand why the model generated a

particular result. There are two known classes of models: gray-box and black-box models, which will be detailed below.

The machine learning models that are considered have gray box models; they stand out for their lack of non-linear computations, such as decision trees and linear models. Those models can be performed some operations to achieve some understanding of how the system makes the predictions. Also can be explained by counterfactual cases, which means to know which features and how can be modified in order to change its output to some desired result of the model.

The black box models generally have a huge amount of hyper-parameters and weights, the same for the non-linear computations that perform. The most popular black-box models are the Neural Networks due to a large amount of successive non-linear transformations. It is well known the lack of interpretability of these models. However, they can achieve better behavior in many complex tasks.

2.1.2 Scope of interpretability

There are different perspectives in terms of model interpretability; one of them is focused on generating an explanation for each instance that the model evaluates. On the other hand, there is a perspective that aims to explain the entire behavior of the model. Both trends have their own advantages and drawbacks. The selection of one of them depends on the application domain and goals.

Global interpretability aims to generate an explanation of the entire solution surface of the model. The most common artifact to create such an explanation relies upon rule-based

models. These approaches consist of creating a knowledge base of the entire domain of the problem from the solution surface. Within the most outstanding works, it is found that of Lakkaraju (Lakkaraju et al., 2016), where the knowledge base is composed of decision sets, and in the learning process, various objectives are taken into account, such as fidelity, disambiguation, interpretability.

Local interpretability is focused on generating an explanation for each prediction (or evaluation). Usually, this explanation relies on identifying the essential attributes for some model output. There are some approaches where its interpretability relies upon the measure of certainty to give to the user of machine learning model (van der Waa et al., 2018).

2.1.3 Interpretable models

Another perspective of model explanations depends on the model itself. There are techniques of interpretability that only can be used in certain machine learning models, such as the Learning Vector Quantization model (Brinkrolf and Hammer, 2018; Hofmann et al., 2014), Multi-operator Temporal Decision Trees (Shalaeva et al., 2018), Bayesian-based model for learning rule sets (Wang et al., 2017). On the other hand, some techniques do not require some specific model to work. Generally, the explanation techniques that require a specific model are more interpretable because of understanding model behavior. However, the model-agnostic techniques can be used in a broader application domain.

Model-dependent interpretability techniques depend on the machine learning model that is used. The insights in those techniques are more easily understandable by the data scientist. Conversely, In model-agnostic interpretable techniques, the selected model for pattern discovery is irrelevant; a common way to understand them is to visualize the model

as a black box where only their inputs and outputs are analyzed.

2.2 Granular Computing

The Granular Computing paradigm is inspired by how the human mind processes information through multiple perspectives in hierarchy form. This hierarchy thinking form allows focusing on different essential attributes inherent of some object, depending on the requirements in reasoning processes.

Granular Computing is a multidisciplinary area that encompasses theories, techniques, models, and methodologies to solve complex problems. It is a paradigm of solving problems with a single fundamental element called a granule, formulated through the abstraction of properties in common between objects or patterns in the data. The properties that can exist between these objects or patterns are similarity, equality, proximity (Pal et al., 2010).

In granular computing, the process of creating the information granules is crucial and can be represented by various models and interactions between them. Granular representation is advantageous when the problem presents uncertainty and incomplete data. A granule can be modeled as $\mathbf{A} : X \rightarrow \mathcal{G}(X)$ where X is the universe of discourse and \mathcal{G} is the framework of information granules. The process of granulation can be represented as a mapping from one domain to another.

There is an important concept in Granular Computing that is called Granular World (\mathbf{G}) that refers to the environment of information granules that supports all process in information granulation, information processing and information exchange (Bargiela and Pedrycz, 2003). In this environment can be composed by different granules order but same domain, each order has a different perspective of data, in other words, different hierarchies of granules;

where the formal representation of Granular World is $\mathbf{G} = \langle X, \mathcal{G}, \mathbf{A}, \mathbf{C} \rangle$, where

- \mathbf{A} : is a granule of information.
- \mathbf{G}^n : is the granular world.
 - X : universe of data discourse.
 - $\mathcal{G}(X)$: denotes the *framework* of granules of information.
 - \mathbf{A}^n : is a information granule (of type n).
 - \mathbf{C} : is a mean of communication

Some exciting concepts un Granular Computing paradigm, such as i) principle of justifiable granularity, ii) coverage, and iii) specificity, the first one is referred to as a fundamental concept in the paradigm and is about forming information granule from data by available evidence. The concept of specificity refers to the level of characterization of a granule to detail some abstraction of the information granule. Finally, coverage is referred to the number of elements of the domain that correspond to some description of an information granule. There is a relationship between specificity and coverage; while the specificity increases, the coverage decreases, and vice versa.

The figure 2.1 is shown the behavior of the coverage and specificity. If the specificity is very high, then the coverage is the number of elements in the universe of discourse that belong to the given specification. Otherwise, to achieve high coverage, reducing the specificity and describing the information granules in a very general form is necessary.

2.2.1 Granular Computing Models

A granule of information can be a class of numbers, a group of images, a class of regions, a set of concepts, a set of objects, and a category of data. The granules of information

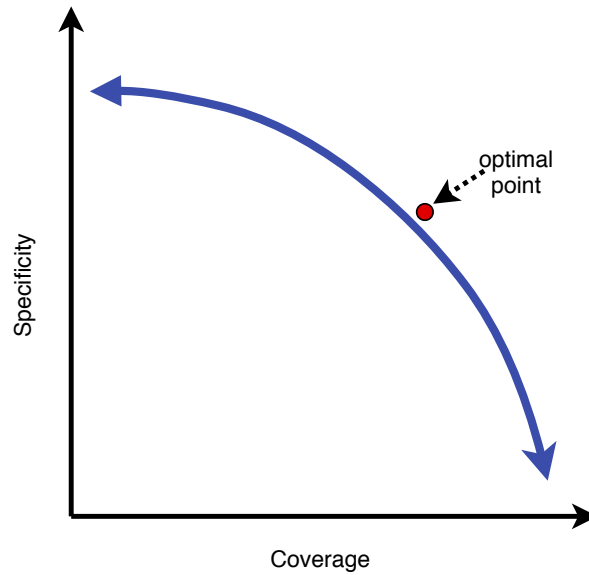


Figure 2.1: Relationship between coverage and specificity in granular models.

can be represented in the form of fuzzy sets, rough sets, fuzzy rough sets, neural networks, probabilistic sets. (Bargiela and Pedrycz, 2003). Each model represents a different perspective of data. In addition, the composition in the granules is possible; therefore, a granule of a particular order can be used to grant an explanation considering different points of view or several explanations according to the perspective.

There are some approaches that combine both worlds, machine learning and granular computing, to overcome some issues related to the considerable amount of data (Guo and Wang, 2019), related to interpretability issues (Xu et al., 2015; Nápoles et al., 2018), for knowledge representation as in Pedrycz (Pedrycz and Chen, 2011) proposes the use of granules of information to represent a knowledge base using fuzzy sets, this knowledge base is created from different learning models.

In the following, it is described some of the most used models in the area of Granular Computing.

Rough sets

In rough sets, situations are modeled where one can not distinguish the belonging of an object in a given set, but an element can be related to other elements, so upper and lower limits are defined for the description of the element in a given set, as shown below:

$$\underline{R}X = \bigcup \{Y \in U/R | Y \subseteq X\}$$

$$\overline{R}X = \bigcup \{Y \in U/R | Y \cap X \neq \emptyset\}$$

Where X is a classic set of objects, in a universe of speech U , R is the relation between existing elements in U . This description of elements is usually used to represent relationships between them.

Fuzzy sets

Theory introduced by Zadeh (Zadeh, 1965) to model the uncertainty in natural language. It is a model to describe the degree of belonging in continuous space $A = \{(x, \mu_A(X)) | x \in X\}$, where $\mu_A : X \rightarrow M$ is the membership function of x in A that maps X to the M membership space. The membership $\mu_A(X)$ indicates the degree of similarity (compatibility) of a x object to a concept characterized by the fuzzy set A . The domain of M corresponds to the range $[0, 1]$.

One of the main characteristics of the modeling of granules by fuzzy sets is the one that can relate a granule to a linguistic variable and thus provide a higher degree of interpretability for the data scientist.

Fuzzy Rough Sets

In fuzzy rough sets (Dubois and Prade, 1992), the aim is not only to model uncertainty, as in the case of fuzzy sets, but also to add concepts of incompleteness and imprecision. Here is how to model this type of sets:

$$(R_B \downarrow R_A)(x) = \inf_{y \in U} \max\{1 - R_B(x, y), R_A(x)\}$$

$$(R_B \uparrow R_A)(x) = \sup_{y \in U} \min\{1 - R_B(x, y), R_A(x)\}$$

Where the upper and lower approximations involve fuzzy equivalence and decision classes.

Neural Networks

Granular neural networks are models capable of processing granular data, which may well come from both numeric and linguistic data (Ding et al., 2014). This processing may consist of prediction of new information, extraction of information in the form of granules, a fusion of sets of information granules, or compression thereof.

The use of neural networks for processing or identification of granules is widely used, the most well-known works for generation of rules using granular neurons (Ding et al., 2014), neural networks using a scheme of functional neurons that allow various representations of granules (Loia and Tomasiello, 2017), neuro-fuzzy networks for generating rules by granular computation (Panoutsos and Mahfouf, 2010).

2.2.2 Principle of justifiable granularity

The principle of justifiable granularity is a technique to find pseudo-optimal model parameters in order to define the adequate information granule size. The representational parameters of the granule Ω directly impact its quality design. There are two properties of every granule that reflects indirectly the model design quality that can be used by the principle of justifiable granularity to find a suitable design granule representation of phenomena. In general terms, the granule size should be as smaller as possible but at the same time should cover as many samples as possible. The specificity is defined as the granule semantic meaning, a highly detailed granule should imply that the smaller the information granule is, the better. The coverage is the amount of experimental evidence that supports the granule design. For instance, if the granule Ω is defined as an interval, we should expect to have a significant amount of data between the bounds of the granular model (Pedrycz and Homenda, 2013).

There is a conflict in both granules' properties; the lower the specificity is, the higher coverage is. Therefore, the ideal granule representation should have the highest coverage and specificity. Due to this trade-off, finding the best granule parameters can be set up as a multi-objective optimization problem whose objectives are to maximize both the specificity and coverage.

2.3 Fuzzy Logic

Fuzzy logic is a multivariate logic that is contrary to bivariate in which the truth values are $\{0,1\}$, their truth values belong to the interval $[0,1]$. This is an interval that represents the perceived degree of a person in a given context. The term Fuzzy Logic was coined by Zadeh (Zadeh, 1965) as an approach to model how people make decisions based on imprecise information and non-numerical information. Human beings can grasp complex concepts and

make decisions under uncertainty using linguistic information.

2.3.1 Fuzzy sets

Fuzzy sets were proposed in contrast with a classical set; they do not have a crisp boundary; instead, the elements belong in some degree to a fuzzy set that is represented by a real number between 0 and 1. Membership functions model the characterization of degree transition of truth values.

A set of ordered pairs define a fuzzy set A in X domain (equation 2.1):

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

$\mu_A(x) \in [0, 1]$ is the membership function that models the membership degree of the elements x in the universe of discourse X .

2.3.2 Fuzzy Systems

The Fuzzy system is a type of rule-based knowledge system that represents their propositions using natural language for antecedents (also for consequents in Mamdani systems). This singular characteristic allows the expert to easily understand the behavior of the model if their linguistic representation is aligned to the user.

These knowledge models can be built manually by experts or be automatically designed, usually by clustering techniques. The following fuzzy rule base shows an example of Mamdani-type FIS knowledge base for m rules, n inputs, and p outputs classes:

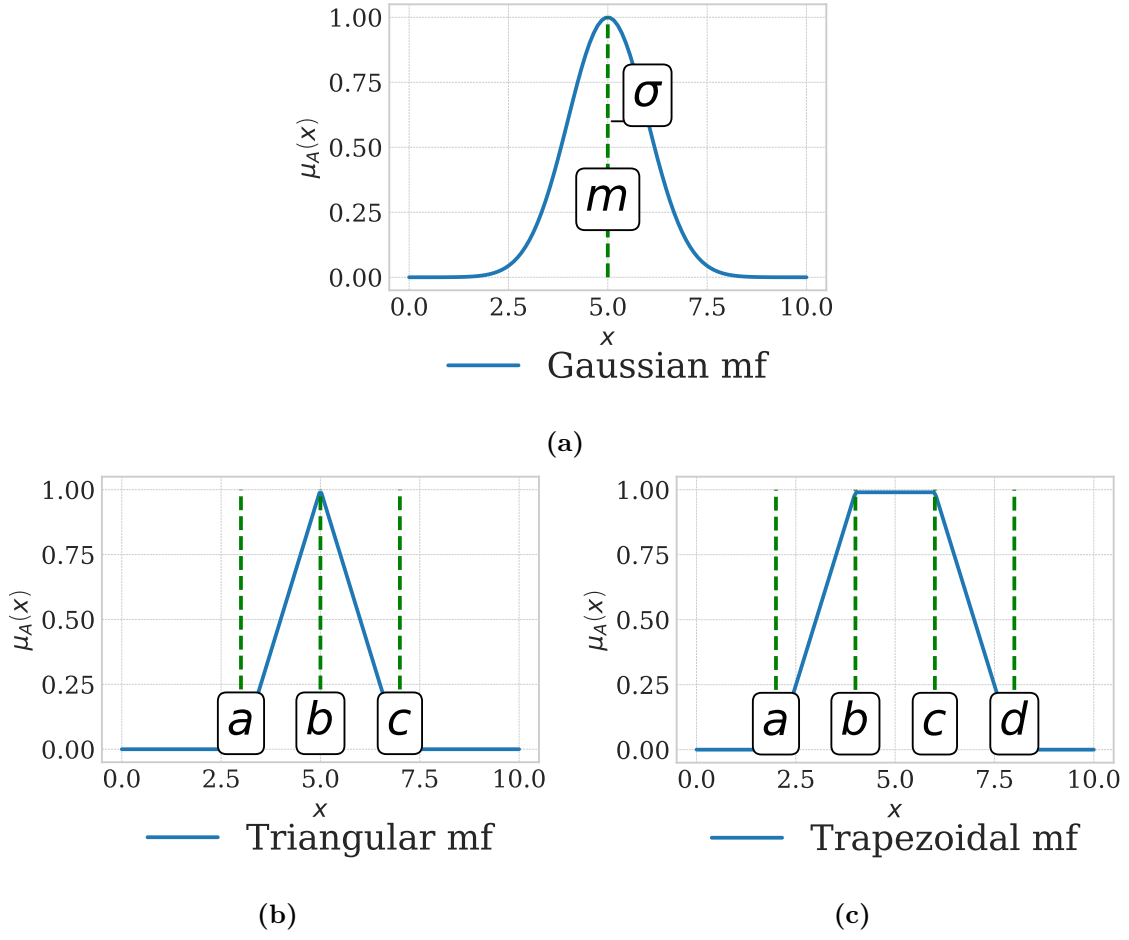


Figure 2.2: a) Example of Gaussian membership function with values $m = 5, \sigma = 3$. b) Example of Triangular membership function with values $a = 3, b = 5$. c) Example of Trapezoidal membership function with values $a = 2, b = 4, c = 6, d = 8$.

R^1 : IF x_1 is *low* and ... and x_n is *low* THEN, y is G_1

R^2 : IF x_1 is *low* and ... and x_n is *high* THEN, y is G_2

\vdots

R^m : IF x_1 is *high* and ... and x_n is *low* THEN, y is G_p

(2.2)

Where x_i and y are fuzzy variables, they represent the rule antecedents and consequents,

respectively. The Zadeh's linguistic variable (Zadeh, 1975) is characterized by a quintuple $(x, T(x), X, \mathcal{G}, M)$, which x is the name of the variable; $T(x)$ is the term set of x , linguistic terms; X is the universe of discourse; \mathcal{G} is a synthetic rule which generates linguistic terms in $T(x)$; M is a semantic rule which associates each linguistic value A its meaning $M(A)$, where $M(A)$ denotes a fuzzy set in A . A fuzzy set A in X domain is defined as a set of ordered pairs (equation 2.1).

Fuzzy systems perform the inference of the complete model by three principal steps:

1. Fuzzification of all inputs values through the defined membership functions.
2. Inference process based in the knowledge base rules, and finally.
3. The process of defuzzification that maps the resulted fuzzy value (inference process) to a crisp value.

Given the following generic rule structure: IF x_1^i is A^1 and \dots and x_n^i is A^n THEN y^i is G^p .

The fuzzification process performs a mapping of the input $(x_1^i$, where the super-index is the instance and sub-index i the attribute domain) to a fuzzy space by a membership function $(\mu_A(x))$. The membership function corresponds to the antecedent design of a given rule. After the fuzzification process, every input belongs to the interval $[0,1]$ domain, which represents the membership degree of belongingness to a fuzzy set (concept).

The inference process is conducted by the evaluation of every rule in the knowledge base. It is important to notice that the output in these step belongs to the fuzzy domain, which means that every rule would have a *firing strength* that represent the compatibility of the input to a given rule. The inference operation are performed by the *t-norm* operator $(\tilde{*})$, that could be set to different process such as: *minimum* or *product* operation, among others. The equation 2.3 shows the compatibility degree (α) process of a given fuzzy rule (l) .

$$\alpha^l(x_i) = \mu_{A_1^l}(x_i) \tilde{*} \dots \tilde{*} \mu_{A_n^l}(x_i) = \tilde{T}_{r=1}^p \mu_{A_r^l}(x_i) \quad (2.3)$$

Where x_i is a given instance input, $\mu_{A_n^l}$ is the membership function that represent the perception model of the antecedent n (n^{th} input attribute) in rule l .

In order to get output fuzzy value of the rule, the compatibility value perform a $\tilde{*}$ operation to the consequent defined in the rule l , $\mu_{B^l}(y) = \mu_{G^l}(y) \tilde{*} \alpha^l$.

They are many methods to transform the resulting fuzzy value to a crisp value, one of them is the center-of-set defuzzification method, equation 2.4.

$$\hat{y}^{COS}(\mathbf{x}') = \frac{\sum_{l=1}^m COG(G^l) \alpha^l(\mathbf{x}')}{\sum_{l=1}^m \alpha^l(\mathbf{x}')} = \frac{\sum_{l=1}^m c^l \alpha^l(\mathbf{x}')}{\sum_{l=1}^m \alpha^l(\mathbf{x}')} \quad (2.4)$$

Which COG is the center of gravity of the membership function, m is the number of fuzzy rules in the FIS, \mathbf{x}' is an arbitrary input crisp value to perform the inference and defuzzification process; c^l is the center of the l th consequent set; α^l is the firing level of the rule.

Linguistic hedges

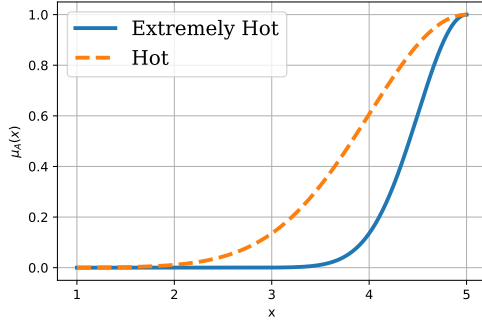
Linguistic hedges are modifiers that transform a membership function (Zadeh, 1972), usually this modifier is represented by the expression A^p (equation 2.5) where A is a fuzzy set, and p is the value that modifies membership function.

$$A^p = \{(x, \mu_A(x)^p) | \mu_A \in [0, 1]\} \quad (2.5)$$

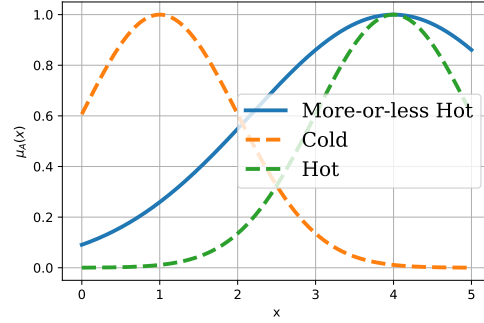
When the value of $p > 1$, the membership function concentrates its support, but if $p < 1$

performs an opposite operation, the dilation. A linguistic term can represent the different values of the exponent, a common (and arbitrary) value for the “*very*” term is $p = 2$, which is coherent by its semantic notion of concentration (imprecision/uncertainty reduction), see figure 2.3a. On the contrary, the linguistic term “*kind-of*” is related to an increment of uncertainty, therefore should be represented by a $p < 1$, see figure 2.3b. These linguistic modifiers can be composed of more than one hedge, named hedge chain, for example, “*very very*”.

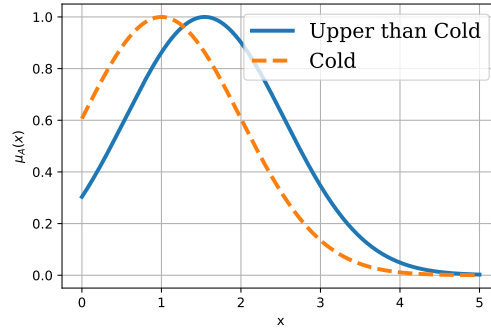
This work also considers another linguistic modifier that modifies the membership function by translation instead of operating on uncertainty. For example, the hedge chain “*more than*”, which can be modeled by calculating $\mu_A(x - r)$, where r is the shift factor of its core (see figure 2.3c). The main reason to use a fuzzy hedge transformation is to create an interpretable layer better to understand the machine learning model after the optimization process. Due to this, it is important to select easily understandable linguistic terms and associate them with the transformation function that corresponds to their semantic meaning.



(a) Concentration operation.



(b) Dilation operation



(c) Translation operation

Figure 2.3: In a) The linguistic hedge “*Extremely*” is applied to the fuzzy set “*Hot*”, which performs a concentration operation over its membership function. In b) The linguistic hedge “*More-or-less*” is applied to the fuzzy set “*Hot*”, which performs a dilation operation over its membership function. In c) The linguistic hedge “*Upper than*” is applied to the fuzzy set “*Cold*”, which performs a translation operation over its membership function.

2.4 Grammar-Guided Genetic Algorithms

Grammar-Guided Genetic Programming (GGGP) is a Genetic Algorithm (GA) used in optimization problems. GA is inspired by biological evolution, where the fittest individual survives and crosses with another to create offspring equally or better fitted. These abstractions are formally described to be computable and get a pseudo-optimal solution of an optimization problem relying on operators such as mutation, crossover, and selection.

The phenotype represents the domain solution space of a given problem, and in order to transform this representation to a computable form, the phenotype is mapped to a genotype space. Usually, the genotype is a binary representation of the solution space. For example, given an equation whose solution space is in the real values domain, their computable representation could be binary strings representing a limited range of possible real number representations. Generally, the genotype length is fixed to a certain number of bits in their representation; therefore, for a wide range of problems whose phenotype length is highly variable, this kind of representation is not suitable. Genetic Programming aims to cover problems that require variable-length representation, commonly used to find structures rather than parameters. The genotype is a tree-based representation. However, these representations are non-restricted; therefore, in the optimization process might be hard to discriminate against well-formed individuals.

In Grammar-Guided Genetic Algorithms, each individual is encoded by a graph and structured by a context-free grammar (Manrique et al., 2009), all possible derivation trees denote the universe of possible solutions. Figure 2.4 shows the whole methodology of grammar-based genetic programming. At the first step, a population is randomly generated, and then those individuals are evaluated by a fitness function to measure their aptitude (how close they are to the goal). Some portion of the population is selected to reproduce using the fitness values, applying a crossover operator that combines two individuals to create

a new one with their parents' information. Once the offspring are generated, the mutation operation is randomly applied to some portion of the new population. The mutation operation modifies some parts of each individual, adding some randomness that might result in the best individuals. After applying the genetic operators to the individuals, the process is repeated until termination criteria are met.

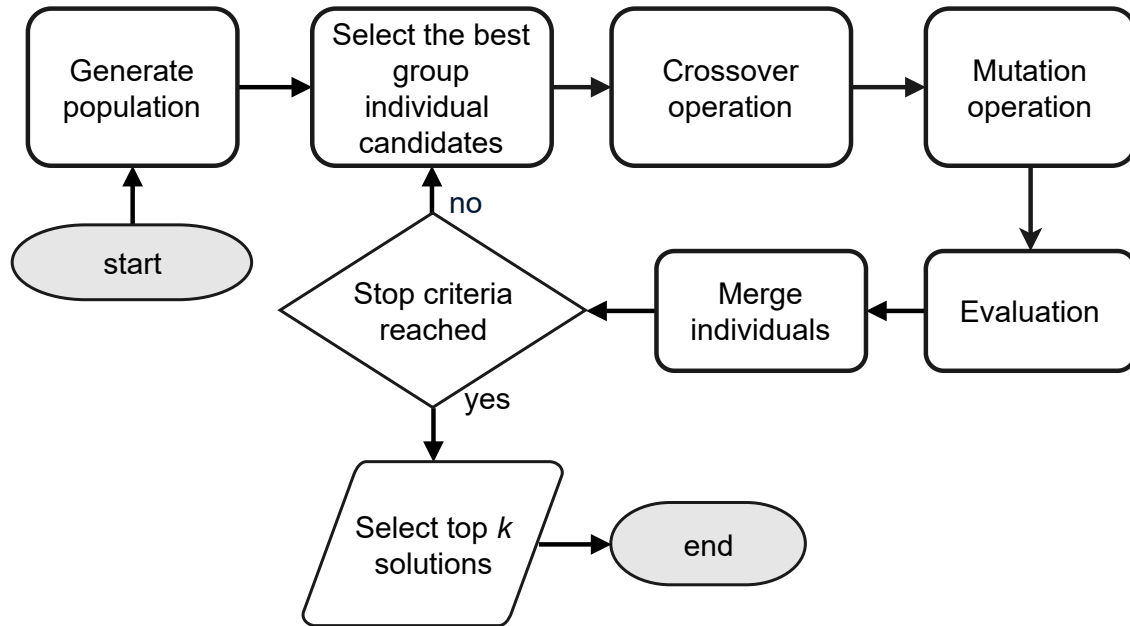


Figure 2.4: Grammar-Guided Genetic Programming flow

The algorithmic view of the methodology is presented in algorithm 1.

Algorithm 1: Grammar-Guided Genetic Programming Algorithm

Data: Source Fuzzy Set: \mathcal{S} ,
 Target Fuzzy Set: \mathcal{T} ,
 Number of generations: g ,
 Mutation rate: mr ,
 Selection percentage: s
 Number of best top k candidates
 Tolerance error: err
Result: Best hedge candidates (\mathcal{R}) to approximate $hedges(\mathcal{S}) \rightarrow \mathcal{T}$

```

1  $i = 0$ ;
2  $\mathcal{R} = \emptyset$ 
3  $\mathcal{P}$  = Generate an initial population;
4 while  $i < g$  and  $\max(\mathcal{F}) < err$  do
5    $\mathcal{F} = fitness(\mathcal{P}; \mathcal{S}, \mathcal{T})$ ;
6    $\mathcal{S}$  = Select the top  $s$  best candidates based on  $\mathcal{F}$ ;
7    $\mathcal{C} = crossover(\mathcal{S})$ ;
8    $\mathcal{M} = mutate(\mathcal{C}, mr)$ ;
9    $\mathcal{P} = merge(\mathcal{M}, \mathcal{C}, \mathcal{P})$ ;
10   $i = i + 1$ 
11 end
Output: The best  $k$  candidates of set  $\mathcal{R}$ 

```

Chapter 3

Interpretable Machine Learning Model

3.1 Linguistic Granule-based Grammar-Guided Genetic Programming Algorithm

In this work, fuzzy systems are the core model to build an interpretable machine learning model. The reason for this selection is the user-friendly interface of fuzzy systems, which offer an explanation of the decision-making process due to the usage of natural language. Moreover, their neural-network representation allows us to optimize the design parameters by gradient-descend-based algorithms for performance improvement.

Even starting from an interpretable and well-designed knowledge base, after the optimization problem, the semantic meaning of the fuzzy sets is lost. In order to find the pseudo-optimal linguistic description to represent the optimized membership functions design parameters, a grammar-guided search is proposed using an evolutionary algorithm.

The optimal linguistic description should have the following characteristics:

- The linguistic description must be as specific as possible, which means that the smaller the length is, the better.
- The linguistic description must approximate a given reference entity.

The characteristics of the linguistic description are aligned to granule characteristics, *specificity* and, *coverage*.

The linguistic granule is defined as the quintuplet $\mathbb{L} = (\mathbf{e}, d, \mathcal{G}, T, E)$, where \mathbf{e} is an entity which interacts in the environment E . d is the linguistic descriptor which is a derivation tree of the grammar \mathcal{G} and, T is the transformation function that modifies a given entity \mathbf{e} through the descriptor d .

The specificity of the granule \mathbb{L} represents the lack of ambiguity and uncertainty in the description. For instance, it could be defined as shown in the equation 3.1. The smaller the descriptor is, the better. ρ is a permittivity parameter that allows controlling the penalization threshold of the description's length; γ is the parameter for slope control which can create a smoother transition.

$$sp(l) = 1 - \frac{1}{1 + e^{\frac{-length(l) - \rho}{\gamma}}} \quad (3.1)$$

The coverage of the granule \mathbb{L} is defined as the performance or aptitude value of the entity \mathbf{e} interacting in the environment E . In order to visualize a function that represents this idea, consider the equation 3.2, which is the correlation between two elements. The higher the approximation is, the better. The coverage is the experimental evidence or support of the actual linguistic granule. The coverage measures the similarity or approximation of the transformed entity e to the target or references y .

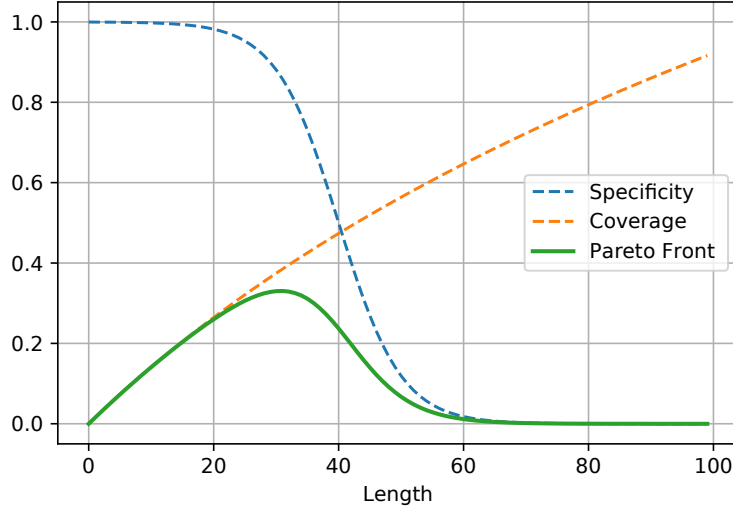


Figure 3.1: Pareto front of specificity and coverage trade-off.

$$cov(l; y) = 1 - \frac{\sum_i (y_i - T(l)_i)}{\sum_i (y_i - \bar{y})} \quad (3.2)$$

In order to find the best descriptor $d \in \mathcal{G}$, it is proposed the definition of a Pareto front by multiplicative arrange of the specificity and coverage, equation 3.3. Figure 3.1 shows an instance of the Pareto front of the proposed metrics for specificity and coverage.

$$pareto(l; y) = sp(l) \times cov(l; y) \quad (3.3)$$

The search process to find the best linguistic description of an entity to perform better an environment is carried out by a proposed evolutionary algorithm. The individual genotype is represented as a tree structure and is a derivation of a grammar \mathcal{G} . The generation process of the individuals is performed following the algorithm 2. In order to generate individuals that belong to a given context-free grammar, it is needed to explore each node until they reach only terminals (see figure 3.2). Due to the infinite loops that could exit, we proposed a

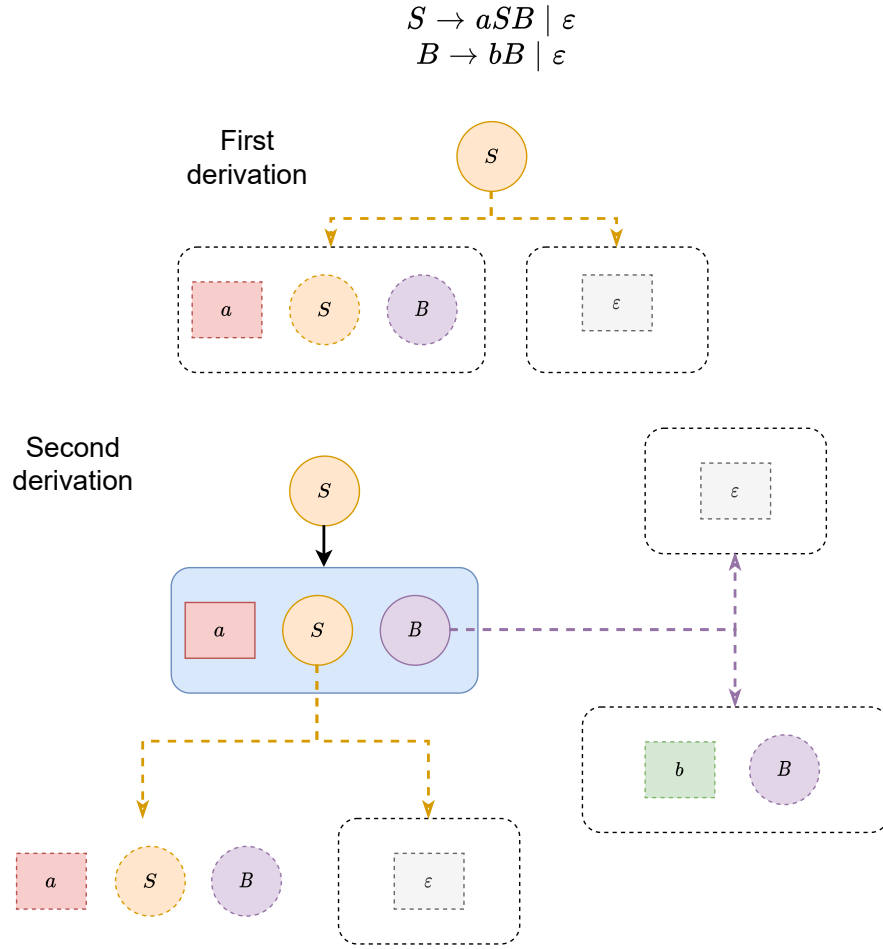


Figure 3.2: Derivation tree exploration example to create individuals restricted by CFG.

heuristic to increasingly high the probability to reach all terminal states by exploring paths closer to leaf nodes.

The principal parameters of the algorithm are δ and graph depth d . After d iterations it is performed a reduction of the probability of being selected certain paths farther to terminal nodes. The probability decrease value is given for δ , as lower the value is, the exhaustive the exploration is. On the contrary, if δ has a big value, fewer explorations are performed.

Algorithm 2: Population generation algorithm in 3GP**Data:** Context-Free Grammar: CFG ,Number of individuals: N Delta: δ ,Graph depth: d **Result:** Population set \mathcal{P}

```

1  $\mathcal{P} = \emptyset$ ;
2  $i=0$ ;
3 while  $i < N$  do
4   Generate an initial random derivation tree based on the grammar  $CFG$ .
    $G = random\_derivation(G)$ ;
5    $j=0$ ;
6   while  $has\_non\_terminals(G)$  do
7     if  $d < j$  then
8        $G = random\_derivation(G)$ 
9     end
10    else
11       $dist = steps\_to\_terminal(node) \forall node \in nodes(G)$ ;
12       $weights = \max(dist)/dist - j \times \delta$ ;
13       $\rho = weights / \sum weights$ , where  $weights \in [0, \inf)$ ;
14      Random derivation of a node based in probability distribution  $\rho$ 
       $G = G \cup random\_derivation(G, \rho)$ ;
15    end
16     $j = j+1$ ;
17  end
18   $\mathcal{P} = \mathcal{P} \cup G$ ;
19   $i = i+1$ ;
20 end

```

Output: The population set \mathcal{P}

Algorithm 3 describes the general grammar-guided genetic programming procedure. The main parameters are the Context-Free Grammar CFG , the environment in which each individual interacts and get a reward value (*fitness*), number of individuals N to be generated and maintained, the maximum number of generations g , mutation rate m , selection percentage s , number of best top k candidates as output, tolerance error (*err*) to early stop the procedure, δ and δ_m are the values to restrict the exploration depth in the population generation and mutation, respectively. Graph depth d is the iteration threshold to start applying the regularization parameters δ and δ_m . Parsimony (ρ) is a regularization parameter for penalizing long individuals. Diversify (\tilde{d}) establish the range of iteration where the diversification procedure is executed, which prunes similar individuals (algorithm 4).

Algorithm 3: Grammar-Guided Genetic Programming Algorithm

Data: Context-Free Grammar: CFG ,Environment: $fitness$,Number of individuals: N Number of generations: g ,Mutation rate: m ,Selection percentage: s Number of best top k candidatesTolerance error: err Delta: δ Delta for mutation: δ_m Parsimony: ρ Graph depth: d Diversify: \tilde{d} **Result:** Best top k individuals (\mathcal{P})

```

1  $i = 0$ ;
2 //Generate an initial population restricted by  $CFG$ 
3  $\mathcal{P} = \text{generate}(\mathcal{C}, N, \delta, d)$ ;
4 while  $i < g$  and  $\max(\mathcal{F}) < err$  do
5    $\mathcal{F} = \{fitness(p) - length(p) \times \rho | \forall p \in \mathcal{P}\}$ ;
6    $\mathcal{S} = \text{Select the top } s \text{ best candidates based on } \mathcal{F}$ ;
7    $\mathcal{C} = \text{crossover}(\mathcal{S})$ ;
8    $\mathcal{M} = \text{mutate}(\mathcal{C}, m, \delta_m)$ ;
9    $\mathcal{P} = \text{merge}(\mathcal{M}, \mathcal{C}, \mathcal{P})$ ;
10  if  $i \% \tilde{d} = 0$  then
11     $\mathcal{P} = \text{diversify}(\mathcal{P})$ ;
12  end
13   $i = i + 1$ 
14 end

```

Output: The best k candidates of set \mathcal{P}

Algorithm 4: Algorithm for diversification of the population in 3GP

Data: Sorted Population: \mathcal{P} ,
Number of desired individuals: N
Result: Population subset $\bar{\mathcal{P}}$

- 1 Select the individual with the highest fitness value.
- 2 $\bar{\mathcal{P}} = p_0 \in \mathcal{P}$;
- 3 $\mathcal{P} = \text{remove}(\mathcal{P}, p_0)$;
- 4 $i=0$;
- 5 **while** $i < N$ **do**
- 6 $p_i = \{p \mid \min \text{similarity}(p, \bar{\mathcal{P}}) \forall p \in \mathcal{P}\}$;
- 7 $\bar{\mathcal{P}} = \bar{\mathcal{P}} \cup p_i$;
- 8 $\mathcal{P} = \text{remove}(\mathcal{P}, p_i)$;
- 9 $i = i+1$;
- 10 **end**

Output: The population subset $\bar{\mathcal{P}}$

3.1.1 Evolutionary operators

The evolutionary operators such as crossover and mutation work with derivation trees formed by the proposed grammar. Those operators ensure that the resulting individuals belong to the grammar. Therefore, they have a well-formed structure avoiding incoherent solutions. The crossover operator performs a combination of two individuals, in which their offspring are expected to create better solutions to the proposed problem. The mutation operator performs a random change over some portion of the population to extend the region search over the solution space.

Crossover

The crossover operator applies a combination process in a set of selected population individuals. Given two parents (derivation trees) in this specific domain, randomly choose one of their nodes that both parents have a node in common. The branches are exchanged, creating a new individual. Figure 3.3 shows a simple crossover process example. The crossover operator

is described in detail in the algorithm 5.

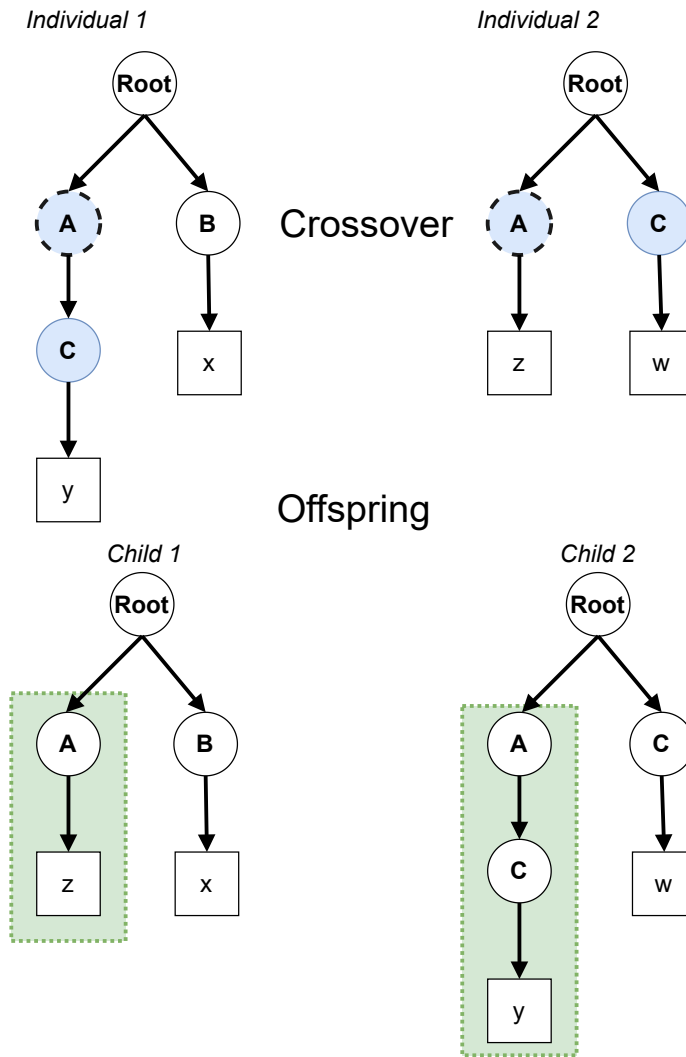


Figure 3.3: Crossover operator over two individuals. Nodes with blue backgrounds represent common nodes. The nodes with dotted lines indicate the selection over the common nodes. Child 1 and child 2 are the resulting individuals of crossover operation.

Algorithm 5: Proposed crossover operation in 3GP

Data: Parents set $P = \{ind^1, ind^2, \dots, ind^n\}$
Result: Offsprings set O

```

1  $O = \emptyset$ 
2 for all pair  $(P_a, P_b)$  parents in  $P$  do
3    $a = copy(P_a)$ 
4    $b = copy(P_b)$ 
5    $N_a =$  get the nodes in parent  $a$  tree.
6    $N_b =$  get the nodes in parent  $b$  tree.
7   if  $N_a \cap N_b \neq \emptyset$  then
8     Select a random common node between the parent trees  $a$  and  $b$ .
9      $N_c = random(N_a \cap N_b)$ 
10    Select a random node  $N_c$  in the parent tree  $a$ .
11     $sn_a = select(a, N_c)$ 
12    Select a random node  $N_c$  in the parent tree  $a$ .
13     $sn_b = select(b, N_c)$ 
14    Insert the selected node  $sn_a$  in the subtree created by  $b$  in the node  $sn_b$ .
15     $a[sn_a] = subtree(b, sn_b)$ 
16    Insert the selected node  $sn_b$  in the subtree created by  $a$  in the node  $sn_a$ .
17     $b[sn_b] = subtree(a, sn_a)$ 
18    Append the newly created trees to the array of Offsprings  $O$ .
19     $O = O \cup (a, b)$ 
20  end
21 end
Output: The offsprings set  $O$ 

```

Mutation

The mutation operator performs a random change to selected individuals that bring the possibility to cover a broader range on the search space, that it could be better in terms of the fitness value (and therefore, a better solution). Figure 3.4 shows a simple mutation process in an individual. Given an individual to perform a mutation operation, a randomly generated individual has to be created (random derivation tree). Then a crossover process operates over these two individuals, creating a new set of individuals. Algorithm 6 details the mutation operator.

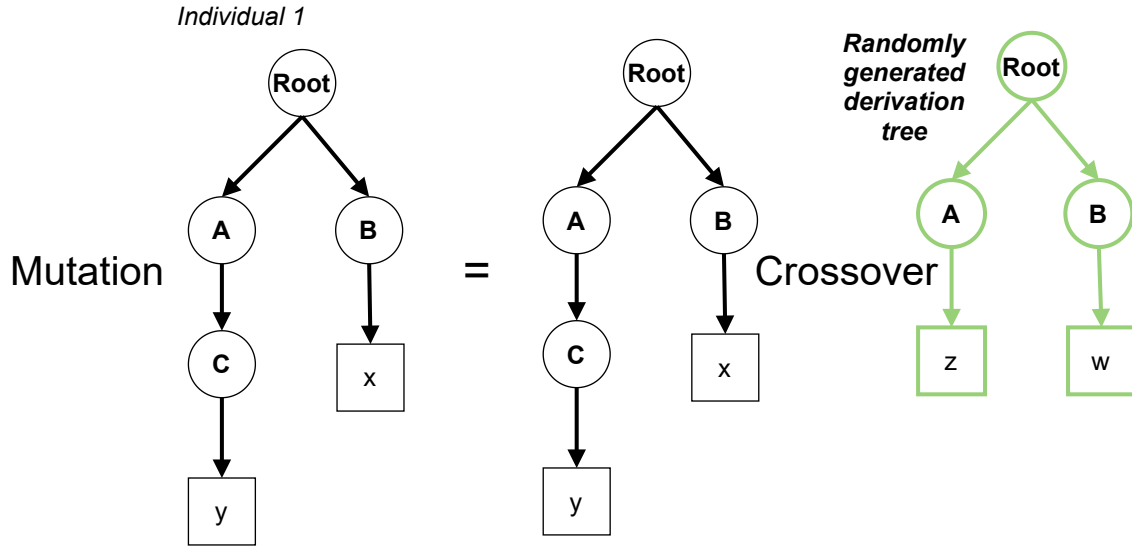


Figure 3.4: Mutation operator

Algorithm 6: Mutation operation algorithm in 3GP**Data:** Individuals to mutate set $M = \{ind^1, ind^2, \dots, ind^m\}$,Context-Free Grammar: CFG Delta mutation: δ_m **Result:** Offsprings set O

```

1  $O = \emptyset$ 
2 for all individual  $ind$  in  $M$  do
3    $n_i =$  get the nodes in the individual tree  $ind$ .
4    $R = \emptyset$ 
5   while  $(n_i \cap nodes(R)) \neq \emptyset$  do
6     Generate a random derivation tree based on the grammar.
7      $R = generate(CFG, 1, \delta_m)$ 
8   end
9   Perform a crossover operation between the random generated tree  $R$  and the
    individual  $ind$ .
10   $O = O \cup Crossover(R, ind)$ 
11 end
Output: The offsprings set  $O$ 

```

3.2 Problem definition

For a better conceptualization of the problem that this work aims to solve, suppose a fuzzy set *hot* designed by an expert. After some optimization process, their initial parameters are changed, but the label is not. To the expert, their perception of *hot*, in a particular domain application, is represented by the initial membership function (defined by their parameters). Still, the semantic meaning of the label may not correspond to the new membership function (after some optimization process). Figure 3.5 shows this idea, where X represents the unknown label that has to be found. This label is getting the proposed methodology.

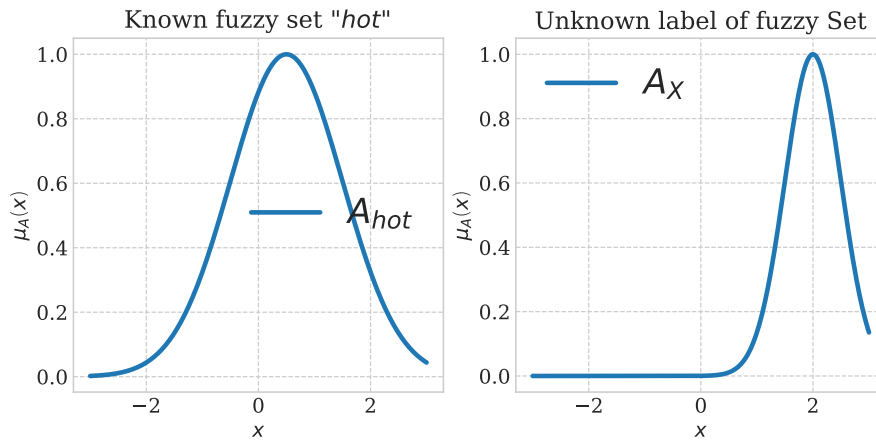


Figure 3.5: Example of an unknown label of the fuzzy set after some optimization process.

The principal advantages of automatically labeling fuzzy sets using the primary terms, those that compose the initial fuzzy sets' labels, are to create an interpretable description using natural language based on prior expert knowledge.

The proposed methodology aims to resolve the problem of finding the best hedge chain to approximate a fuzzy set modified by some process (e.g., by optimization) from an initial fuzzy set defined by the expert domain. A similarity measure compares the initial fuzzy set (\mathcal{S}) modified by hedge transformation function (h) and the fuzzy set in which its parameters have changed due to some process (\mathcal{T}). The formal definition is shown in 3.4.

$$g = \arg \max_{\theta} SIM(h(\mathcal{S}, \Theta), \mathcal{T}) \quad (3.4)$$

Where Θ denotes the universe of linguistic modifiers, h is the hedge transformation function, and SIM is a function that measures the similarity between the source (\mathcal{S}) and target (\mathcal{T}) membership functions. The optimum θ contains an ordered set of chained hedges for *dilation*, *contraction*, *negation* or *shift* operations; all the linguistic terms have arbitrary related values for one of each operation and must be generated by a context-free grammar. The hedge transformation function performs the corresponding operation on the membership function by the assigned values on each term. The result can be treated as a new fuzzy set where is composed of a hedge chain (θ^*) applied to the initial fuzzy set (\mathcal{S}).

The task of finding the best hedge chain overall combinations of linguistic modifiers is considered to be following grammar to avoid incoherent composition such as “*less more below above hot*”. This problem relies on the Grammar-Guided Genetic programming to perform an evolutionary heuristic search of hedge chains that correspond to an established grammar, designed to avoid inconsistent combinations of linguistic modifiers.

3.3 Unary hedge transformation over fuzzy sets

Hedge transformation function

This work defines a hedge transformation function $h(x)$ (equation 3.5) that modifies a membership function with linguistic modifiers such as well known terms: “*very*”, “*more or less*”, “*little*”, etc. Moreover, in this function are considered hedges that shift the membership function on the domain: “*below*”, “*upon*”, etc. The equation’s parameters are p, q, r ; that modify the concentration or dilation, certainty degree and the shift over domain respectively.

$$h(\mu_A, p, q, r) = q\mu_A(x - r)^p \quad (3.5)$$

Where μ_A is a membership function of a fuzzy set A ; $r \in \mathcal{R}$ is the domain shift proportion of the fuzzy set. $p \in \mathcal{R}$ is an exponent that modifies the structure of the membership function; if $p < 1$, then it applies an operation of dilation; otherwise, when $p > 1$, applies an operation of concentration. $q \in [0, 1]$ is a certainty modifier, if $q < 1$ the maximum value of the membership function $\mu_A(x_{max}) = q$.

It is considered a set of linguistic hedges that can shift the support of the membership function over the domain of the fuzzy set. This shift domain term (r) is characterized as a proportion of the domain range that the membership function shifts.

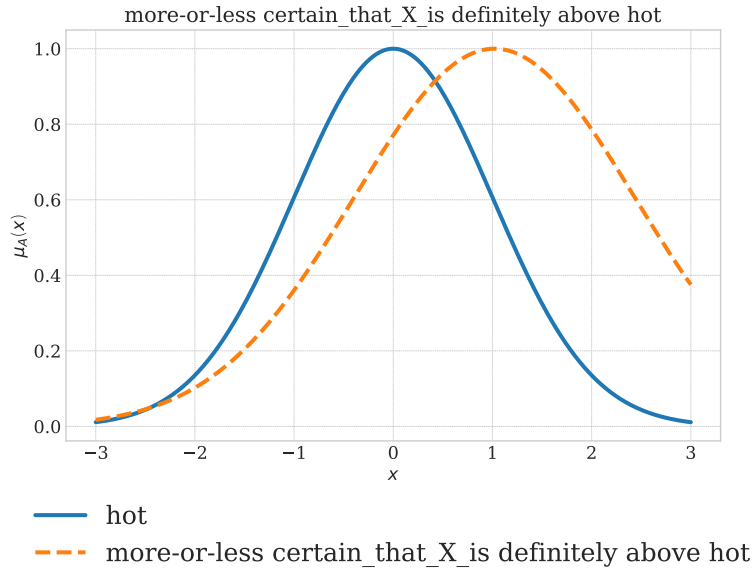


Figure 3.6: Example of shift domain and dilation operation over a fuzzy set.

It is considered a term that modifies the certainty or precision (q) that can limit the maximum value of some membership function. It could provide another dimension of flexibility to represent the modeling of human thinking more accurately.

Each hedge has an associated tuple of values (arbitrarily selected by the expert) that are

evaluated by the hedge transformation function, for example the linguistic modifier *very* has the values $q = 1$, $r = 1.50$, $p = 2$. The selected values for each linguistic term is shown in table 4.1, their visual transformation is shown in figure 3.7.

This work considers three kinds of hedge chains: i) only refers to the precision aspect of the membership function (dilation and concentration operations); ii) sequence of linguistic modifiers referring only to the shift operation (ex. “*very above hot*”); iii) a mixed hedge chain, that applies the corresponding operation to the precision and the shift operations, to identify the segments (that belong to certainty or shift operations) on the chain is used the separator called “*certain that X is*”, in this configuration, it can modify both certainty and shift, such as in the following hedge chain “*more-or-less **certain that X is** very above*”, every linguistic modifier before the separator are evaluated as i), and those that are after it, are evaluated as ii), (see figure 3.6).

The values associated with the linguistic terms are not applied in all scenarios. The shift value (r) only applies when the hedge chain corresponds to the shift operation, which means in those that referred to the certainty aspect (dilation and concentration operations), the shift value is not considered.

Figure 2.4 shows the general flow of the methodology, where the objective is to maximize the defined problem definition in equation 3.4, where the pseudo-best hedge chain is found considering the similarity function described in the equation 3.6. Each hedge chain candidate is evaluated in the fitness function to get their aptitude score. Intuitively, the greater the value is, the better performance is achieved in terms of similarity (or interpretability). As a result, the selected top k candidates that better maximize the fitness function are obtained.

The algorithmic view of the methodology is presented in algorithm 7.

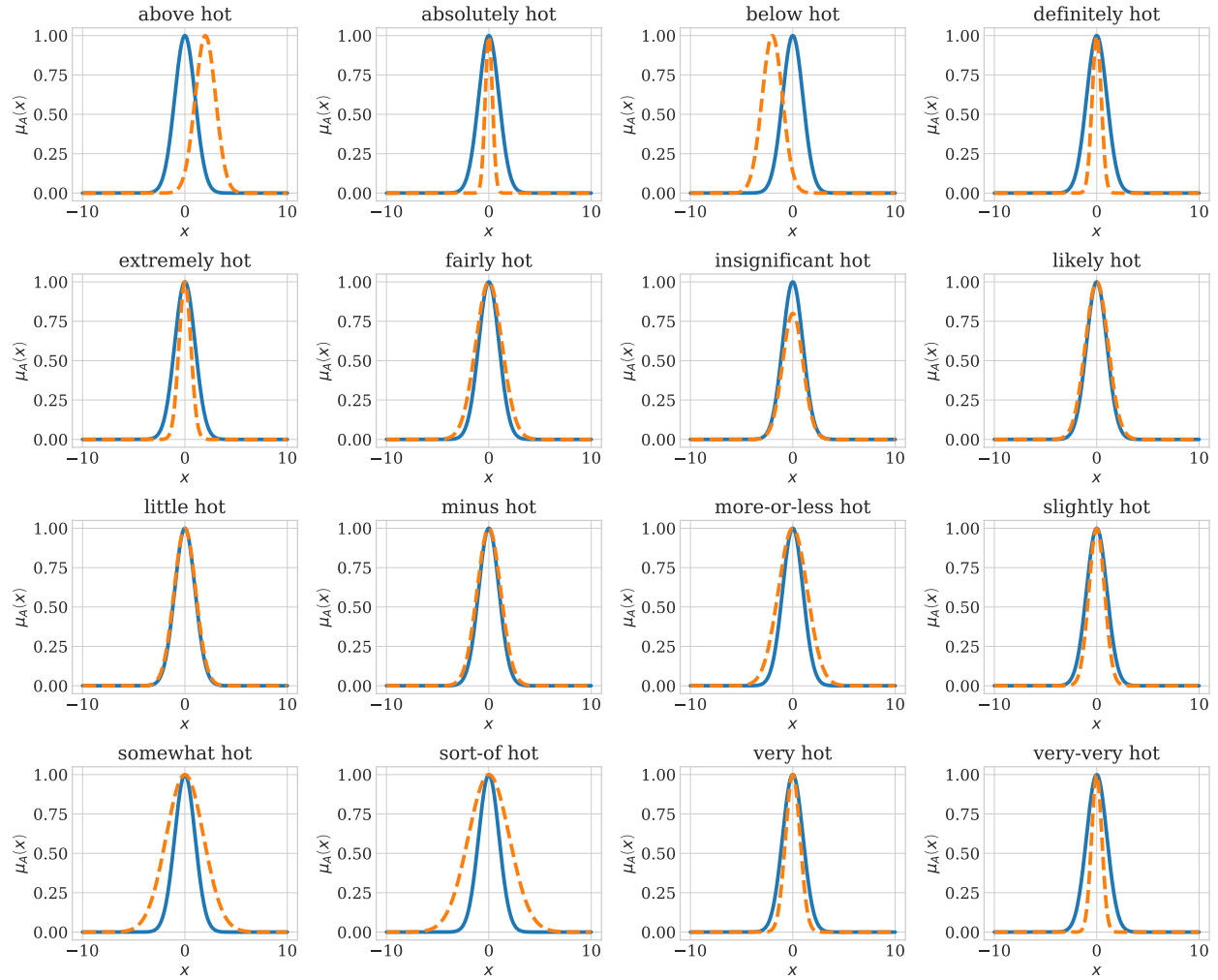


Figure 3.7: All hedge transformations. Blue line represents the original membership function and orange dotted line represents the transformed membership function by the linguistic hedge.

Algorithm 7: Grammar-Guided Genetic Programming Algorithm

Data: Source Fuzzy Set: \mathcal{S} ,

Target Fuzzy Set: \mathcal{T} ,

Number of generations: g ,

Mutation rate: mr ,

Selection percentage: s

Number of best top k candidates

Tolerance error: err

Result: Best hedge candidates (\mathcal{R}) to approximate $hedges(\mathcal{S}) \rightarrow \mathcal{T}$

```

1  $i = 0$ ;
2  $\mathcal{R} = \emptyset$ 
3  $\mathcal{P}$  = Generate an initial population;
4 while  $i < g$  and  $\max(\mathcal{F}) < err$  do
5    $\mathcal{F} = fitness(\mathcal{P}; \mathcal{S}, \mathcal{T})$ ;
6    $\mathcal{S}$  = Select the top  $s$  best candidates based on  $\mathcal{F}$ ;
7    $\mathcal{C} = crossover(\mathcal{S})$ ;
8    $\mathcal{M} = mutate(\mathcal{C}, mr)$ ;
9    $\mathcal{P} = merge(\mathcal{M}, \mathcal{C}, \mathcal{P})$ ;
10   $i = i + 1$ 
11 end
```

Output: The best k candidates of set \mathcal{R}

Fitness function

The presented optimization problem aims to find the (pseudo-)best set of hedges that approximate an initial fuzzy set with a linguistic term defined by the expert to some fuzzy set where its parameter design is changed. A fuzzy similarity measure is used as a criterion for the selection of best hedges candidates, the fuzzy similarity measure (equation 3.6) presented in (Pappis and Karacapilidis, 1993). The fuzzy similarity value refers only to the accuracy aspect of the approximation. The proposed fitness function also considers the “interpretability” of the hedge chain. This term reflects how easily the expert can understand the solutions in this work. In terms of i) fewer terms are easier to read the hedge chain; and ii) as fewer linguistic modifier repetitions occur, the better is the solution.

The function that measures how well structured is the hedge chain (hs) is presented in 3.7. Where the argument $hedges$ corresponds to a hedge chain candidate, it can be defined as an ordered set of terms, f_h refers to a frequency vector of each term that belongs to it.

$$fs(A, B) = \frac{A \cap B}{A \cup B} = \frac{\sum_i^N \min(\mu_A(x_i), \mu_B(x_i))}{\sum_i^N \max(\mu_A(x_i), \mu_B(x_i))} \quad (3.6)$$

$$hs(hedges) = \frac{1}{(f_h^\top \cdot f_h)^{\left(\frac{length(hedges)}{2}\right)}} \quad (3.7)$$

The fitness function proposed involves a fuzzy similarity function, the repeated terms, and the length of the hedge modifier (that can be composed of many linguistic terms).

Formally we define the following fitness function:

$$fitness(hedges; S, T) = (1 + \beta^2) \cdot \frac{fs(h(S, hedges), T) \cdot hs(hedges)}{(\beta^2 \cdot fs(h(S, hedges), T) + hs(hedges))} \quad (3.8)$$

Where $fitness(hedges; S, T)$ is the function that the system has to maximize, the argument $hedges$ is a set containing the hedge chain to apply on the Source membership function (S) T is the target membership function. The function fs returns the similarity value between two given membership functions. The function hs refers to the hedge score, linguistically measures how well is composed a hedge (it is defined as a set of terms), and h is the hedge transformation function that applies the corresponding operations to some fuzzy set given a hedge chain (in this case $hedges$). The fitness function is a general F score, where β weighs fuzzy similarity over hedge score if $\beta < 1$. In the opposite direction, weights hedge score over fuzzy similarity if $\beta > 1$. It might be more relevant to get high accuracy in some application domains than the hedge's terms structure.

Individual encoding

Each individual is encoded by the context-free grammar G_{hedges} and they are composed by a set of linguistic hedges that computes *concentration*, *dilation*, and *shift* operations. There is a separator “*certain that X is*” which delimits which linguistic hedges apply just to the uncertainty or the shift distance over its domain of the fuzzy set. In this work just are considered *concentration* and *dilation* operations to modify their uncertainty (imprecision). A complete sentence that involves concentration, dilation and shift operation, such as: “*more-or-less **certain that X is** very above*” is showed in figure 3.6.

The context-free grammar G_{hedges} , that guides the construction of hedge chain candidates, constructs the three considered kinds of modifiers: i) certainty modifier, ii) shift modifier and iii) hedge chains with certainty and shift modifiers. The context-free grammar is described as $G_{hedges} = (V, \Sigma, R, S)$, where

$$\mathbf{V} = \{ \langle \text{PROPOSITION} \rangle, \langle \text{TRUTH-PROP} \rangle, \langle \text{SHIFT-PROP} \rangle, \langle \text{SHIFT-PLUS} \rangle, \\ \langle \text{SHIFT-MINUS} \rangle, \langle \text{MODIFIER} \rangle, \langle \text{STACK-MOD} \rangle, \langle \text{SHIFT-MOD} \rangle, \langle \text{PLUS-MOD} \rangle,$$

$\langle \text{MINUS-MOD} \rangle, \langle \text{SPLUS-MOD} \rangle, \langle \text{SMINUS-MOD} \rangle, \langle \text{COMPLEMENT} \rangle, \langle \text{NEG} \rangle\}$

$\Sigma = \{\text{above, more, less, little, } \dots \}$

$\mathbf{S} = \{\langle \text{PROPOSITION} \rangle\}$

$\mathbf{R} =$

$\langle \text{PROPOSITION} \rangle ::= \langle \text{COMPLEMENT} \rangle \mid \text{'Is'} \langle \text{TRUTH-PROP} \rangle \text{'certain that'} \langle \text{SHIFT-PROP} \rangle$

$\langle \text{TRUTH-PROP} \rangle ::= \langle \text{MODIFIER} \rangle \langle \text{TRUTH-PROP} \rangle \mid \langle \text{MODIFIER} \rangle$

$\langle \text{SHIFT-PROP} \rangle ::= \langle \text{SHIFT-PLUS} \rangle \mid \langle \text{SHIFT-MINUS} \rangle$

$\langle \text{SHIFT-PLUS} \rangle ::= \langle \text{SPLUS-MOD} \rangle \text{'above'} \mid \langle \text{MODIFIER} \rangle \langle \text{SPLUS-MOD} \rangle \text{'above'} \mid$
 'above'

$\langle \text{SHIFT-MINUS} \rangle ::= \langle \text{SMINUS-MOD} \rangle \text{'below'} \mid \langle \text{MODIFIER} \rangle \langle \text{SPLUS-MOD} \rangle \text{'below'}$
 $\mid \text{'below'}$

$\langle \text{MODIFIER} \rangle ::= \langle \text{PLUS-MOD} \rangle \mid \langle \text{MINUS-MOD} \rangle \mid \langle \text{STACK-MOD} \rangle \langle \text{SPLUS-MOD} \rangle$
 $\langle \text{MODIFIER} \rangle \mid \langle \text{SPLUS-MOD} \rangle \langle \text{MODIFIER} \rangle \mid \langle \text{STACK-MOD} \rangle \langle \text{SMINUS-MOD} \rangle$
 $\langle \text{MODIFIER} \rangle$

$\langle \text{STACK-MOD} \rangle ::= \langle \text{STACK-MOD} \rangle \langle \text{STACK-MOD} \rangle \mid \text{'very'} \mid \langle \text{BLANK} \rangle$

$\langle \text{SHIFT-MOD} \rangle ::= \langle \text{SMINUS-MOD} \rangle \mid \langle \text{SPLUS-MOD} \rangle \mid \langle \text{MODIFIER} \rangle$

$\langle \text{PLUS-MOD} \rangle ::= \text{'absolutely'} \mid \text{'extremely'} \mid \text{'definitely'}$

$\langle \text{MINUS-MOD} \rangle ::= \text{'slightly'} \mid \text{'more-or-less'} \mid \text{'insignificant'} \mid \text{'sort-of'} \mid \text{'fairly'}$
 $\mid \text{'somewhat'} \mid \text{'likely'}$

$\langle \text{SPLUS-MOD} \rangle ::= \langle \text{PLUS-MOD} \rangle \mid \text{'more'}$

$\langle \text{SMINUS-MOD} \rangle ::= \langle \text{MINUS-MOD} \rangle \mid \text{'less'} \mid \text{'little'}$

$\langle \text{COMPLEMENT} \rangle ::= \langle \text{NEG} \rangle \langle \text{TRUTH-PROP} \rangle \mid \langle \text{NEG} \rangle \langle \text{SHIFT-PROP} \rangle$

$\langle \text{NOT} \rangle ::= \text{'not'}$

3.4 Methodology for building Interpretable Mamdani type Neuro-Fuzzy Model

3.4.1 Knowledge base construction

The Fuzzy inference system's construction process consists of domain application fuzzification and fuzzy rules selection. In the domain application fuzzification, each input is fuzzified with Gaussian membership functions equally distributed over a domain. Also, it can be designed by an expert or by clustering.

The initial knowledge base construction must be interpretable due to linguistic modifiers that will adjust their fuzzy sets. If the membership functions are not easily interpretable from the start, they won't be interpretable with the hedge chain applied to them. The following properties are considered to create in an automated way an initial knowledge base that aims to be as interpretable as possible (Gacto et al., 2011): i) Completeness or Coverage, which refers that a membership function should cover every element in the universe of discourse; ii) Normalization, that refers that at least one element in the universe of discourse should have its membership value equal to one; iii) Distinguishability, states that every linguistic term attached to the fuzzy set of a fuzzy variable should have a relevant semantic meaning and easily discriminable from others; iv) Complementarity, this property refers that the sum of membership values for every element in the Universe of Discourse should be close to one.

Every feature in the input space has to be fuzzified, with the primary goal of building an interpretable knowledge base that satisfies the previous properties. According to the elements that the human easily deals with for decision making (Miller, 1956), a low number of fuzzy sets is used, in the range 5 ± 2 , Gaussian membership functions (eq. 3.9) (Wang et al., 1992), and equally distributed on the universe of discourse.

$$f_g(x; m, \sigma) = e^{-\left(\frac{x-m}{2\sigma}\right)^2} \quad (3.9)$$

Where $x \in \mathbb{R}$ is the element to be fuzzified, m is the mean and the core of the membership function, σ is the standard deviation of the function.

The generation of fuzzy rules is a combinatorial problem if all possible fuzzy sets for each input are selected. To avoid this problem, the method proposed in (Wang and Mendel, 1992) selects only a subgroup of the possibilities. All the instances of the dataset set are evaluated in the initial fuzzy sets; the top k fuzzy sets with higher activation value for each output are selected (figure 3.8).

A dataset is characterized by a set of tuples $\{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=0}^Q$, which \mathbf{x}^i is the input vector, and \mathbf{y}^i is the output vector; the value Q is the number of instances that conform the dataset; an instance is an element of the dataset that is referred by the index value i . For every pair, instance i and fuzzy variable j , the maximum membership value is selected.

$$m_{\mathbf{x}}^{i,j} = \max\{\mu_{A_1^j}(\mathbf{x}^i), \dots, \mu_{A_n^j}(\mathbf{x}^i)\} \quad (3.10)$$

In which $m_{\mathbf{x}}^{i,j}$ is the maximum membership value in the set $M(x)$ of fuzzy sets attached to the fuzzy variable j . The compatibility degree ϕ^i denotes the strength of the rule composition for each instance.

$$\phi^i = \prod_{j=1}^{|\mathbf{x}^i|} m_{\mathbf{x}}^{i,j} \quad (3.11)$$

Where ϕ^i is the compatibility grade value of the fuzzy membership values on the instance i , the cardinality of an instance $|\mathbf{x}^i|$ is the number of attributes in the input, for each attribute in the dataset, a fuzzy variable is defined with its domain and fuzzy sets attached to a linguistic

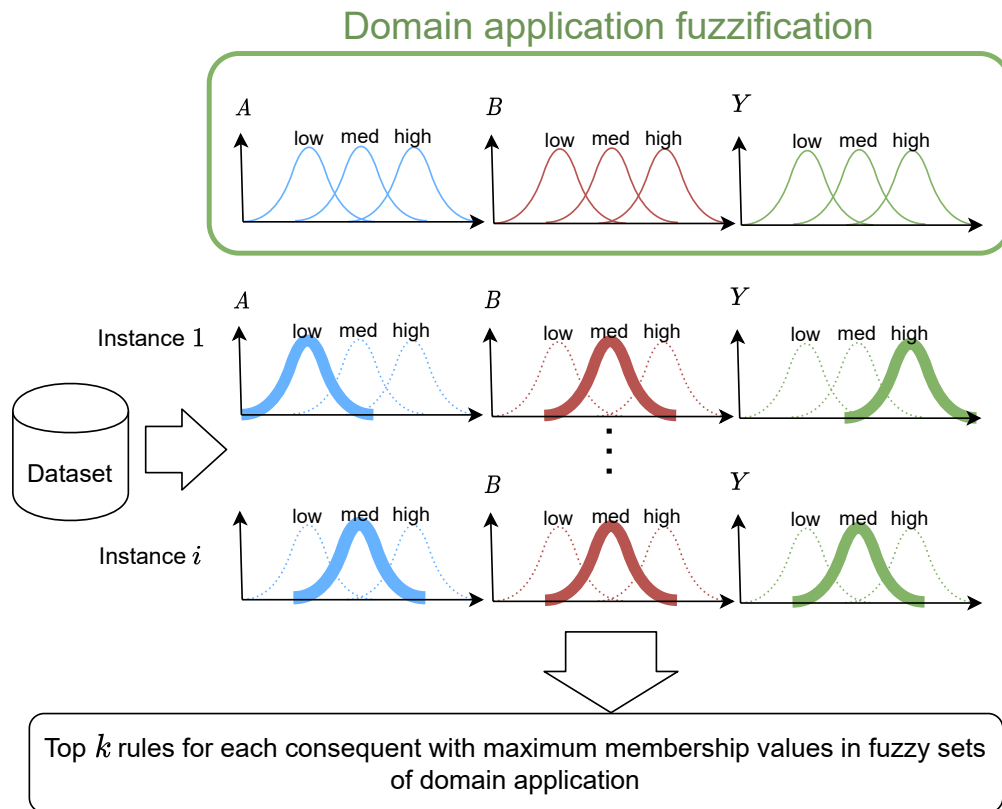


Figure 3.8: The Wang&Mendel procedure selects the k most relevant rule composition by membership belongingness. A and B are fuzzy variables of input feature fuzzification in the antecedent and Y to the consequent. Each fuzzy variable is evaluated by instance, and the higher membership value in the fuzzy set is selected to create a proposition. For instance 1, the rule “IF A is *low* and B is *med* THEN Y is *high*” is selected

term.

The top k composition for each class p is carried out to select the candidate rules to generate the neuro-fuzzy model. Also, the number of fuzzy rules that belong to an output class p can vary for each class. These rules are selected by the maximum firing strength value α^i , which is better fitted to the domain fuzzification.

$$\mathcal{R}^p = \arg \max_{\mathcal{R}' \subseteq \left\{ \left(\max_{A_i^1}, \dots, \max_{A_i^j} \right) \right\}_{i=0}^N, |\mathcal{R}'| \leq k} \{ \phi^i | \mathbf{y}^i = p \}_{i=0}^N \quad (3.12)$$

A fuzzy rule is characterized by a n -tuple $(A^1, \dots, A^{|\mathbf{x}^i|})$, $n = |\mathbf{x}^i|$, in which each element is a fuzzy set A^j that belongs to a universe of a fuzzy variable ($A^j \in M(x)$), denoted by the identifier j ; the logical operator by default is set to **and** ($\tilde{*}$). In the form of Zadeh's fuzzy rule, it can be read as “ IF x_1^i is A^1 and \dots and x_n^i is A^n THEN y^i is G^p ”.

3.4.2 Neuro-fuzzy model optimization

Once the knowledge base is designed, some optimization methods can adjust their membership function parameters to fit the data better.

A neuro-fuzzy model is created from the structure of the fuzzy inference system to perform an optimization process of the membership functions parameters. This architecture comprises five layers: input, fuzzification, inference, implication, and defuzzification layer. The connections between the layers are not fully connected, as is common in the traditional neural network architecture. In the fuzzification layer are only connected the membership functions belonging to the input domain. Also, they are not completely connected neurons in the implication layer, as Zadeh's fuzzy rule fixes their connections. Figure 3.9 shows a visual representation of a Mamdani-type neuro-fuzzy model.

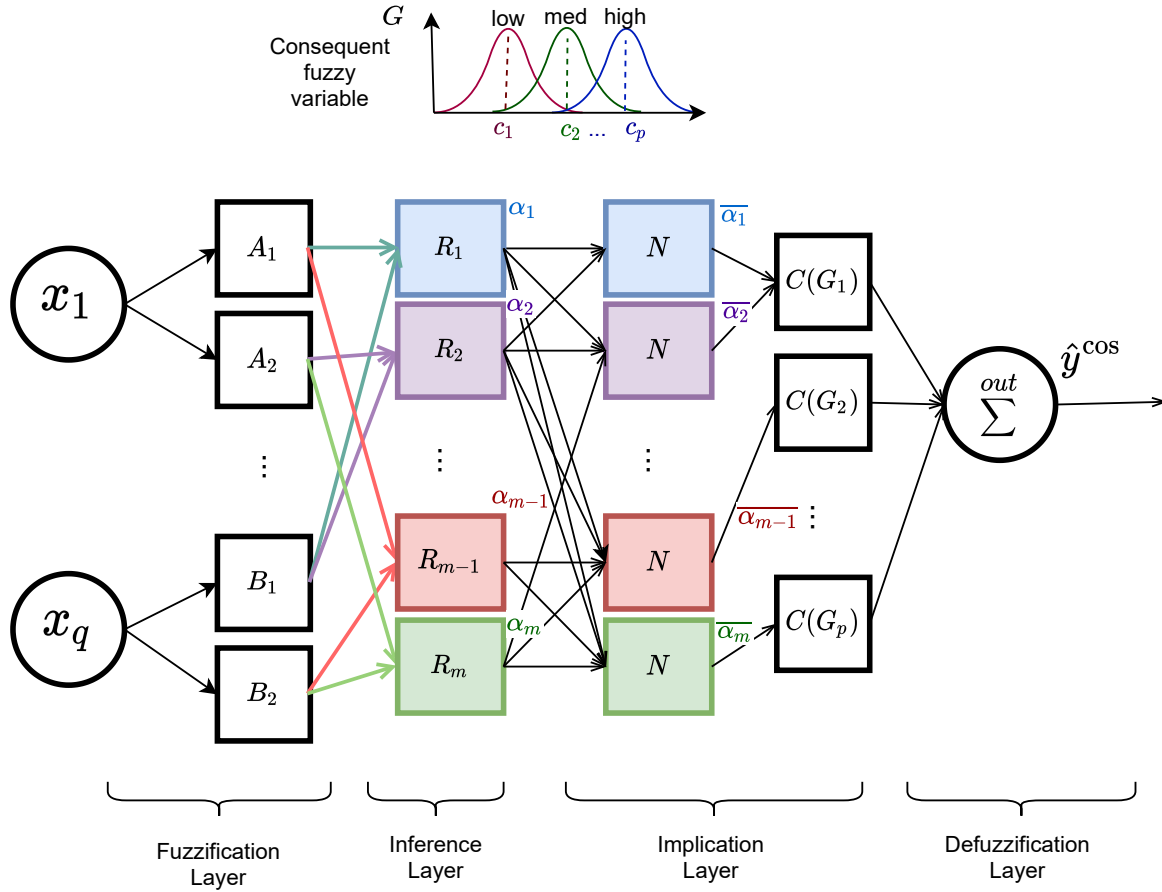


Figure 3.9: The Mamdani Neuro-fuzzy representation is composed by a non-fully connected 5-layer artificial neural network, all the computations corresponds to the involved operations in a Mamdani-type fuzzy inference system for q inputs, m rules, p fuzzy sets in consequent with Center-of-Sets defuzzification method. $C(G^i)$ denotes the centroid of consequent G^i .

The parameters to optimize the model belong to the antecedent part of the fuzzy rule once the parameters are selected to achieve better performance to adjust the data.

The neural architecture is shown in the figure 3.9. The first layer is non-fully connected among neurons that represent the fuzzification process.

$$f^{A_k^j}(x_i) = \mu_{A_k^j}(x_i) \quad (3.13)$$

Where $A_r^j \in V_k$ is a fuzzy set that belongs to the fuzzy variable V_k , the domain of each fuzzy variable is shared by its corresponding attribute domain in the dataset. Only the membership functions directly related to the attribute are evaluated by the input value, which results in a semi-connected layer.

The inference layer is also a non-fully connected layer that generates a firing strength value. The implication operation is calculated a *t-norm* ($\tilde{*}$) as a product.

$$\alpha^l(x_i) = T_{r=1}^p f^{A_r}(x_i) \quad (3.14)$$

The implication layer performs a normalization operation that conforms a step to the defuzzification process.

$$\bar{\alpha}^l(x_i) = \frac{\alpha^l(x_i)}{\sum_{j=1}^L \alpha^j(x_i)} \quad (3.15)$$

After the normalization process, for each rule that has a consequent (G_i), a $\tilde{*}$ as the product is calculated.

$$z^j(x_i) = S_{l=1}^r \{c^j \times \bar{\alpha}^l(x_i)\} \quad (3.16)$$

Where c^j corresponds to the centroid of the consequent value G_j that multiplies the firing strength of each r -rule associated with this consequent; if the membership function that describes the consequent is a Gaussian function, then c^j is represented by the mean value.

The center-of-set defuzzification method is used to get the crisp output value from the inference process. This method is implicit in the neuro-fuzzy model; due to the consequent part being designed by Gaussian membership functions, the calculation can be expressed as:

$$\hat{y}^{COS}(\mathbf{x}') = \frac{\sum_{l=1}^m COG(G^l) \alpha^l(\mathbf{x}')}{\sum_{l=1}^m \alpha^l(\mathbf{x}')} = \frac{\sum_{l=1}^m c^l \alpha^l(\mathbf{x}')}{\sum_{l=1}^m \alpha^l(\mathbf{x}')} \quad (3.17)$$

Which COG is the center of gravity of the membership function, m is the number of fuzzy rules in the FIS, \mathbf{x}' is an arbitrary input crisp value to perform the inference and defuzzification process; c^l is the center of the l th consequent set; α^l is the firing level of the rule.

The output layer performs an aggregation operation of the resulting implications. This aggregation operation is performed by a s -norm ($\tilde{+}$) as a sum.

$$\hat{y}_i = S_{j=1}^p z^j(x_i) \quad (3.18)$$

In where p is the number of consequents that are involved in the fuzzy inference system design.

This neuro-fuzzy model adjusts its parameters to adjust a given dataset better. The sum of square errors (sse) is used to measure the error of the signal.

$$sse(y, \hat{y}) = \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (3.19)$$

Where N is the number of instances on the batch, y is the target value, and \hat{y} is the predicted value by the model.

The trainable parameters on the neuro-fuzzy model are the design parameters of the fuzzy sets; if Gaussian functions define the membership functions, then the trainable parameters are the mean and σ values, where $\sigma > 0$.

A gradient descent optimization method is applied to find the pseudo-best parameters. The learning rule is shown in equation 3.20

$$\theta^{\text{new}} = \theta^{\text{old}} - \eta \nabla E(\theta^{\text{old}}) \quad (3.20)$$

Where θ are the parameter vector values; $E(\theta)$ is the gradient of the error value of the model with the parameters θ ; η is the learning rate value in $0 < \eta < 1$.

The hyper-parameters of the model are:

- **Number of fuzzy sets for each attribute:** each attribute in the dataset is represented by a fuzzy variable, so the number of fuzzy sets in each fuzzy variable has to be selected.
- **Number of rules:** the selection of the antecedents, that is, the combination by a *t-norm* operation of fuzzy sets of different fuzzy variables, is an exponential growth problem. Because of that, this parameter has to be carefully selected; lower values are preferred. This parameter is related to the value k of the Wang and Mendel procedure, described in section 3.4.1.
- **Learning rate value:** this value scales the directional vector generated by gradient calculation; as the lower the value is, the better search is but slower. Usually, the default value is set to a value of 0.01.
- **Batch size:** the selection of dataset partition to train the model is set by this value.

- **Epoch number:** an epoch represents an entire iteration overall dataset (train dataset partition).
- **Goal error value:** is a threshold value to consider to stop the training process.

3.4.3 Linguistic granule optimization for semantic enhancement of pseudo-optimal knowledge base using binary relationships

After the optimization process of the fuzzy inference system, the initial parameters of the fuzzy sets are changed. Therefore, the initial meaning of the linguistic term attached to the membership function has also changed. To recover the semantic meaning, a collection of linguistic terms has to find it that describes the new fuzzy sets form that better adjust the data. However, in some way, the new linguistic labels have to be linked to the prior knowledge, implicitly in the fuzzy inference system design, to maintain the context.

The proposed evolutionary methodology is used to find the best linguistic modifiers to adjust the previous membership function form to the optimized ones. The linguistic hedge types considered in this work to transform fuzzy variables are identity, unary, and binary. Here, a grammar is proposed to generate those linguistic hedges, where the pseudo best candidates are searched by a genetic algorithm guided by grammar.

Hedge transformation function

A unary hedge transformation functions $h(x)$ is used to perform modifications on the membership function. This function can effectuate concentration, dilation, and translation operations.

$$h(A; p, r, n) = \begin{cases} \mu_A(x - r)^p & \text{if } n = 0 \\ \mu_{\bar{A}}(x - r)^p & \text{if } n = 1 \end{cases}$$

Where $A = \{(x, \mu_A(x)) | \mu_A \in [0, 1]\}$ is a fuzzy set, $r \in \mathbb{R}$, $n \in \{0, 1\}$, and $p \in \mathbb{R}$.

Where r is a shift domain value; n is a binary value, if it is zero, the modification operates with the original membership function; otherwise, apply to its complement; p is an exponential value, where $0 < p < 1$ generates a dilation modification, if $p > 1$, then a concentration operation is applied to the membership function.

For binary hedge chains, a binary hedge function $binhedge((v_j, v_k), (h_1, h_2, \dots))$ that carry out the binary operations over two fuzzy sets v_j, v_k is proposed.

$$binhedge((v_j, v_k); \mathcal{H}) = \begin{cases} v_j^{left} \cup v_k^{right} & \text{if } \mathcal{H} \text{ relates to BETWEEN operation} \\ v_j \cup v_k & \text{if } \mathcal{H} \text{ relates to OR operation} \\ \bar{v}_j \cup \bar{v}_k & \text{if } \mathcal{H} \text{ relates to NEITHER operation} \\ h(v_j; \mathcal{H}_p, \mathcal{H}_r, \mathcal{H}_n) & \text{if } \mathcal{H} \text{ relates to UNARY operation} \end{cases}$$

Where \mathcal{H} is the universe of hedge chains, v_j^{left} refers to all membership values of v_j that are at the left of the core; in the opposite direction v_k^{right} . In figure 3.10 is shown a visual representation of the binary hedge function *between* over two fuzzy sets.

To find the pseudo-best hedge chain that transforms the initial knowledge base to the optimized one is performed by a Genetic Algorithm. Individuals are derivation trees of the proposed grammar that can change each fuzzy set of a given fuzzy variable. The initial knowledge context is considered to describe the newly arisen concepts (adjusted parameters' design of membership functions). The proposed grammar can be regarded as upon two fuzzy sets as context to build new linguistic descriptors. The selection of these two candidates is based on the distance by the core of membership functions. When the closest two fuzzy

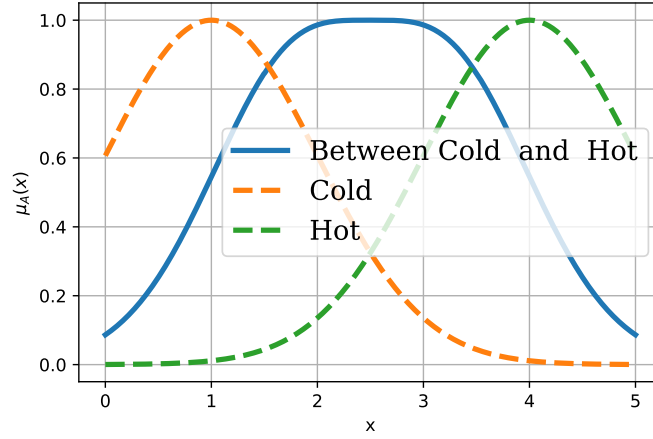


Figure 3.10: The proposed binary linguistic hedge *between* is applied over the two initial fuzzy sets *Cold* and *Hot*

sets are selected, they are given to the genetic optimization algorithm to find the best hedge individuals to describe the new concept better, considering their similarity. The algorithm to describe this process is shown in algorithm 8.

Algorithm 8: Grammar-based Optimization for Interpretable Neuro-fuzzy model.

Data: Initial Fuzzy Variable \mathcal{V} , Optimized Fuzzy Variable \mathcal{V}^* , Hedge candidate set \mathcal{H}

Result: Best hedge candidates to approximate $\text{hedge}(\mathcal{V}) \rightarrow \mathcal{V}^*$

```

1  $i = 0$ ;
2 while there is an  $v^* \in \mathcal{V}^*$  without label do
3   Find the pair  $(v_j, v_k) \in \mathcal{V}$ , closest to  $v_i^*$  by the mf's core;
4   Perform grammar-based optimization of  $(h_1, h_2, \dots) \in \mathcal{H}$  such that
      $\text{binhedge}((v_j, v_k), (h_1, h_2, \dots)) \approx v_i^*$ ;
5    $i = i + 1$ 
6 end
```

A grammar-based genetic algorithm is used to find the best coverage chain candidates, these algorithms not only represent each individual as a graph, but their structure is constrained by a context-free grammar (Manrique et al., 2009). This property allows it to generate only interpretable linguistic modifiers. The processes that are defined in the algorithm are described in algorithm 9.

Algorithm 9: Grammar-Guided Genetic Programming Algorithm

Data: Tuple of initial fuzzy sets: $(v_j, v_k) \in \mathcal{V}$,

Unknown fuzzy set: v_i^* ,

Number of generations: g ,

Mutation rate: mr ,

Selection percentage: s

Number of best top k candidates

Result: Best hedge chain individuals (\mathcal{I}_H) with highest similarity score
 $\text{sim}(\text{hedges}([v_j, v_k], \mathcal{I}_H), v_i^*)$

```

1   $i = 0$ ;
2   $\mathcal{I}_H = \emptyset$ 
3   $\mathcal{P}$  = Generate an initial population;
4  while stop criteria are not reached do
5       $\mathcal{F} = \text{sim}(\text{hedges}([v_j, v_k], \mathcal{P}), v_i^*)$ ;
6       $\mathcal{T} = \text{top}(\mathcal{F}, s)$ ; Select the top  $s$  best candidates based on the aptitude of  $\mathcal{F}$ 
        individuals.
7       $\mathcal{C} = \text{crossover}(\mathcal{T})$ ; Perform crossover by compatibility nodes on the population.
8       $\mathcal{M} = \text{mutate}(\mathcal{C}, mr)$ ; Generate a random individual then apply crossover to  $mr$ 
        percentage of population.
9       $\mathcal{P} = \mathcal{M} + \mathcal{C} + \mathcal{P}$ ; Merge individuals to create a new population.
10      $i = i + 1$ 
11 end
12  $\mathcal{I}_H = \text{top}(\mathcal{P}, k)$  select top  $k$  best individuals in population  $\mathcal{P}$ .
Output: The best  $k$  individuals of set  $\mathcal{I}_H$ 

```

Fitness function

The formalization of hedge function that is considered to operate over the linguistic variables is described as follows:

$$f_{hedge} : V \times \mathcal{H} \rightarrow \mathcal{X} \quad (3.21)$$

Where V is the fuzzy variable to transform, \mathcal{H} is the set of all hedge chains that the grammar can generate, and \mathcal{X} is the fuzzy set domain.

The similarity function, described in (Pappis and Karacapilidis, 1993) (equation 3.22), is used to measure how well the old fuzzy sets with linguistic transformation approximate the new ones.

$$sim(A, B) = \frac{A \cap B}{A \cup B} = \frac{\sum_i^N \min(\mu_A(x_i), \mu_B(x_i))}{\sum_i^N \max(\mu_A(x_i), \mu_B(x_i))} \quad (3.22)$$

Where A and B are fuzzy sets, μ_A and μ_B their respective membership functions; The higher the similarity between fuzzy sets, the closer the resulting value is to 1; otherwise, the value is closer to 0.

Individual encoding

In the proposed context-free grammar, three kinds of relationships based on their arity are considered. The association relationships in this context are treated as transformation functions with hedges, so the three main categories are binary hedge chain, unary hedge chain, and identity. Some examples of the hedge chain generated by the proposed grammar are listed as follows.

Unary hedge chain:

- Extremely A_2
- Kind of A_1
- Higher than Very A_1

Binary hedge chain:

- Between A_1 and A_2
- Lower than Between A_1 and A_2
- Higher than Very A_1 and A_2
- Neither Very A_1 nor Exactly A_2

Identity hedge chain:

- A_1, A_2

The elements A_1 and A_2 represent the fuzzy sets that are involved in the hedge chain. In the binary hedge chain category, both elements appear. On the other hand, the rest of the hedge chain only have one of them associated. It is important to notice that when the hedge chain is put in context, the linguistic terms attached to the fuzzy sets replace the generic notation A_i .

The free-context grammar G part of each linguistic variable is described as $G = (V, \Sigma, R, S)$, where

$$V = \{ \langle PROPOSITION \rangle, \langle MOD_BINARY_HEDGE_CHAIN \rangle, \langle BINARY_HEDGE_CHAIN \rangle, \\ \langle UNARY_HEDGE_CHAIN \rangle, \langle UNARY_HEDGE_CHAIN1 \rangle, \langle UNARY_HEDGE_CHAIN2 \rangle, \\ \langle SUPERLATIVE_HEDGE_CHAIN \rangle, \langle HEDGE_CERTAINTY1 \rangle, \langle HEDGE \rangle, \langle HEDGE1 \rangle, \langle HEDGE2 \rangle, \\ \langle NEG \rangle, \langle UNARY_HEDGE_CHAIN1_NN \rangle, \langle UNARY_HEDGE_CHAIN2_NN \rangle, \langle HEDGE1_NN \rangle,$$

$$\langle HEDGE2_NN \rangle \}$$

$$\Sigma = \{ \text{"Lower than"} , \text{"Higher than"} , \text{"Between"} , \text{" and"} , \text{" or"} , \text{"Neither"} , \text{" nor"} , \text{"Upper than"} , \text{"Exactly"} , \text{"More-or-less"} , \text{"Kind of"} , \text{"Very"} , \text{"Extremely"} , \text{"<A1>"} , \text{"<A2>"} , \text{"not"} , \dots \}$$

$$S = \{ \langle PROPOSITION \rangle \}$$

$$\langle PROPOSITION \rangle ::= \langle MOD_BINARY_HEDGE_CHAIN \rangle$$

$$| \langle UNARY_HEDGE_CHAIN \rangle$$

$$| \langle HEDGE \rangle \langle MOD_BINARY_HEDGE_CHAIN \rangle ::= \text{Lower than} \langle BINARY_HEDGE_CHAIN \rangle$$

$$| \text{Higher than} \langle BINARY_HEDGE_CHAIN \rangle$$

$$| \langle BINARY_HEDGE_CHAIN \rangle$$

$$\langle BINARY_HEDGE_CHAIN \rangle ::= \text{Between} \langle UHC_HEDGE1 \rangle \text{ and } \langle UHC_HEDGE2 \rangle$$

$$| \langle UHC_HEDGE1 \rangle \text{ or } \langle UHC_HEDGE2 \rangle$$

$$| \text{Neither} \langle UNARY_HEDGE_CHAIN1_NN \rangle \text{ nor } \langle UNARY_HEDGE_CHAIN2_NN \rangle$$

$$\langle UNARY_HEDGE_CHAIN \rangle ::= \langle UNARY_HEDGE_CHAIN1 \rangle$$

$$| \langle UNARY_HEDGE_CHAIN2 \rangle$$

$$| \langle HEDGE \rangle$$

$$\langle UNARY_HEDGE_CHAIN1 \rangle ::= \text{Upper than} \langle HEDGE_CERTAINTY1 \rangle$$

$$| \text{Lower than} \langle HEDGE_CERTAINTY1 \rangle$$

$$| \text{Exactly} \langle HEDGE1 \rangle$$

$$| \text{More-or-less} \langle HEDGE1 \rangle$$

$$| \text{Kind of} \langle HEDGE1 \rangle$$

$$| \text{Very} \langle HEDGE1 \rangle$$

$$| \text{Extremely} \langle HEDGE1 \rangle$$

$\langle UNARY_HEDGE_CHAIN2 \rangle ::=$ Upper than $\langle HEDGE_CERTAINTY2 \rangle$
 | Lower than $\langle HEDGE_CERTAINTY2 \rangle$
 | Exactly $\langle HEDGE2 \rangle$
 | More-or-less $\langle HEDGE2 \rangle$
 | Kind of $\langle HEDGE2 \rangle$
 | Very $\langle HEDGE2 \rangle$
 | Extremely $\langle HEDGE2 \rangle$ $\langle SUPERLATIVE_HEDGE_CHAIN \rangle ::=$ Highest
 | Lowest

$\langle UHC_HEDGE1 \rangle ::= \langle UNARY_HEDGE_CHAIN1 \rangle$
 | $\langle A1 \rangle$

$\langle UHC_HEDGE2 \rangle ::= \langle UNARY_HEDGE_CHAIN2 \rangle$
 | $\langle A2 \rangle$

$\langle HEDGE_CERTAINTY1 \rangle ::= \langle A1 \rangle$
 | Exactly $\langle HEDGE1 \rangle$
 | More-or-less $\langle HEDGE1 \rangle$
 | Kind of $\langle HEDGE1 \rangle$
 | Very $\langle HEDGE1 \rangle$
 | Extremely $\langle HEDGE1 \rangle$

$\langle HEDGE_CERTAINTY2 \rangle ::= \langle A2 \rangle$
 | Exactly $\langle HEDGE2 \rangle$
 | More-or-less $\langle HEDGE2 \rangle$
 | Kind of $\langle HEDGE2 \rangle$
 | Very $\langle HEDGE2 \rangle$
 | Extremely $\langle HEDGE2 \rangle$

$\langle HEDGE \rangle ::= \langle HEDGE1 \rangle$
 | $\langle HEDGE2 \rangle$

$$\langle HEDGE1 \rangle ::= \langle A1 \rangle$$

$$| \quad \langle NEG \rangle \langle A1 \rangle$$

$$\langle HEDGE2 \rangle ::= \langle A2 \rangle$$

$$| \quad \langle NEG \rangle \langle A2 \rangle$$

$$\langle NEG \rangle ::= \text{not}$$

$$\langle UNARY_HEDGE_CHAIN1_NN \rangle ::= \text{Upper than } \langle HEDGE1_NN \rangle$$

$$| \quad \text{Lower than } \langle HEDGE1_NN \rangle$$

$$| \quad \text{Exactly } \langle HEDGE1_NN \rangle$$

$$| \quad \text{More-or-less } \langle HEDGE1_NN \rangle$$

$$| \quad \text{Kind of } \langle HEDGE1_NN \rangle$$

$$| \quad \text{Very } \langle HEDGE1_NN \rangle$$

$$| \quad \text{Extremely } \langle HEDGE1_NN \rangle$$

$$| \quad \langle HEDGE1_NN \rangle$$

$$\langle UNARY_HEDGE_CHAIN2_NN \rangle ::= \text{Upper than } \langle HEDGE2_NN \rangle$$

$$| \quad \text{Lower than } \langle HEDGE2_NN \rangle$$

$$| \quad \text{Exactly } \langle HEDGE2_NN \rangle$$

$$| \quad \text{More-or-less } \langle HEDGE2_NN \rangle$$

$$| \quad \text{Kind of } \langle HEDGE2_NN \rangle$$

$$| \quad \text{Very } \langle HEDGE2_NN \rangle$$

$$| \quad \text{Extremely } \langle HEDGE2_NN \rangle$$

$$| \quad \langle HEDGE2_NN \rangle$$

$$\langle HEDGE1_NN \rangle ::= \langle A1 \rangle$$

$$\langle HEDGE2_NN \rangle ::= \langle A2 \rangle$$

A visual example of the resulting individuals, starting with an initial fuzzy variable $input_1$ (see figure 3.11a), after a process of optimization, some changes may affect parameters' design

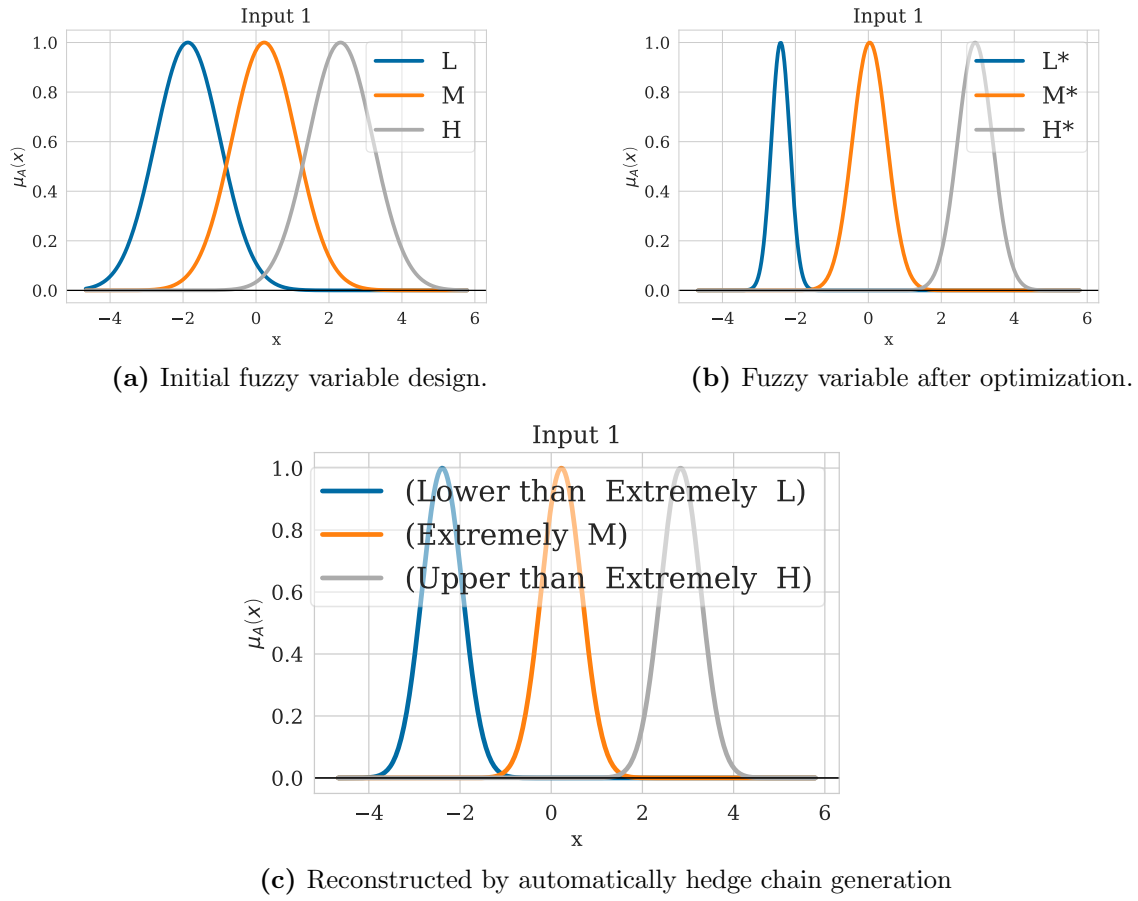


Figure 3.11: In a) All their fuzzy sets are equally distributed and interpretable. In b) Its initial design changed due to the parameter tuning process, hence the semantic meaning. In c) is the reconstructed fuzzy variable from the initial to the optimized one by automatically hedge chain generation.

(figure 3.11b). The goal of the proposed methodology is to approximate the optimized model by initial knowledge context using hedge chains (see figure 3.11c).

Chapter 4

Experimentation and results

4.1 Semantic enhancement of fuzzy sets using unary hedge transformations

The presented experimentation comprises two case studies; 1) Finding the best set of hedges to modify a fuzzy set to a target fuzzy set. Both membership functions are of the same kind. 2) Finding the best set of hedges to approximate a source fs to target fs where the membership functions are of a different kind. The fuzzy set design was made by the FuzzSystem python framework ¹. NLTK python framework (Bird et al., 2009) was used for grammar processing, transformation, and generation of derivation trees.

All cases propose a set of test scenarios in which they simulated the following: given a fuzzy set by an expert, in this domain application is related to the concept “*hot*” (that is called source membership function \mathcal{S}). Some optimization process is applied to \mathcal{S} that its parameters change (that is called target membership function \mathcal{T}). The related initial linguistic term “*hot*” does not represent the actual expert perception. In consequence, taken

¹<https://github.com/Raul-Navarro/fuzzy-framework>

as reference the initial expert conceptualization, the method searches the (pseudo-)best hedge chain that approximates the initial fuzzy set (\mathcal{S}) to the modified one (\mathcal{T}).

In section 3.4 membership function parametrization for Gaussian, Triangular, and Trapezoidal are described. The linguistic terms used for both experiments are the same and arbitrarily valued (table 4.1), however, the semantic meaning is correlated by the operations of *dilation* and *concentration*, where the *dilation* operation refers to the increasing of uncertainty while the *concentration* operation refers to the reduction of uncertainty.

In both experiment scenarios, a variation of the β value is performed. The β value represents the importance of accuracy and “interpretability”; the lower the value, the more focus on accuracy is given. In the opposite direction, the higher the β value is, the more importance gives to the interpretability. Figure 4.5 shows the overall behavior of this parameter in both cases, where the source and target membership functions are of the same kind, and figure 4.6 shows the membership functions are of a different kind.

Table 4.1: Summary of the proposed linguistic terms, certainty, shift, hedge modifier value used in the cases studies.

Linguistic term	certainty modifier (q)	shift modifier (r)	hedge modifier (p)
very	1	1.50	2
minus	1	0.50	0.75
little	1	0.70	0.9
slightly	1	0.90	1.7
extremely	1	3	3.5
somewhat	1	0.80	0.3
very-very	1	3	4
absolutely	1	4.5	8
above	1	0.1	1
below	1	-0.1	1
more-or-less	1	0.70	0.5
definitely	1	1.70	5
insignificant	0.9	1	1
more	1	1.2	1
less	1	0.20	1
likely	1	0.95	0.8
fairly	1	0.85	0.6
sort-of	1	0.75	0.25

The parameters used in the optimization method using GGGP, described in section ??, are presented in table 4.2.

Table 4.2: Parameters used in the Grammar-Guided Genetic Programming in each step of the methodology.

Parameter	Value
Number of population individuals	100
Selection method	Elitist
Selection percentage	30%
Mutation method	One point (node)
Mutation percentage	20%
Generations	500
β value in the fitness function	0.01, 0.05, 0.1, 0.5, 0.7, 0.99

4.1.1 Finding linguistic modifiers for the approximation of same kind membership functions

In this study case, we establish eight fuzzy sets, where its label represents the arbitrary concept “hot” in some application domain, the selected membership functions are *Trapezoidal*, *Gaussian*, and *Triangular*. Both source fuzzy set and target fuzzy set have the same membership functions, and they only differ by their parameters.

The specific membership function parameters and the similarity obtained after the optimization process, with value $\beta = 0.01$, that refers to an accuracy preference, are shown in table 4.3. The same collection of the fuzzy sets are tested to find the best hedge candidates, but with $\beta = 0.99$, which refers to an “interpretability” preference is shown in the table 4.4. The parameters order in the table correspond to the design parameters described in section 3.4. Trapezoidal $[a, b, c, d]$, Triangular $[a, b, c]$, and Gaussian $[\sigma, m]$.

Table 4.3: Case study results where the source and target membership functions are of the same type, and β with value 0.01.

Membership function	source MF parameters	target MF parameters	similarity	hedge score
Trapezoidal MF	[0, 1, 2, 3]	[5, 6, 7, 8]	0.973	0.167
Trapezoidal MF	[7, 8, 9, 10]	[0, 1, 2, 3]	0.971	0.071
Gaussian MF	[1, 0]	[2, 2]	0.989	0.286
Triangular MF	[0, 1, 2]	[3, 6, 9]	0.536	0.200
Gaussian MF	[1, 1]	[5, 6]	0.936	0.125
Gaussian MF	[5, 8]	[1, 1]	0.926	0.020
Gaussian MF	[5, 8]	[8, 1]	0.995	0.042
Triangular MF	[3, 7, 9]	[1, 2, 3]	0.757	0.286
Mean			0.885	0.150
σ			0.161	0.104
σ^2			0.026	0.011

Table 4.4: Case study results where the source and target membership functions are of the same type, and β with value 0.99.

Membership function	source MF params	target MF params	similarity	hedge score
Trapezoidal MF	[0, 1, 2, 3]	[5, 6, 7, 8]	0.600	1.000
Trapezoidal MF	[7, 8, 9, 10]	[0, 1, 2, 3]	0.891	0.500
Gaussian MF	[1, 0]	[2, 2]	0.794	1.000
Triangular MF	[0, 1, 2]	[3, 6, 9]	0.455	0.667
Gaussian MF	[1, 1]	[5, 6]	0.869	1.000
Gaussian MF	[5, 8]	[1, 1]	0.535	1.000
Gaussian MF	[5, 8]	[8, 1]	0.796	1.000
Triangular MF	[3, 7, 9]	[1, 2, 3]	0.754	0.400
Mean			0.712	0.821
σ			0.161	0.258
σ^2			0.026	0.066

The visualization of the source, target, and resulting membership function for each case is shown in the following images grouped by its membership function type: *Trapezoidal mf*) figure 4.1; *Gaussian mf*) figure 4.2 and figure 4.3; *Triangular mf*) figure 4.4.

4.1.2 Finding hedges for the approximation of different kind membership functions

In this study case, we establish eight fuzzy sets, where its label represents the arbitrary concept “hot” in some application domain, the selected membership functions are *Trapezoidal*, *Gaussian*, and *Triangular*. The source and target fuzzy sets have different kinds of membership

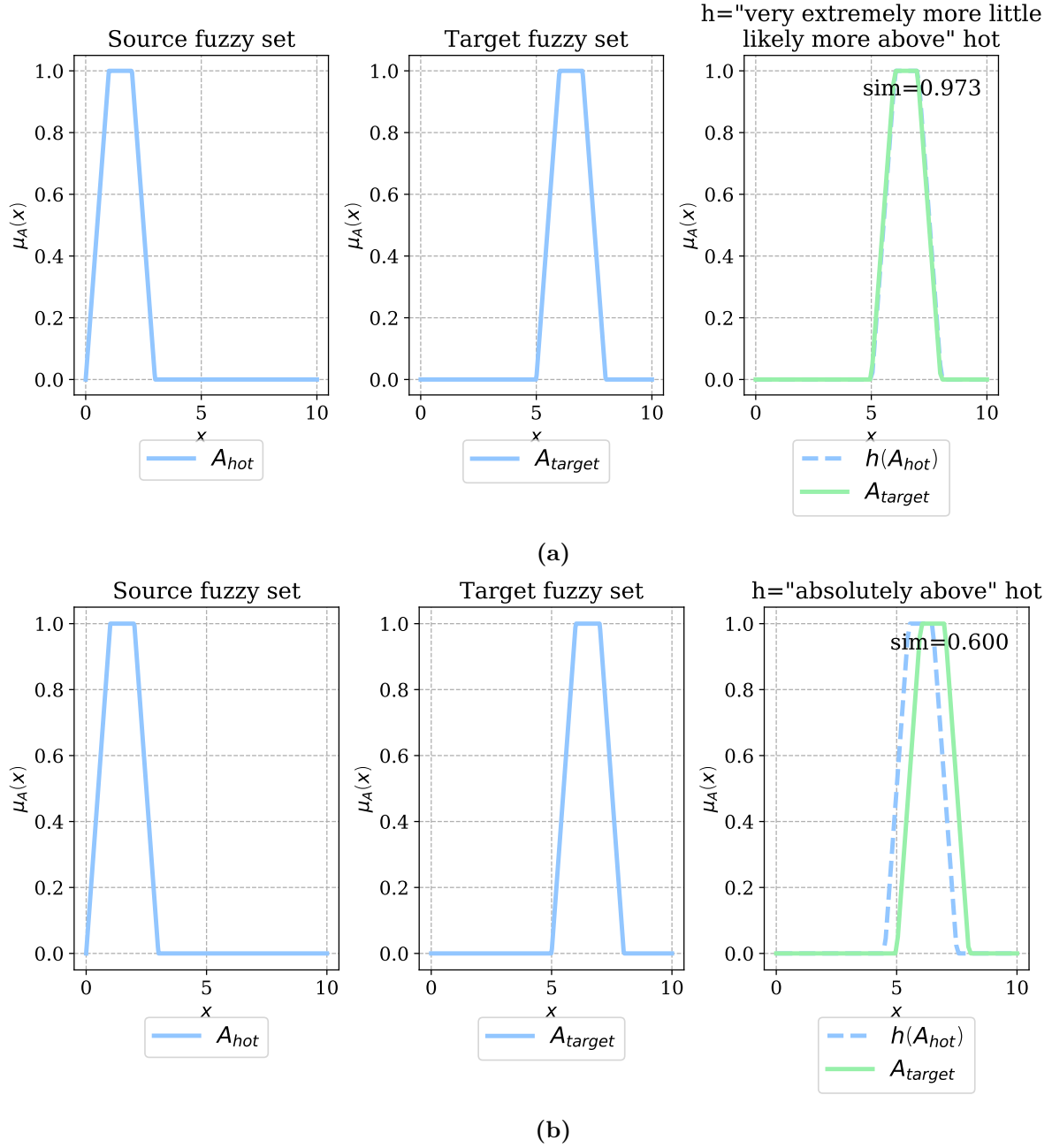


Figure 4.1: Trapezoidal membership function in source and target fuzzy set either a) and b). In a) the value of β is 0.05, which prioritizes accuracy over interpretability; and, in b) the value of β is 0.99, which equilibrates accuracy and interpretability.

functions. This experiment aims to analyze how well can be approximated some membership function to another of a different kind using the proposed methodology.

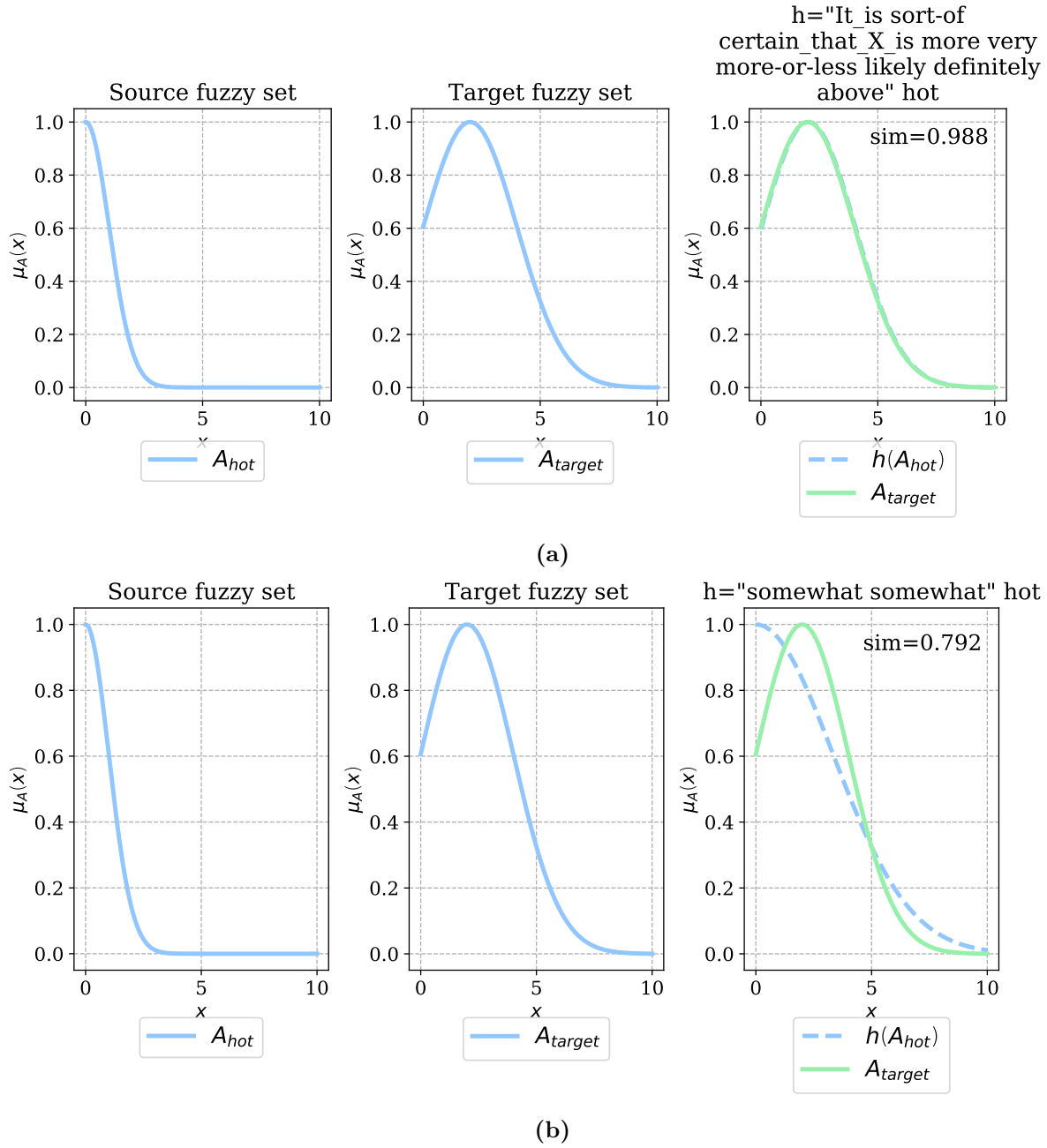


Figure 4.2: Gaussian membership function in source and target fuzzy set, where the target is shifted to the right respect to source in either a) and b). In a) the value of β is 0.05, which prioritizes accuracy over interpretability; and, in b) the value of β is 0.99, which equilibrates accuracy and interpretability.

The specific membership function parameters for each source and target, as well as the similarity obtained after the optimization process, are shown in table 4.5. The parameters

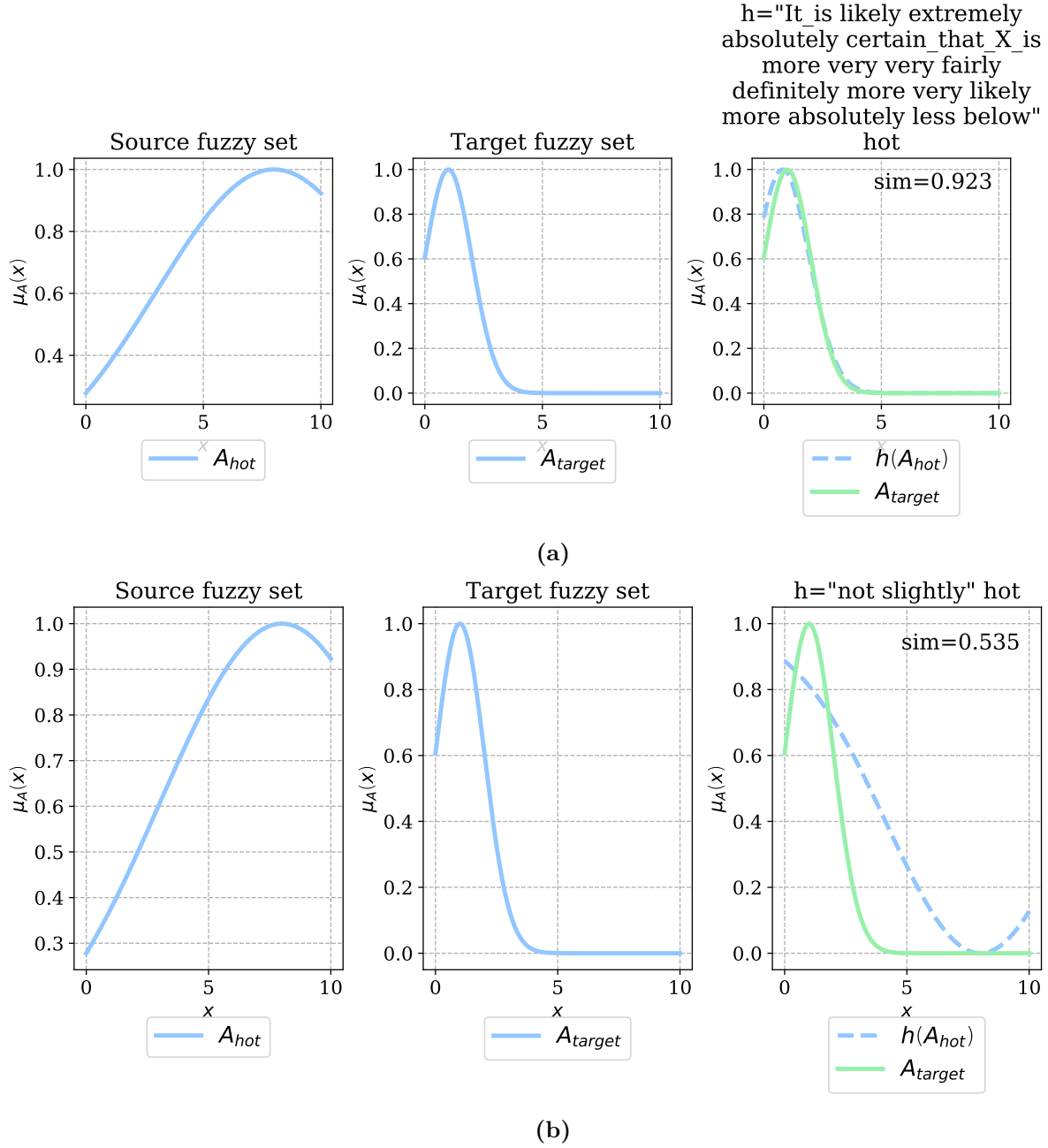
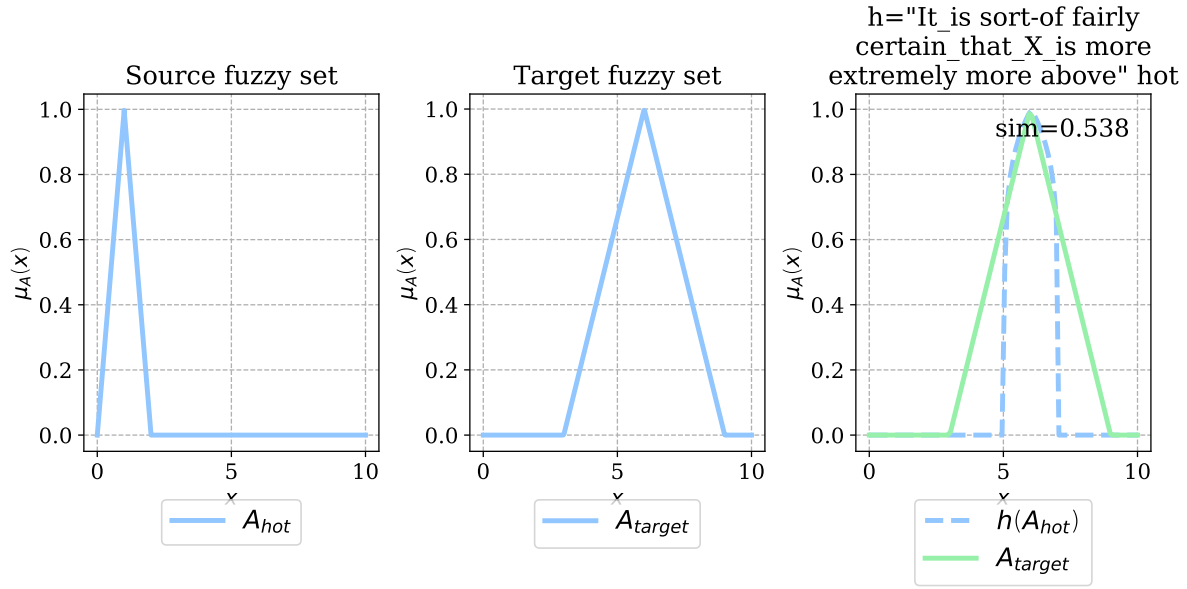
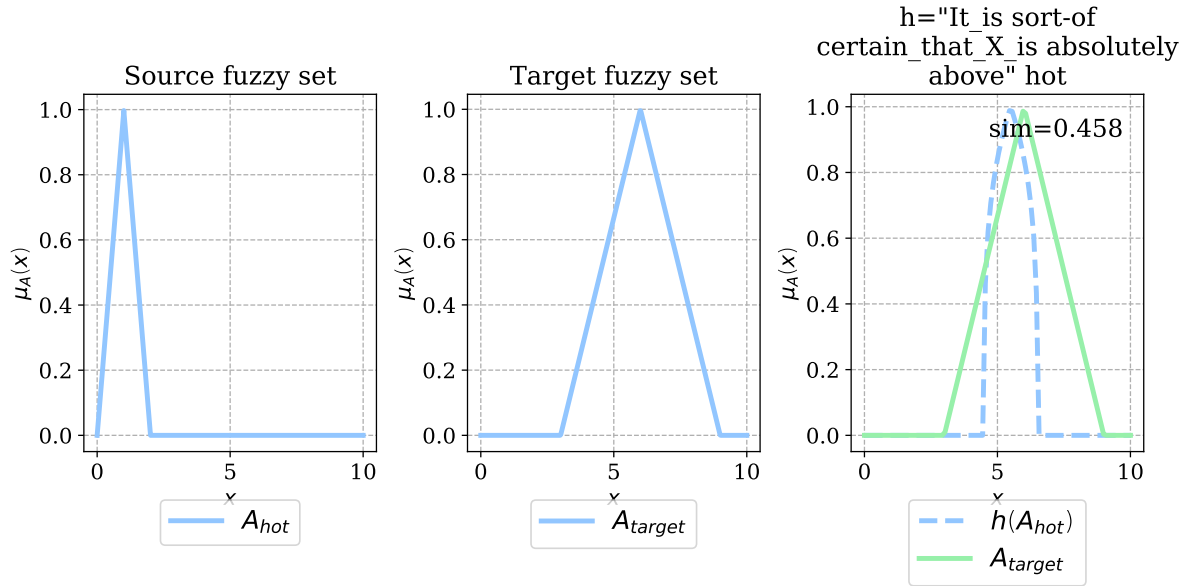


Figure 4.3: Gaussian membership function in source and target fuzzy set, where the target is shifted to the left with respect to source in both a) and b). In a) the value of β is 0.05, which prioritizes accuracy over interpretability; and, in b) the value of β is 0.99, which equilibrates accuracy and interpretability.

order in the table correspond to the design parameters described in section 3.4. Trapezoidal $[a, b, c, d]$, Triangular $[a, b, c]$, and Gaussian $[\sigma, m]$.



(a)



(b)

Figure 4.4: Gaussian membership function in source and target fuzzy set, where the target is shifted to the left with respect to source in both a) and b). In a) the value of β is 0.05, which prioritizes accuracy over interpretability; and, in b) the value of β is 0.99, which equilibrates accuracy and interpretability.

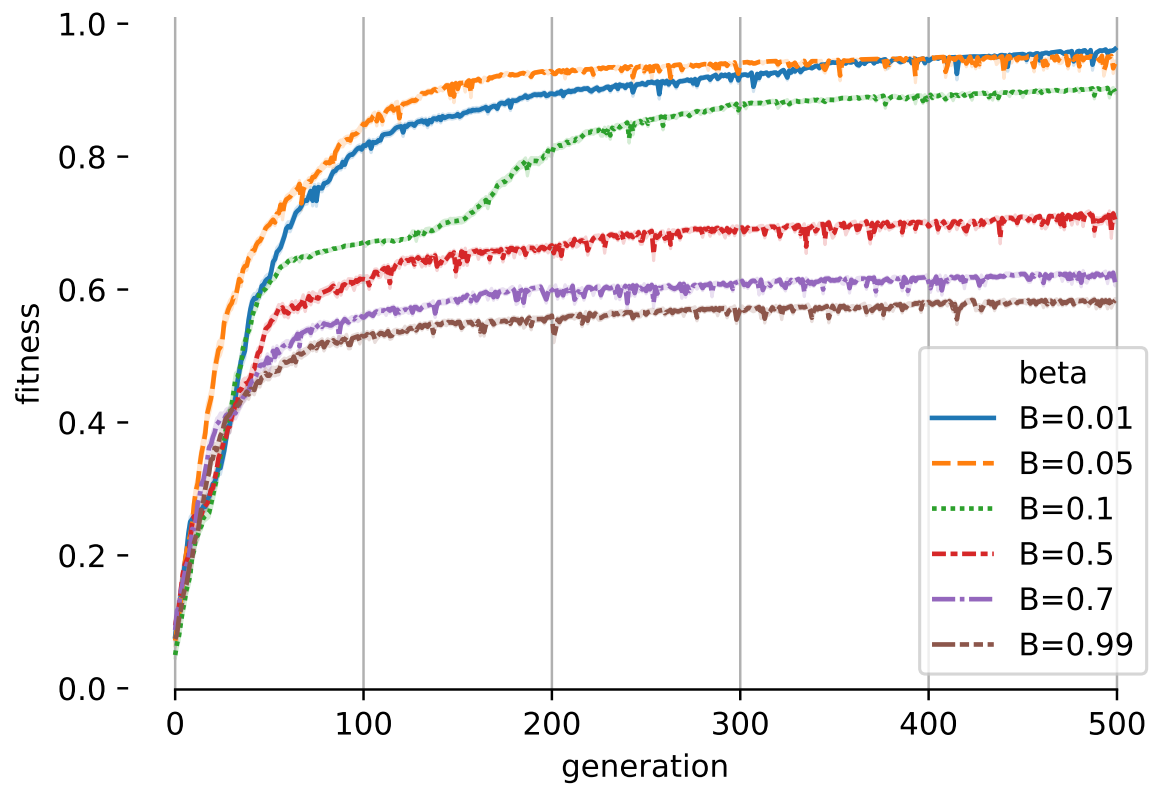


Figure 4.5: The overall evolution of fitness value over generations with different β values in the eight proposed experiments related to same membership function type.

Table 4.5: Case study results where the source and target membership functions are of a different type, and β with value 0.01

source MF	target MF	source MF parameters	target MF parameters	similarity	hedge score
Trapezoidal MF	Gaussian MF	[0, 1, 3, 4]	[2, 2]	0.761	1.000
Trapezoidal MF	Gaussian MF	[7, 8, 9, 10]	[2, 2]	0.594	0.333
Trapezoidal MF	Triangular MF	[7, 8, 9, 10]	[3, 6, 9]	0.696	0.286
Gaussian MF	Trapezoidal MF	[1, 0]	[7, 8, 9, 10]	0.862	0.125
Gaussian MF	Triangular MF	[1, 1]	[3, 6, 9]	0.899	0.250
Gaussian MF	Triangular MF	[3, 3]	[3, 6, 9]	0.900	0.333
Triangular MF	Trapezoidal MF	[0, 1, 2]	[7, 8, 9, 10]	0.816	0.019
Triangular MF	Gaussian MF	[3, 8, 9]	[1, 1]	0.847	0.111
Mean				0.797	0.307
σ				0.107	0.302
σ^2				0.012	0.091

Table 4.6: Case study results where the source and target membership functions are of a different type, and β with value 0.99

source MF	target MF	source MF parameters	target MF parameters	similarity	hedge score
Trapezoidal MF	Gaussian MF	[0, 1, 3, 4]	[2, 2]	0.761	1.000
Trapezoidal MF	Gaussian MF	[7, 8, 9, 10]	[2, 2]	0.568	1.000
Trapezoidal MF	Triangular MF	[7, 8, 9, 10]	[3, 6, 9]	0.599	0.667
Gaussian MF	Trapezoidal MF	[1, 0]	[7, 8, 9, 10]	0.621	0.500
Gaussian MF	Triangular MF	[1, 1]	[3, 6, 9]	0.688	1.000
Gaussian MF	Triangular MF	[3, 3]	[3, 6, 9]	0.727	0.667
Triangular MF	Trapezoidal MF	[0, 1, 2]	[7, 8, 9, 10]	0.767	0.500
Triangular MF	Gaussian MF	[3, 8, 9]	[1, 1]	0.810	0.500
Mean				0.693	0.729
σ				0.088	0.235
σ^2				0.008	0.055

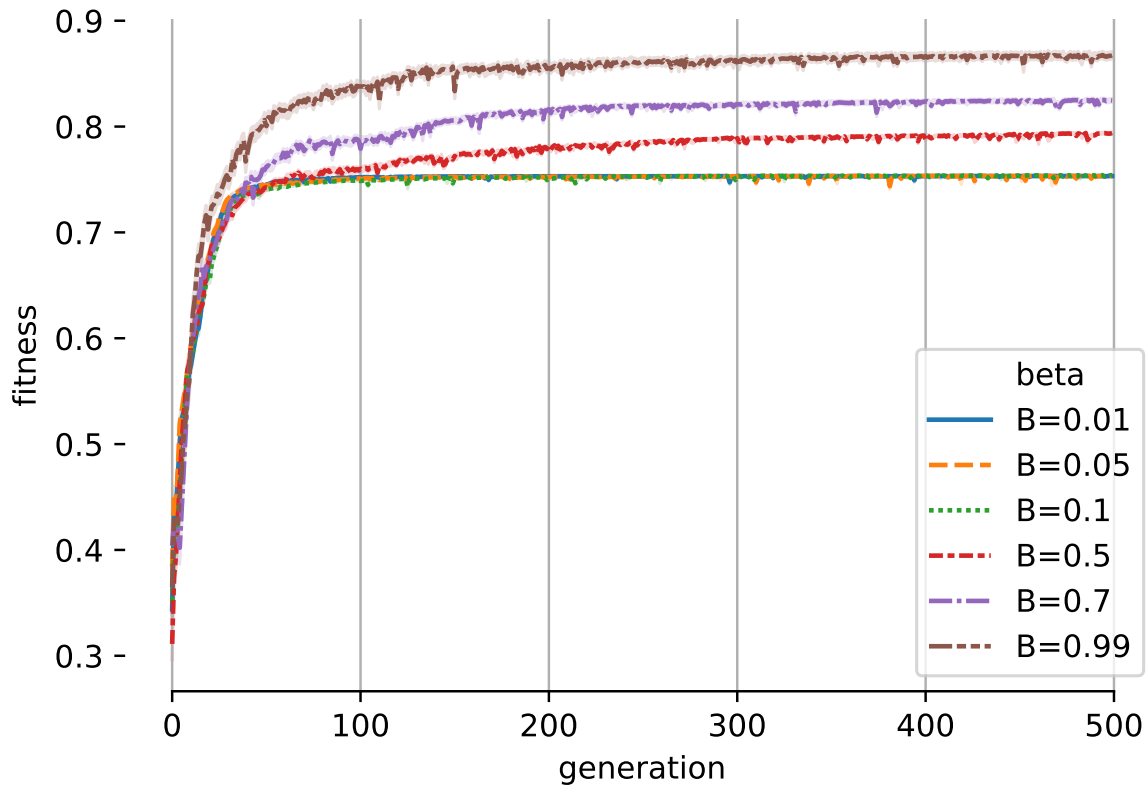


Figure 4.6: The overall evolution of fitness value over generations with different β values in the eight proposed experiments related to different membership function type.

4.1.3 Sensitivity Analysis

This section provides a sensitivity analysis of the main parameter in this proposal, β , which is the threshold value to give more importance to the interpretability (hedge function 3.7) or the accuracy (similarity function 3.6). The positive real factor value of β assigns this importance ratio value to the fuzzy similarity.

Diverse types of applications might have different requirements concerning interpretability. Some critical decision-making systems should need to explain their outcome in a form that the user can easily understand. Conversely, in non-critical applications, interpretability can be secondary and focus on accuracy. The following analysis is to have an intuition of how

the system would respond given a specific situation.

The sensitivity analysis is performed individually in each scenario following the previous study cases. In the first case, the source and target fuzzy set belong to the same type of membership functions; on the contrary, in the second case, they are not. The arbitrary hyper-parameters such as population, selection percentage, mutation percentage, and max generations remain the same as in the experimentation setup; table 4.5 shows the values for these parameters.

Same type of membership functions

The source and target fuzzy set are of the same kind of membership functions in this setup. The only change is in their design parameters. The convex membership functions used in this experiment are Trapezoidal, Gaussian, and Triangular. Table 4.7 shows the model's behavior given different β values.

Table 4.7: Performances of the model, approximating membership functions of the same type, in terms of similarity, hedge score, and fitness; with respect to the following values of β : 0.01, 0.05, 0.1, 0.5, 0.7, 0.99.

β		0.01	0.05	0.10	0.50	0.70	0.99
similarity	mean	0.885	0.858	0.838	0.789	0.755	0.712
	σ	0.161	0.164	0.198	0.148	0.173	0.161
hedge score	mean	0.150	0.297	0.382	0.746	0.871	0.946
	σ	0.104	0.157	0.171	0.550	0.527	0.492
fitness	mean	0.884	0.852	0.820	0.731	0.730	0.762
	σ	0.160	0.164	0.188	0.149	0.171	0.209

Figure 4.7 shows that as the value of β increases, the hedge score (interpretability of hedge chain) also does. The opposite occurs to the similarity score; it decreases. Depending on the

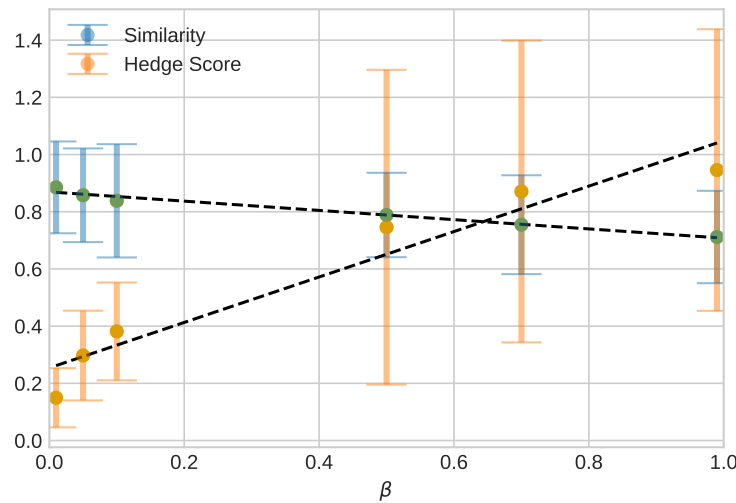


Figure 4.7: System performance changes in terms of similarity and hedge score with respect to the changes in β when the source and target membership functions are of the same type.

necessity of the domain application, the domain expert should fix this β value to align with the specifications. In the scenario described in the table 4.3 the slope for similarity is -0.16 and for hedge score 0.794 with p-values 2×10^{-4} and 1×10^{-3} respectively. The changes in β affect interpretability (hedge score) more. It might be a desirable option to sacrifice a small portion of accuracy in order to get a more understandable representation of the linguistic modifiers.

Different type of membership functions

In this setup, contrary to the previous one, the source and target fuzzy set belong to different types of memberships functions. Although they are the same type of functions, the objective is to approximate a membership function of a kind to another of a different kind. Table 4.8 shows the model's behavior given different β values.

Table 4.8: Performances of the model, approximating membership functions of different type, in terms of similarity, hedge score, and fitness; with respect to the following values of β : 0.01, 0.05, 0.1, 0.5, 0.7, 0.99.

β		0.01	0.05	0.10	0.50	0.70	0.99
similarity	mean	0.797	0.793	0.791	0.702	0.713	0.693
	σ	0.107	0.107	0.106	0.132	0.113	0.088
hedge score	mean	0.307	0.504	0.565	0.821	0.833	0.854
	σ	0.302	0.609	0.584	0.530	0.519	0.507
fitness	mean	0.796	0.790	0.783	0.685	0.702	0.718
	σ	0.107	0.105	0.101	0.109	0.117	0.173

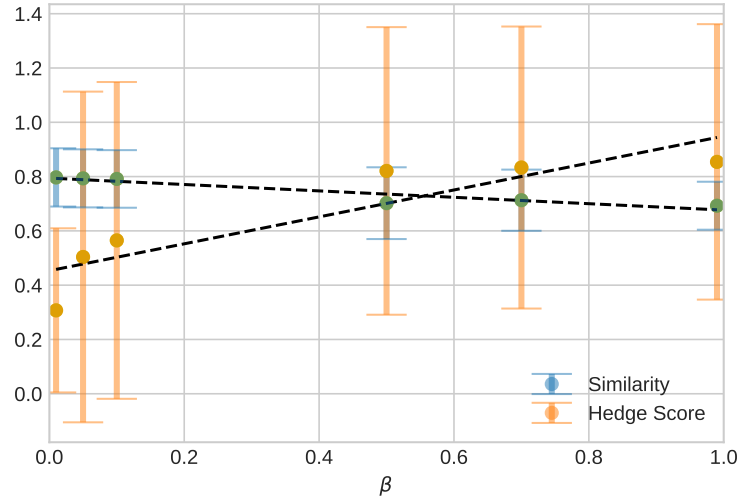


Figure 4.8: System performance changes in terms of similarity and hedge score with respect to the changes in β when the source and target membership functions are of different kind.

Figure 4.8 supports the parameter β has a positive correlation to the hedge score and a negative one with the similarity. In the scenario described in the table 4.5, the slope for similarity is -0.117 and for hedge score 0.495 with p-values 5×10^{-3} and 0.01 respectively. As in the first case, the changes in β affect interpretability more. In contrast, it would be harder to improve the interpretability in this scenario without sacrificing the similarity score.

This result could be expected due to the complexity to approximate memberships functions of a different kind.

4.2 Semantic enhancement of fuzzy variables using binary hedge transformations

This experiment shows a fully automated way to build the initial knowledge base, optimize design parameters in a neuro-fuzzy architecture, and the interpretable layer to explain new fuzzy sets.

In the proposed experimental setup, two different steps are considered. The first step is a data-driven neuro-fuzzy design and training process. The second step is to generate the interpretable interface that creates a linguistical explanation of the fitted neuro-fuzzy parameters. There are 16 datasets for classification tasks to measure how well the models behave. The general description of each one of the datasets is presented in the table 4.10.

For each dataset, the proposed methodology performed the process as described in section 3.4. Also, different values are tested to analyze the method behavior; 3, 5, and 7 fuzzy partitions per input value are selected; and, from 5 to 50 fuzzy rules in increments of 5. In the experiment is performed, for every problem, 30 different cases with 5-fold cross-validation. In table 4.9 is shown the complete hyper-parameters values.

Table 4.9: Selected hyper-parameters values in the optimization of automatically building fuzzy inference system, using Mamdani-type Neuro-fuzzy representation.

Hyper parameter	Value(s)
Input partition	3,5,7
Learning rate	0.01
Batch size	Full batch
Epoch	3,000
Goal error	1e-6
K-Fold cross-validation	5 Folds
T-norm	Product
S-norm	Sum

4.2.1 Generation of linguistic modifiers to explain model adjustment parameters

The parameters used to find the best hedge chain candidates through Grammar-Guide Genetic Algorithm are 100 individuals for initial population, elitist selection of 20%, mutation method of one node modification, mutation percentage of 20%, 50 maximum generations, top best $k = 5$ individuals as a result based in similarity function.

The hedges that concern to uncertainty modifier, considered for this experiment, for concentration, are: “*very*” with $p = 2$, “*extremely*” with $p = 4$, “*exactly*” with $p = 6$; those who refer to dilation modifiers: “*more – or – less*” with $p = 0.5$, “*kind – of*” with $p = 0.3$. These hedge modifier values are set arbitrarily, but their values must reflect their semantic meaning in the operation of the membership functions. The linguistic terms that evoke an increment in uncertainty should have a $p < 1$ value for a dilation operation; otherwise, if the semantic meaning refers to a decrement in uncertainty, it should have a $p > 1$, for a

concentration operation.

The linguistic modifiers which refer to shift operation of the membership function considered in this experiment are: “*Upper – than*” that displace to the right the support of membership function at α -cut = 0.5 of 25%; On the other hand, the linguistic modifier “*Lower – than*” displace to left the support of membership function at α -cut = 0.5 of 25%. Given a fuzzy set’s membership function $\mu_A(x)$, if a “*Upper – than*” operation is applied, its core change to: $core(\mu_A(x))^{new} = core(\mu_A(x))^{old} - |support(\mu_{A\alpha=0.5}(x))| \times 0.25$.

The overall obtained result of Mamdani neuro-fuzzy architecture evaluation, using K-Fold cross-validation with $k = 5$ for every dataset are shown in the table 4.10, where the mean value of the 16 datasets is 0.814 with a standard deviation of 0.027. The worst performance was obtained for the dataset *movement_libras* with 90 attributes and 15 target classes. The best performances were achieved in the dataset *segment0* with 0.998 with a standard deviation of 0.001. The first quartile is 0.73 and the third 0.968 of f1-score values.

Table 4.10: Result f1-score and standard deviations of 5 k-fold cross-validation of automatically built neuro-fuzzy model for each of 16 datasets.

Dataset	Instances	Attributes	Classes	f1 score	σ
iris	150	4	3	0.967	0.052
bupa	345	6	2	0.700	0.026
ecoli	336	7	8	0.755	0.038
yeast1	1484	8	2	0.742	0.020
page-blocks	5472	10	5	0.830	0.066
flare	1066	11	6	0.661	0.027
cleveland	297	13	5	0.636	0.043
wine	178	13	3	0.976	0.021
penbased	10992	16	10	0.837	0.039
segment0	2308	18	2	0.998	0.001
twonorm	7400	20	2	0.971	0.002
wdbc	569	30	2	0.971	0.008
satimage	6435	36	6	0.747	0.004
spambase	4597	57	2	0.898	0.010
sonar	208	60	2	0.879	0.041
movement_libras	360	90	15	0.450	0.040
mean				0.814	0.027

The image 4.9 shows the overall performance in terms of f1-score. Their values are grouped by the three different experimental setups in which the number of fuzzy sets belonging to each fuzzy variable (fuzzified input feature domain) is set to 3, 5, and 7. Furthermore, we compare their distribution by considering the number of rules in the neuro-fuzzy model. In most cases, as fewer rules are considered, the better is the model performance. As the number

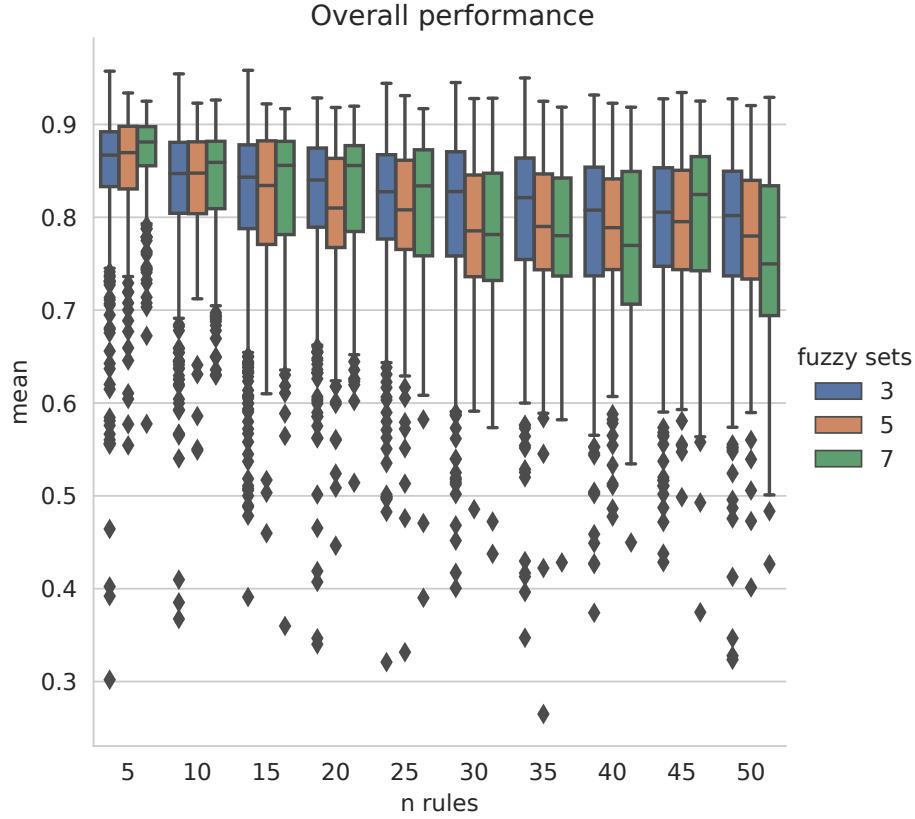


Figure 4.9: Overall mean performance of Mamdani Neuro-fuzzy model for each number of the rules, grouped by their three input clusters configurations.

of rules increases, fewer fuzzy sets achieve slightly better performance.

The overall result of hedge approximation to the optimized FIS's knowledge base is shown in table 4.11 and figure 4.10. The mean of the similarity score (listed in 3.6) are: for three fuzzy sets per fuzzy variable (feature) is 0.911, with a standard deviation of 0.057; in the case of five fuzzy sets per fuzzy variable is 0.931 with 0.035 of standard deviation; in the last setup, with seven fuzzy sets per fuzzy variable is 0.948 with a standard deviation of 0.028. In most cases, with three partitions per input, it tends to be too difficult for the hedge chain searching approach to approximate the model. As the number of fuzzy sets increases in the searching space, the better the fitness score gets. The difference between those is 3.9%.

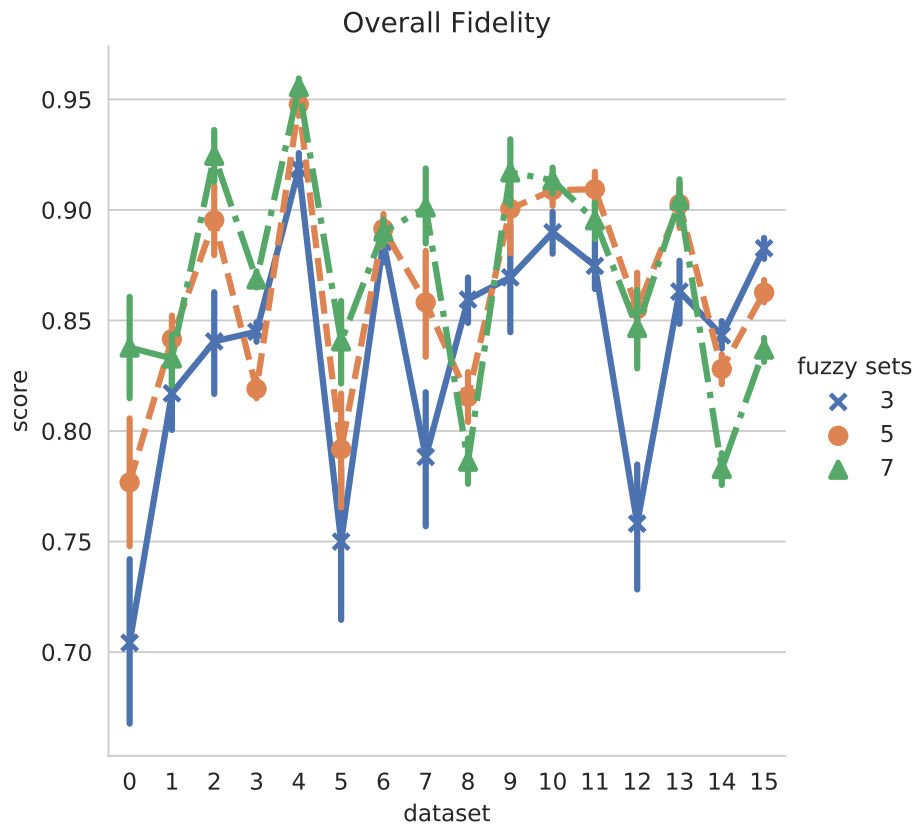


Figure 4.10: Overall fidelity score (equation 3.6) grouped by each fuzzy variable configuration: 3, 5 and, 7 fuzzy sets each one.

Table 4.11: Overall result of approximate model similarity between the optimized membership functions and linguistically transformed original knowledge. The similarity measure is shown in equation 3.6.

Dataset	Number of fuzzy sets			Mean by dataset	σ by dataset
	3.000	5.000	7.000		
iris	0.761	0.892	0.937	0.863	0.092
cleveland	0.907	0.905	0.914	0.909	0.004
flare	0.923	0.940	0.954	0.939	0.016
movement_libras	0.926	0.937	0.964	0.942	0.020
spambase	0.939	0.962	0.967	0.956	0.015
page-blocks	0.818	0.903	0.945	0.888	0.065
wdbc	0.938	0.937	0.942	0.939	0.002
bupa	0.866	0.943	0.970	0.926	0.054
penbased	0.919	0.865	0.885	0.890	0.027
yeast1	1.000	0.918	0.962	0.960	0.041
wine	0.923	0.955	0.969	0.949	0.024
twonorm	0.979	0.971	0.950	0.967	0.015
ecoli	0.893	0.938	0.935	0.922	0.025
segment0	0.922	0.947	0.966	0.945	0.022
satimage	0.926	0.881	0.914	0.907	0.023
sonar	0.930	1.000	1.000	0.977	0.040
Mean	0.911	0.931	0.948		
σ	0.057	0.035	0.028		

Chapter 5

Conclusion

This research proposes a linguistic granule model representing generic entities affected by a linguistic description restricted by a context-free grammar. These abstract elements interact in an environment, and their aptitude or performance can be measured given a particular metric. This linguistic granule model is optimized using Grammar-Guided Genetic Programming and Granular Computing.

This work aims to solve an open problem in the fuzzy logic domain area, which is the semantic detachment of linguistic description after an optimization process of fuzzy inference systems. The semantic meaning is essential to maintain the model's interpretability and explainability to transfer knowledge to the expert or user using natural language.

Obtained results show that neuro-fuzzy systems could play an essential role in interpretable machine learning, providing natural language explanations from a well-defined knowledge keeping its semantic meaning after the optimization process by finding a well-formed linguistic granule.

In order to validate the proposal were conducted two independent experiments; the first one (described in section 4.1) is a synthetic setup in which initial fuzzy sets should

approximate as much as possible to a target fuzzy set design. These transformations should be performed only by linguistic modifiers. The second experiment, described in section 4.2, is a complete setup machine learning flow for classification problems. A collection of 16 publicly available datasets were used to measure the performance of the Mamdani-type fuzzy inference system with the interpretability enhancement posthoc method.

The following sections present the particular conclusions for each of the experimentation scenarios. In section 5.1 is presented the conclusion of the unary hedge transformation function design as a multi-objective optimization problem in which the linguistic granule should maximize the specificity and coverage criteria. The particular conclusion to the restricted grammar to build intrinsic interpretable linguistic modifiers and binary hedge transformation function tacking into account the context given by fuzzy variables design is presented in section 5.2.

5.1 Unary hedge transformations over fuzzy sets

In this experimental setup, we conducted two different experiments. The first one refers to the approximation of some membership function to another one of the same kind. And, in the second one, the approximation of some membership function to another one of a different kind.

In the first case study, we considered eight different membership functions. The membership function types are *Trapezoidal*, *Triangular*, and *Gaussian*. The parameters of the membership functions are shown in table 4.4. The mean of similarity obtained, with $\beta = 0.99$ is 0.712 with a standard deviation of 0.161, the minimum values are 0.455 and 0.535 that belong to *Triangular* and *Gaussian* type membership functions respectively. The proposed “hedge score” is 0.821 with a standard deviation of 0.258.

In the same study case, but with the parameter $\beta=0.01$, the results are shown in table 4.3. The mean similarity value obtained, with $\beta = 0.01$, is 0.885 with a standard deviation of 0.161. The proposed “hedge score” is 0.150 with a standard deviation of 0.104.

The overall results and sensitivity analysis of the first study case show that it is possible to approximate membership functions one to another of the same kind (but different parameter values) using only linguistic modifiers. The proposed methodology using GGGP is finding the best hedge chain restricted by a grammar that might promote the expert’s straightforward explanation of the resulting fuzzy set. Accordingly to the sensitivity analysis, the changes in β might increase the interpretability without substantially decreasing the accuracy, which is convenient in critical scenarios where interpretability plays an important role.

In the context of the second case study, the membership function of the source and target are of different kinds (with different parameter values) and are considered the same as in the first case study (triangular, trapezoidal, and Gaussian). The parameters of each membership function, their similarity and hedge score values after the optimization are shown in table 4.5, considering the parameter $\beta = 0.01$. Considering the similarity value, the mean of those results is 0.797 and the standard deviation of 0.107. However, concerning the hedge score (“interpretability” value) are 0.307 and 0.302 the mean and standard deviation respectively.

In the same experiment, only varying the value β to 0.99, the results concerning the similarity (accuracy) are 0.693 and 0.088 the mean and standard deviation. The results for the hedge score (“interpretability” score) are 0.729 and 0.235, the mean and standard deviation, respectively. The sensitivity analysis supports that the changes in β affect interpretability more than the similarity. However, it does not achieve the same trade-off ratio as the first study case but still can gain more interpretability and lose less performance.

The results in cases where the source is a *Gaussian* membership function and is fitted to represent either *Trapezoidal and Triangular* membership functions are better (in terms of

accuracy) than the rest of the cases, considering the parameter $\beta = 0.01$, which prioritizes the accuracy. The mean grouped by the source are: *Gaussian*, $mean = 0.887, \sigma = 0.012$; *Triangular*, $mean = 0.831, \sigma = 0.015$; and, *Trapezoidal*, $mean = 0.683, \sigma = 0.048$.

These sets of results show that the *Gaussian* membership function achieves a better adjustment than the rest of the membership functions. This outcome is an interesting finding because if the similarity is good enough between the source and the target, then it can generalize a knowledge base only by *Gaussian* membership functions and reduce its complexity. For example, the consequent computation can be analytical in fuzzy inference systems. Exploring these cases further could be interesting, and using this assumption as a baseline to optimize analytical calculation in the interval and generalized type-2 fuzzy systems.

An important fact to highlight is that this work relies on the proposed linguistic modifier's parameter values in the optimization process. The results can be improved by the variations of the values presented in table 4.2. Moreover, the grammar might achieve a wide variety of results with slight changes.

5.2 Binary hedge transformations over fuzzy variables

The presented setup describes a methodology to generate an explanation layer to optimize Mamdani-type neuro-fuzzy models and use them on broad application domains. After the optimization process, the models built upon a rule-based system suffer from a lack of interpretability, turning them into black-boxes. The proposed methodology aims to turn back the neuro-fuzzy model to a white-box model, giving back the interpretable meaning to the fuzzy sets through a defined context-free grammar that generates hedge chains to modify the fuzzy sets' membership functions.

After optimization, the generated hedge chains can build unary and binary expressions to approximate the initial knowledge (fuzzy sets) to the resulting fuzzy sets. Those linguistic modifiers are optimized by a Genetic Algorithm that works over the context-free grammar derivation trees as individuals, which a similarity function measures its fitness aptitude (equation 3.6).

The proposed methodology was evaluated over a collection of 16 datasets for classification with different characteristics (shown in table 4.10). For each dataset, an initial domain fuzzification with 3, 5, and 7 fuzzy sets per input was built, where well-distributed Gaussian functions designed their membership functions. The number of fuzzy rules is fixed from 5 to 50 with an increment of 5 rules each step. In the setup to train the neuro-fuzzy model, cross-validation K-fold with $k = 5$ was used; with the following stopping conditions: 3,000 maximum epochs, error tolerance of $1e^{-6}$, early stopping when 15 consecutive performance decay epochs.

The f1-score measures how well the model fitted to the dataset and the similarity score to measure the approximation of the interpretable fuzzy inference system and the optimized one.

In terms of performance, the neuro-fuzzy model resulted in 0.814 of mean, and 0.026 in the standard deviation on the best configuration models, which are related to the number of fuzzy sets per feature, and the number of rules (showed in figure 4.9). The maximum f1-score is 0.998, and the minimum 0.45, which is the only value below 0.60. It is relevant to note that no feature selection or feature extraction process was carried out to get a baseline. Results show that the overall performance increases as a lower number of rules are used, in addition to obtaining a better performance when a higher number of fuzzy sets per input feature is used (figure 4.9). This is an important insight because a higher model complexity becomes less interpretable.

The similarity between the optimized and interpretable systems resulted in 0.93 of mean and a standard deviation of 0.018, supporting the hypothesis that it is feasible to build interpretable layers of abstraction after an optimization process. When the similarity score is high, the solution hedge chain can be used to replace the missing-label optimized fuzzy set; or, the transformation of the fuzzy set through the hedge chain (identity, unary and binary) can be performed over the initial knowledge base. The overall model fidelity is shown in figure 4.10, the similarity score increases as the number of fuzzy sets increase too. A relatively small number of fuzzy sets should be selected to achieve better interpretability. Based on obtained results, it is better to select a few fuzzy rules with many fuzzy sets per input feature for better performance and high fidelity.

Chapter 6

Future work

There are open study cases and improvement opportunities that will be covered in future works, such as:

- Extend the application domain of the proposed model to other specific Machine Learning processes, such as feature engineering and structure discovering. Those steps have special relevance in the definition and discovery of interpretable models.
- Modeling uncertainty by higher Linguistic Granule order definition. The generic model definition allows the incorporation of different model representations, which could be used in probabilistic sets, rough sets, possibility sets, or any model that improves model understanding through linguistic descriptions.
- Usage of higher arity of hedge chain relationships with more complex behaviors. The usage of second-order relationships allowed to reduce the multi-optimization to a single objective showing significant results over 90% of similarity between the optimized and linguistic-only modified model.
- The extension of linguistic modifiers in the context-free grammar would increase the

domain solution space; therefore, a better pseudo-optimal solution could be found.

- The proposed methodology can be used for building hedge chain descriptions to explain the uncertainty interval type-2 fuzzy inference systems. In the binary transformation function, the certainty-related description might describe the actual uncertainty footprint.
- The exploration process proposed in this work could be improved by another method that could incorporate more information about the nodes to explore and not just the distance. This specific step could be defined in a Reinforcement Learning context, in which the generator is an agent and exploration path the actions.
- The overall performance and interpretability can be improved and build a better robust initial knowledge base construction, applying feature selection, rule selection or, adaptive number of fuzzy sets per feature.

Bibliography

- Adhyaru, D. M., Patel, J., and Gianchandani, R. (2010). Adaptive neuro-fuzzy inference system based control of robotic manipulators. In *ICMET 2010 - 2010 International Conference on Mechanical and Electrical Technology, Proceedings*, pages 353–358.
- Arras, L., Horn, F., Montavon, G., Müller, K.-R., and Samek, W. (2017). "What is relevant in a text document?": An interpretable machine learning approach. *PLoS ONE*, 12(8).
- Azadeh, A., Gaeini, Z., Motevali Haghighi, S., and Nasirian, B. (2016). A unique adaptive neuro fuzzy inference system for optimum decision making process in a natural gas transmission unit. *Journal of Natural Gas Science and Engineering*, 34:472–485.
- Bargiela, A. and Pedrycz, W. (2003). *Granular Computing*.
- Bastani, O., Kim, C., and Bastani, H. (2017). Interpretability via Model Extraction.
- Basu, S., Kumbier, K., Brown, J. B., and Yu, B. (2018). Iterative random forests to discover predictive and stable high-order interactions. *Proceedings of the National Academy of Sciences of the United States of America*, 115(8):1943–1948.
- Beaton, B. (2018). Crucial Answers about Humanoid Capital. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 5–12. IEEE Computer Society.

- Belle, V. (2017). Logic meets probability: Towards explainable AI systems for uncertain worlds. *IJCAI International Joint Conference on Artificial Intelligence*, pages 5116–5120.
- Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly, Beijing.
- Brinkrolf, J. and Hammer, B. (2018). Interpretable machine learning with reject option. *At-Automatisierungstechnik*, 66(4):283–290.
- Caywood, M. S., Roberts, D. M., Colombe, J. B., Greenwald, H. S., and Weiland, M. Z. (2017). Gaussian process regression for predictive but interpretable machine learning models: An example of predicting mental workload across tasks. *Frontiers in Human Neuroscience*, 10.
- Chan, V. K. and Chan, C. W. (2020). Towards explicit representation of an artificial neural network model: Comparison of two artificial neural network rule extraction approaches. *Petroleum*, 6(4):329–339.
- Cheng, M.-Y., Tsai, H.-C., Ko, C.-H., and Chang, W.-T. (2008). Evolutionary fuzzy neural inference system for decision making in geotechnical engineering. *Journal of Computing in Civil Engineering*, 22(4):272–280.
- De Medeiros, I. B., Soares Machado, M. A., Damasceno, W. J., Caldeira, A. M., Dos Santos, R. C., and Da Silva Filho, J. B. (2017). A Fuzzy Inference System to Support Medical Diagnosis in Real Time. In Sh Y. Ahuja V., D. D. S. Y. B. D. T. Y. T. J. M. A. N., editor, *Procedia Computer Science*, volume 122, pages 167–173. Elsevier B.V.
- Deshpande, S. U. and Bhosale, S. S. (2013). Adaptive neuro-fuzzy inference system based robotic navigation. In *2013 IEEE International Conference on Computational Intelligence and Computing Research, IEEE ICCIC 2013*. IEEE Computer Society.

- Ding, S., Jia, H., Chen, J., and Jin, F. (2014). Granular neural networks. *Artificial Intelligence Review*, 41(3):373–384.
- Dubois, D. and Prade, H. (1992). Putting Rough Sets and Fuzzy Sets Together. In *Intelligent Decision Support*.
- Gacto, M. J., Alcalá, R., and Herrera, F. (2011). Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. *Information Sciences*, 181(20):4340–4360.
- Gayathri, B. M. and Sumathi, C. P. (2016). Mamdani fuzzy inference system for breast cancer risk detection. In Karthikeyan M., K. N., editor, *2015 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2015*. Institute of Electrical and Electronics Engineers Inc.
- Goodman, B. and Flaxman, S. (2016). European Union regulations on algorithmic decision-making and a "right to explanation". pages 1–9.
- Guo, H. and Wang, W. (2019). Granular support vector machine: a review. *Artificial Intelligence Review*, 51(1):19–32.
- Haspiel, J., Du, N., Meyerson, J., Robert, L. P., Tilbury, D. M., Yang, X. J., and Pradhan, A. K. (2018). Explanations and Expectations: Trust Building in Automated Vehicles. *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*.
- Hofmann, D., Schleif, F.-M., Paaßen, B., and Hammer, B. (2014). Learning interpretable kernelized prototype-based models. *Neurocomputing*, 141:84–96.
- Honka, A. M., Van Gils, M. J., and Pärkkä, J. (2011). A personalized approach for predicting the effect of aerobic exercise on blood pressure using a Fuzzy Inference System. In

- Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pages 8299–8302.
- Huang, S. H., Held, D., Abbeel, P., and Dragan, A. D. (2018). Enabling robots to communicate their objectives. *Autonomous Robots*, pages 1–18.
- Lakkaraju, H., Bach, S. H., and Leskovec, J. (2016). Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 13-17-Aug, pages 1675–1684. Association for Computing Machinery.
- Li, X., Liu, H., Yang, J., Xie, G., Xu, M., and Yang, Y. (2017). Using machine learning models to predict in-hospital mortality for st-elevation myocardial infarction patients. *Studies in Health Technology and Informatics*, 245:476–480.
- Loia, V. and Tomasiello, S. (2017). Granularity into functional networks. *2017 3rd IEEE International Conference on Cybernetics, CYBCONF 2017 - Proceedings*.
- Manrique, D., Rios, J., and Rodríguez-Patón, A. (2009). Grammar-Guided Genetic Programming. In *Encyclopedia of Artificial Intelligence*.
- Mencar, C. and Fanelli, A. M. (2008). Interpretability constraints for fuzzy information granulation. *Information Sciences*, 178(24):4585–4618.
- Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97.
- Miller, T. (2017). Explanation in Artificial Intelligence: Insights from the Social Sciences. *CoRR*, abs/1706.0.
- Molnar, C. (2019). *Interpretable Machine Learning*.
<https://christophm.github.io/interpretable-ml-book/>.

- Nápoles, G., Mosquera, C., Falcon, R., Grau, I., Bello, R., and Vanhoof, K. (2018). Fuzzy-Rough Cognitive Networks. *Neural Networks*, 97:19–27.
- Pal, S. K., Ray, S. S., and Ganivada, A. (2010). *Granular Neural Networks, Pattern Recognition and Bioinformatics*.
- Panoutsos, G. and Mahfouf, M. (2010). A neural-fuzzy modelling framework based on granular computing: Concepts and applications. *Fuzzy Sets and Systems*, 161(21):2808–2830.
- Panoutsos, G., Mahfouf, M., Mills, G. H., and Brown, B. H. (2010). A generic framework for enhancing the interpretability of granular computing-based information. In *2010 IEEE International Conference on Intelligent Systems, IS 2010 - Proceedings*, pages 19–24.
- Pappis, C. P. and Karacapilidis, N. I. (1993). A comparative assessment of measures of similarity of fuzzy values. *Fuzzy Sets and Systems*, 56(2):171–174.
- Pedrycz, W. and Chen, S.-M. (2011). *Granular Computing and Intelligent Systems*.
- Pedrycz, W. and Homenda, W. (2013). Building the fundamentals of granular computing: A principle of justifiable granularity. *Applied Soft Computing Journal*, 13(10):4209–4218.
- Pota, M., Esposito, M., and De Pietro, G. (2017). Designing rule-based fuzzy systems for classification in medicine. *Knowledge-Based Systems*, 124:105–132.
- Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier.
- Shalaeva, V., Alkhoury, S., Marinescu, J., Amblard, C., and Bisson, G. (2018). Multi-operator decision trees for explainable time-series classification. *Communications in Computer and Information Science*, 853:86–99.

- Smith, A. and Nolan, J. J. (2018). The problem of explanations without user feedback. *CEUR Workshop Proceedings*, 2068.
- Tsakonas, A., Ampazis, N., and Dounias, G. (2006). Towards a comprehensible and accurate credit management model: Application of four computational intelligence methodologies. In *Proceedings of the 2006 International Symposium on Evolving Fuzzy Systems, EFS'06*, pages 295–299.
- Valdes, G., Luna, J. M., Eaton, E., Simone, C. B., Ungar, L. H., and Solberg, T. D. (2016). MediBoost: A Patient Stratification Tool for Interpretable Decision Making in the Era of Precision Medicine. *Scientific Reports*, 6.
- van der Waa, J., Van Diggelen, J., Neerincx, M., and Raaijmakers, S. (2018). ICM: An intuitive model independent and accurate certainty measure for machine learning. In van den Herik J., R. A. P., editor, *ICAART 2018 - Proceedings of the 10th International Conference on Agents and Artificial Intelligence*, volume 2, pages 314–321. SciTePress.
- Varshney, K. R. (2016). Engineering Safety in Machine Learning. *arXiv e-prints*, page arXiv:1601.04126.
- Varshney, K. R. (2017). Interpretable machine learning via convex cardinal shape composition. In *54th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2016*, pages 327–330. Institute of Electrical and Electronics Engineers Inc.
- Vasilev, N., Mincheva, Z., and Nikolov, V. (2020). Decision tree extraction using trained neural network. In *SMARTGREENS 2020 - Proceedings of the 9th International Conference on Smart Cities and Green ICT Systems*, pages 194–200.
- Wang, L. X. and Mendel, J. M. (1992). Generating Fuzzy Rules by Learning from Examples. *IEEE Transactions on Systems, Man and Cybernetics*, 22(6):1414–1427.

- Wang, L.-X., Mendel, J. M., and Others (1992). Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE transactions on Neural Networks*, 3(5):807–814.
- Wang, T., Rudin, C., Velez-Doshi, F., Liu, Y., Klampfl, E., and Macneille, P. (2017). Bayesian rule sets for interpretable classification. In Bonchi F. Wu X., B.-Y. R. D.-F. J. Z. Z.-H., editor, *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 1269–1274. Institute of Electrical and Electronics Engineers Inc.
- Williams, J. J., Ang, A., Rafferty, A. N., Lasecki, W. S., Tingley, D., and Kim, J. (2018). Enhancing online problems through instructor-centered tools for randomized experiments. In *Conference on Human Factors in Computing Systems - Proceedings*, volume 2018-April. Association for Computing Machinery.
- Xu, X., Wang, G., Ding, S., Jiang, X., and Zhao, Z. (2015). A new method for constructing granular neural networks based on rule extraction and extreme learning machine. *Pattern Recognition Letters*, 67:138–144.
- Yang, J.-G., Kim, J.-K., Kang, U.-G., and Lee, Y.-H. (2014). Coronary heart disease optimization system on adaptive-network-based fuzzy inference system and linear discriminant analysis (ANFIS-LDA). *Personal and Ubiquitous Computing*, 18(6):1351–1362.
- Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3):338–353.
- Zadeh, L. A. (1972). A fuzzy-set-theoretic interpretation of linguistic hedges. *Journal of Cybernetics*, 2(3):4–34.
- Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning—I. *Information Sciences*, 8(3):199–249.

- Zein-Sabatto, S., Mikhail, M., Bodruzzaman, M., DeSimio, M., and Derriso, M. (2013). Multistage fuzzy inference system for decision making and fusion in fatigue crack detection of aircraft structures. In *AIAA Infotech at Aerospace (I at A) Conference*.
- Zhuang, Y.-t., Wu, F., Chen, C., and Pan, Y.-h. (2017). Challenges and opportunities: from big data to knowledge in AI 2.0. *Frontiers of Information Technology & Electronic Engineering*, 18(1):3–14.